ELCIRC: overview of formulation and code structure

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

ELCIRC User Group Meeting

OGI SCHOOL OF SCIENCE & ENGINEERING

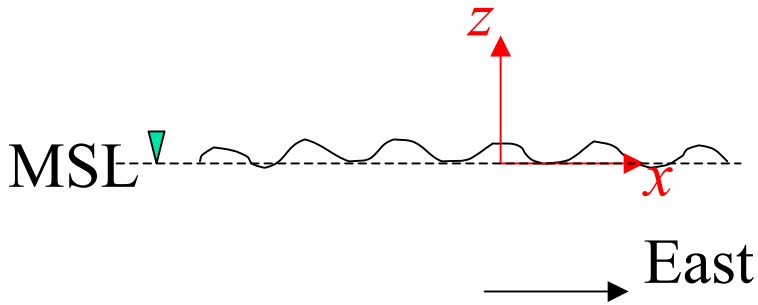
Talk outline

- Physical model
- Numerical model
- Code structure
- Benchmarks
- SELFE: a higher-order alternative

Physical model

- 3D Navier-Stokes equations
  - Continuity (Conservation of mass):

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0$$



  - Conservation of momentum:

$$\frac{Du}{Dt}=fv-\frac{1}{\rho}\frac{\partial p}{\partial x}+\cancel{\frac{1}{\rho}\nabla\cdot(E^h\nabla u)}+\frac{1}{\rho}\frac{\partial}{\partial z}\left(K_{mv}\frac{\partial u}{\partial z}\right),$$

$$\frac{Dv}{Dt}=-fu-\frac{1}{\rho}\frac{\partial p}{\partial y}+\cancel{\frac{1}{\rho}\nabla\cdot(E^h\nabla v)}+\frac{1}{\rho}\frac{\partial}{\partial z}\left(K_{mv}\frac{\partial v}{\partial z}\right),$$

$$\cancel{\frac{Dw}{Dt}=-g-\frac{1}{\rho}\frac{\partial p}{\partial z}+\frac{1}{\rho}\nabla\cdot(E^h\nabla w)+\frac{1}{\rho}\frac{\partial}{\partial z}\left(K_{mv}\frac{\partial w}{\partial z}\right)},$$

$$\left(\frac{D}{Dt}=\frac{\partial}{\partial t}+\mathbf{u}_3\cdot\nabla_3,\quad \mathbf{u}_3=(u,v,w),\quad \nabla_3=\left(\frac{\partial}{\partial x},\frac{\partial}{\partial y},\frac{\partial}{\partial z}\right)\right)$$

Wave continuity equation

- Starting from continuity equation:

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

  - Free-surface boundary condition:

$$w = \eta_t + u\eta_x + v\eta_y, \text{ at } z = \eta(x,y,t)$$

  - bottom boundary condition:

$$w = -uh_x - vh_y \text{ at } z = -h(x,y)$$

- Integrating continuity eq. along *z* from bottom to free surface:

$$\int_{-h}^{\eta} \frac{\partial u}{\partial x} dz + \int_{-h}^{\eta} \frac{\partial v}{\partial y} dz + w\Big|_{z=-h}^{z=\eta} = 0$$

$$\int_{-h}^{\eta} \frac{\partial u}{\partial x} dz = \frac{\partial}{\partial x}\int_{-h}^{\eta} u dz - (u\big|_{z=\eta} \cdot \eta_x + u\big|_{z=-h} \cdot h_x)$$

$$\int_{-h}^{\eta} \frac{\partial v}{\partial y} dz = \frac{\partial}{\partial y}\int_{-h}^{\eta} v dz - (v\big|_{z=\eta} \cdot \eta_y + v\big|_{z=-h} \cdot h_y)$$

$$\Rightarrow \frac{\partial \eta}{\partial t} + \frac{\partial}{\partial x}\int_{-h}^{\eta} u dz + \frac{\partial}{\partial y}\int_{-h}^{\eta} v dz = 0,$$

$$\text{or in vector form}: \quad \frac{\partial \eta}{\partial t} + \nabla \cdot \int_{-h}^{\eta} \mathbf{u} dz = 0$$

Approximations

- Additional assumptions for pressure and density: hydrostatic shallow water model
  - *p*:

$$\frac{\partial p}{\partial z} = -\rho g \Rightarrow p = p_A + g\int_z^\eta \rho d\zeta \Rightarrow \nabla p = \nabla p_A + \underbrace{g\rho\nabla\eta}_{\text{barotropic}} + \underbrace{g\int_z^\eta \nabla\rho d\zeta}_{\text{baroclinic}}$$

  - Density is constant except for the baroclinic term (Boussinesq approximation):

$$\rho = \rho_0$$

- Adequacy of hydrostatic shallow water model

Governing equations

- Continuity equation

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}+\frac{\partial w}{\partial z}=0,\ \text{or}: \quad \nabla\cdot\mathbf{u}+\frac{\partial w}{\partial z}=0 \qquad (\mathbf{u}=(u,v))$$

- Momentum equation

$$\frac{Du}{Dt}=fv-g\frac{\partial\eta}{\partial x}-\frac{g}{\rho_0}\int_z^\eta\frac{\partial\rho}{\partial x}d\zeta+\frac{\partial}{\partial z}\left(K_{mv}\frac{\partial u}{\partial z}\right),$$

$$\frac{Dv}{Dt}=-fu-g\frac{\partial\eta}{\partial y}-\frac{g}{\rho_0}\int_z^\eta\frac{\partial\rho}{\partial y}d\zeta+\frac{\partial}{\partial z}\left(K_{mv}\frac{\partial v}{\partial z}\right),$$

- Wave continuity equation

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

Barotropic model

- Equation of state:

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

- Transport of salt and temperature

$$\frac{Dc}{Dt}=\frac{\partial}{\partial z}\left(K_{hv}\frac{\partial c}{\partial z}\right)+Q, \qquad c=(S,T)$$

Turbulence closure (Umlauf and Burchard 2003)

- Equations for turbulent kinetic energy (TKE) and mixing length:

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

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

$$M^2 = \left(\frac{\partial u}{\partial z}\right)^2 + \left(\frac{\partial v}{\partial z}\right)^2, \quad N^2 = \frac{g}{\rho_0}\frac{\partial \rho}{\partial z}, \quad \varepsilon = (c_{\mu}^{0})^3 k^{1.5+m/n}\psi^{-1/n}$$

$$K_{mv} \equiv \nu_t = c_{\mu}k^{1/2}\ell, \; K_{hv} = c_{\mu}^{'}k^{1/2}\ell, \; \nu_k^{\psi} = \frac{\nu_t}{\sigma_k^{\psi}}, \; \nu_{\psi} = \frac{\nu_t}{\sigma_{\psi}}$$

- Wall proximity function (for k-kl only)

$$F_{wall} = 1 + E_2\left(\frac{\ell}{kL_b}\right)^2 + E_3\left(\frac{\ell}{kL_s}\right)^2$$

- Stability functions of Kantha and Clayson (1994)

$$c_{\mu} = \sqrt{2}s_m, c_{\mu}^{'} = \sqrt{2}s_h,$$

$$s_h = \frac{0.4939}{1-30.19G_h}, s_m = \frac{0.392+17.07s_hG_h}{1-6.127G_h},$$

$$G_h = \frac{G_{h\_u} - (G_{h\_u} - 0.02)^2}{G_{h\_u} + 0.0233 - 0.04}, \; -0.28 \le G_{h\_u} = \frac{N^2\ell^2}{2k} \le 0.0233$$

Formulation: GLS constants

	p	m	n	σ_k^{ψ}	σ_{ψ}	$c_{\psi 1}$	$c_{\psi 2}$	$c_{\psi 3}^{+}$	$c_{\psi 3}^{-}$ *
k-ε	3	1.5	-1	1.0	1.3	1.44	1.92	1	-0.52
k-kl "MY2.5"	0	1	1	2.44	2.44	0.9	0.5	1	2.53
k-ω	-1	0.5	-1	2.0	2.0	0.555	0.833	1	-0.58
UB	2	1	-0.67	0.8	1.07	1	1.22	1	0.1

*: with Kantha and Clayson's ASM

Vertical boundary conditions

- Free-surface b.c.:

$$\rho_0 K_{mv}\frac{\partial \mathbf{u}}{\partial z} = \boldsymbol{\tau}_{wind}, \quad \boxed{w = \eta_t + \mathbf{u}\cdot\nabla\eta = \eta_t}, \quad K_{hv}\frac{\partial S}{\partial z} = 0, \quad K_{hv}\frac{\partial T}{\partial z} = \hat{T},$$

$$\boxed{k = B_1^{2/3}E^v\left|\frac{\partial \mathbf{u}}{\partial z}\right| = B_1^{2/3}\left|\boldsymbol{\tau}_{wind}\right|/\rho_0, \quad \ell/\kappa \to d_s.}$$

$$\boldsymbol{\tau}_{wind} = \rho_{air}C_{Ds}\left|\mathbf{u}_{wind}\right|\mathbf{u}_{wind}$$

$$C_{Ds} = (0.61 + 0.063\left|\mathbf{u}_{wind}\right|)/1000.$$

- Bottom b.c.:

$$\rho_0 K_{mv}\frac{\partial \mathbf{u}}{\partial z} = \boldsymbol{\tau}_b, \quad w = -\mathbf{u}\cdot\nabla h = 0, \quad K_{hv}\frac{\partial S}{\partial z} = 0, \quad K_{hv}\frac{\partial T}{\partial z} = 0,$$

$$\boxed{k = B_1^{2/3}E^v\left|\frac{\partial \mathbf{u}}{\partial z}\right| = B_1^{2/3}\left|\boldsymbol{\tau}_b\right|/\rho_0, \quad \ell/\kappa \to d_b.}$$

$$\boldsymbol{\tau}_b = \rho_0 C_D\left|\mathbf{u}\right|\mathbf{u} \equiv \rho_0\tau_b\mathbf{u},$$

C_D either specified or or fitted to satisfy the "law of wall"

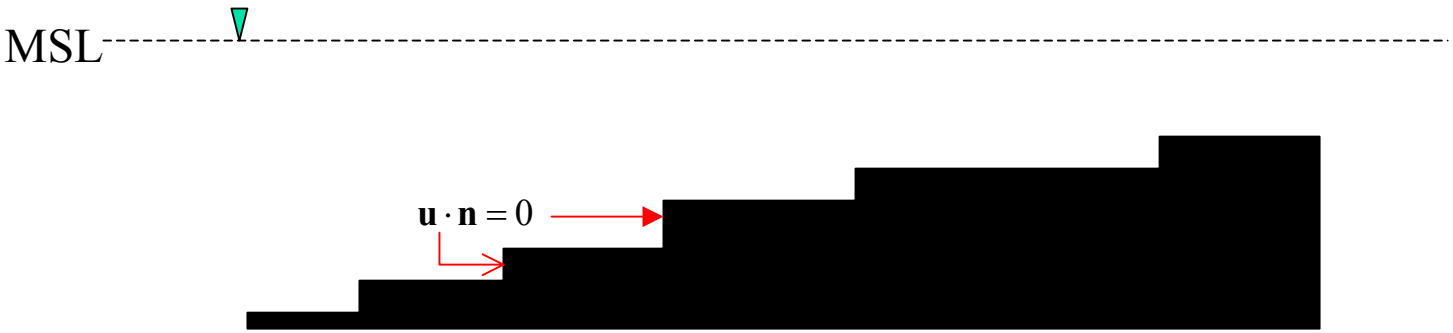
Horizontal boundary conditions

- Problem specific
- Elevations are usually specified on open boundaries (e.g., tides)
- Normal velocity vanishes on all land boundaries
- Velocity may be specified on open boundaries to give rise to a particular river discharge
- Vertical velocity is not specified on horizontal boundaries
- S, T may or may not be specified on open boundaries
- k and ℓ are not specified on horizontal boundaries

	η	(u,v)	T,S	w	k, l
Land boundaries	Not specified	u_n=0; u_t not specified	Not specified	Not specified	Not specified
Open boundaries	1. Direchlet 2. o.b.c.	1. Direchlet (thru total river discharge) 2. Not specified	1. Direchlet 2. Nudging	Not specified	Not specified

Other boundary conditions

- Created by the staircase representation of the bottom (inherent from the numerical model)



Numerical Scheme: horizontal grid

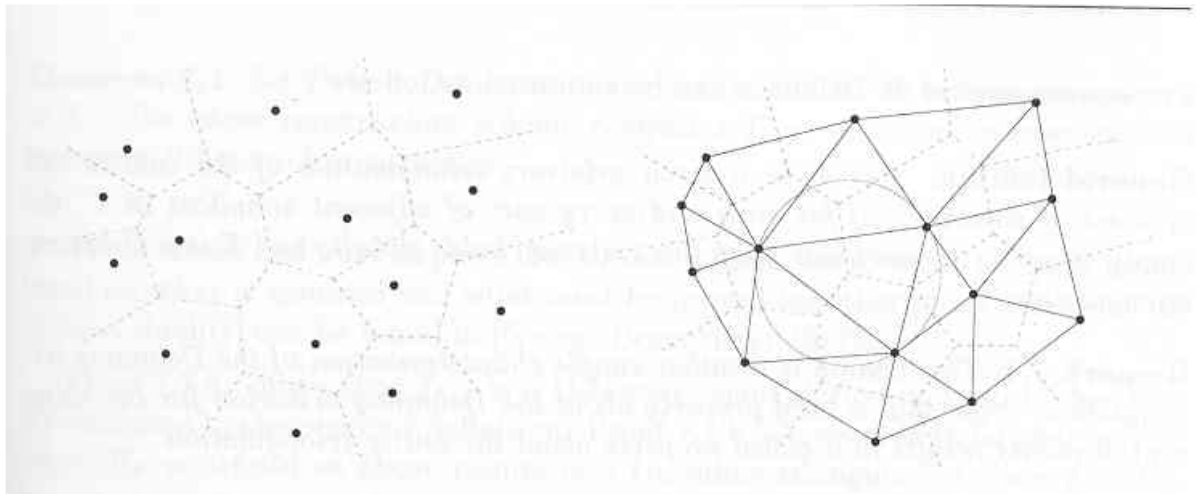
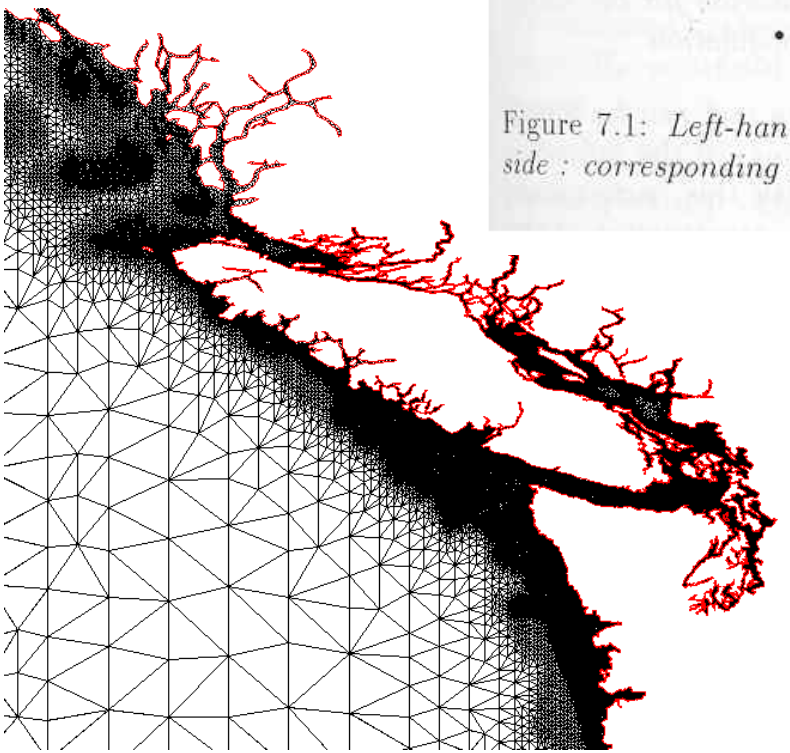
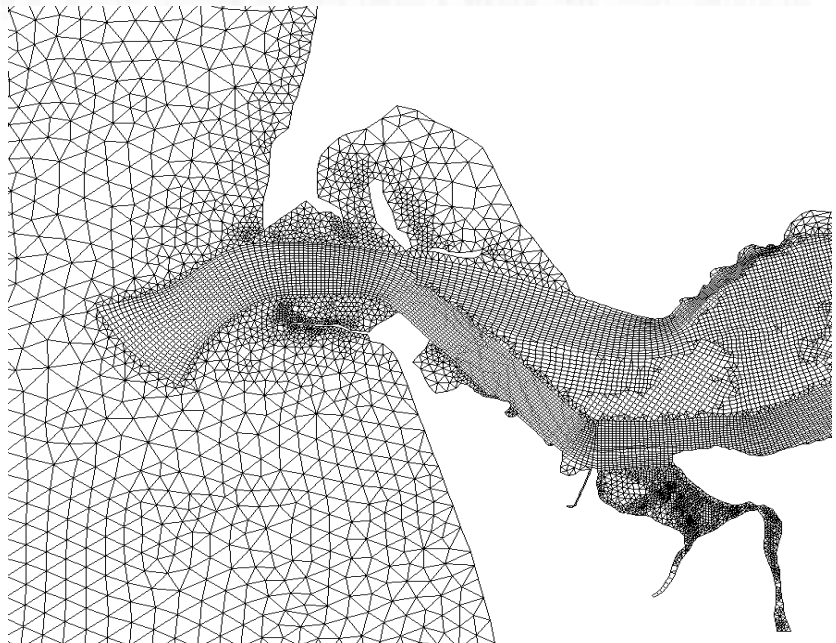
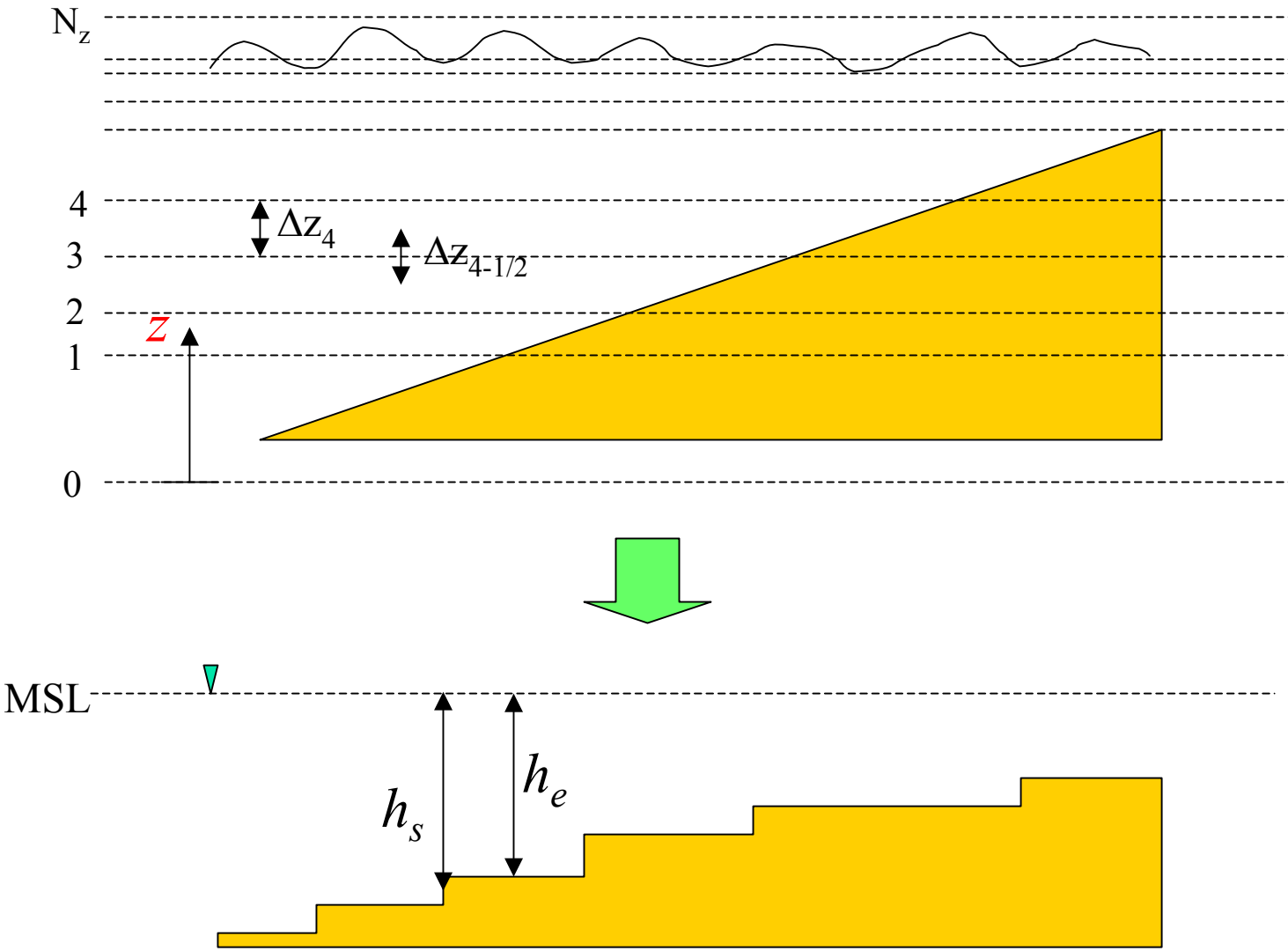


Figure 7.1: *Left-hand side : Voronoï diagram (in two dimensions). Right-hand side : corresponding Delaunay triangulation.*





Numerical Scheme: vertical grid



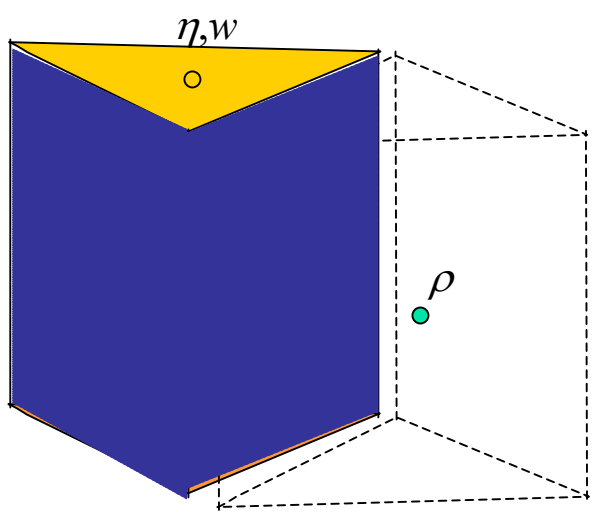
Numerical Scheme: notations

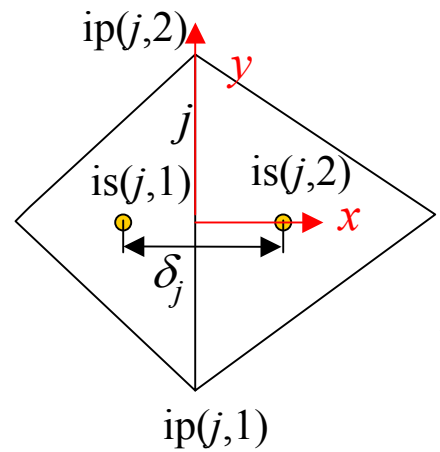
Primary unknowns:

$\eta,\ u,\ v,\ w,\ \rho,\ S,\ T$

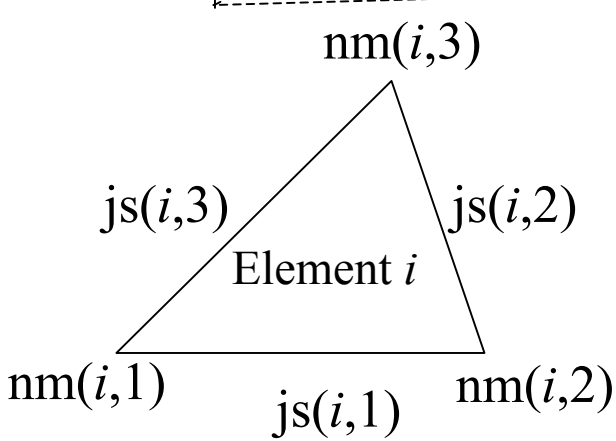
$K_{mv},\ K_{hv},\ \nu_{\psi},\ k,\ \ell$

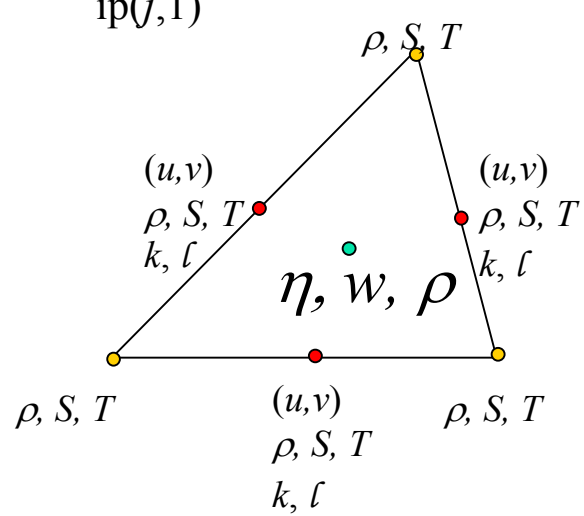
$$u \equiv u_n,\ \ v \equiv u_t$$





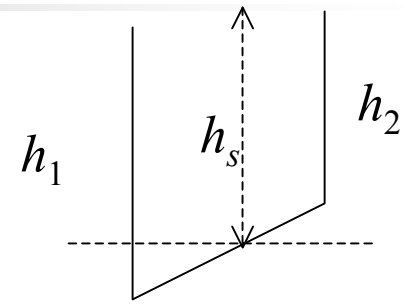
$$\left.\frac{\partial \eta}{\partial x}\right|_j = \frac{\eta_{is(j,2)} - \eta_{is(j,1)}}{\delta_j}$$



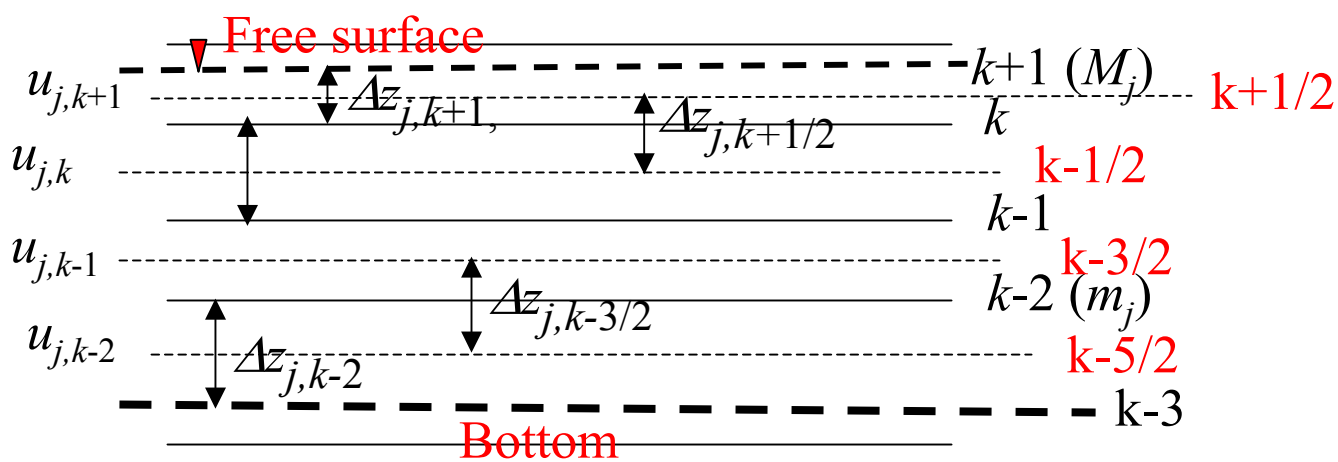


Vertical structure of variables

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



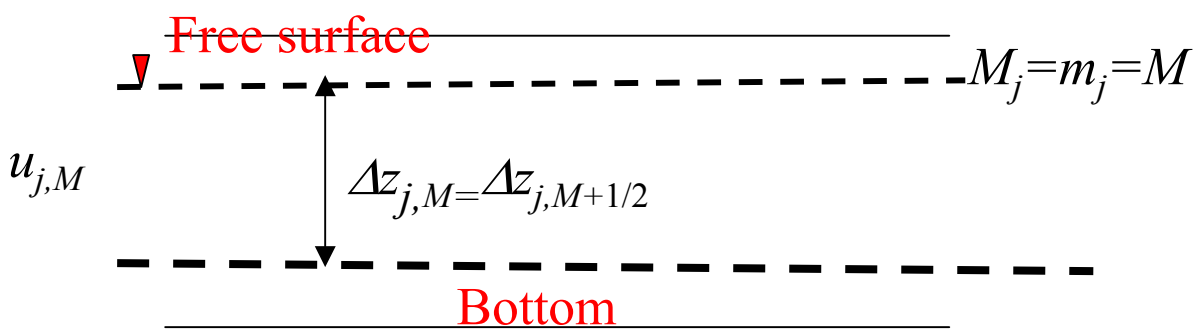
From one *side*'s (side j) perspective:



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

A special case

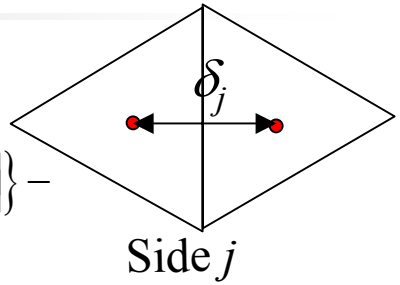
- When there is only one layer:



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

Discretized momentum equation

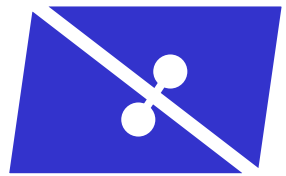
- Finite difference applied to face center (j,k) ($m_j \le k \le M_j$):



$$\frac{u_{j,k}^{n+1}-u_*^n}{\Delta t}=f_j v_{j,k}^n-\frac{g\omega_j}{\delta_j}\left\{\alpha\left[\eta_{is(j,2)}^{n+1}-\eta_{is(j,1)}^{n+1}\right]+(1-\alpha)\left[\eta_{is(j,2)}^{n}-\eta_{is(j,1)}^{n}\right]\right\}-$$

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

$$\frac{1}{\Delta z_{j,k}^n}\left[E_{j,k}^v\frac{u_{j,k+1}^{n+1}-u_{j,k}^{n+1}}{\Delta z_{j,k+1/2}^n}-E_{j,k-1}^v\frac{u_{j,k}^{n+1}-u_{j,k-1}^{n+1}}{\Delta z_{j,k-1/2}^n}\right]$$

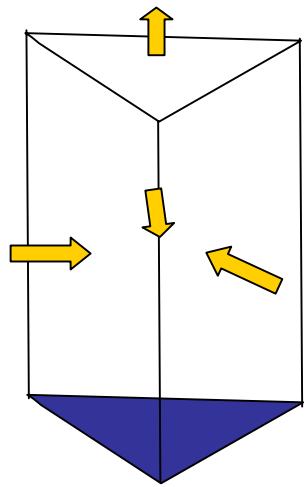


- Vertical boundary conditions:

$$E_{j,m_j-1}^v\frac{u_{j,m_j}^{n+1}-u_{j,m_j-1}^{n+1}}{\Delta z_{j,m_j-1/2}^n}=\tau_b u_{j,m_j}^{n+1},$$

$$E_{j,M_j}^v\frac{u_{j,M_j+1}^{n+1}-u_{j,M_j}^{n+1}}{\Delta z_{j,M_j+1/2}^n}=\tau_{wind}^x/\rho_0.$$

- Similar for v



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

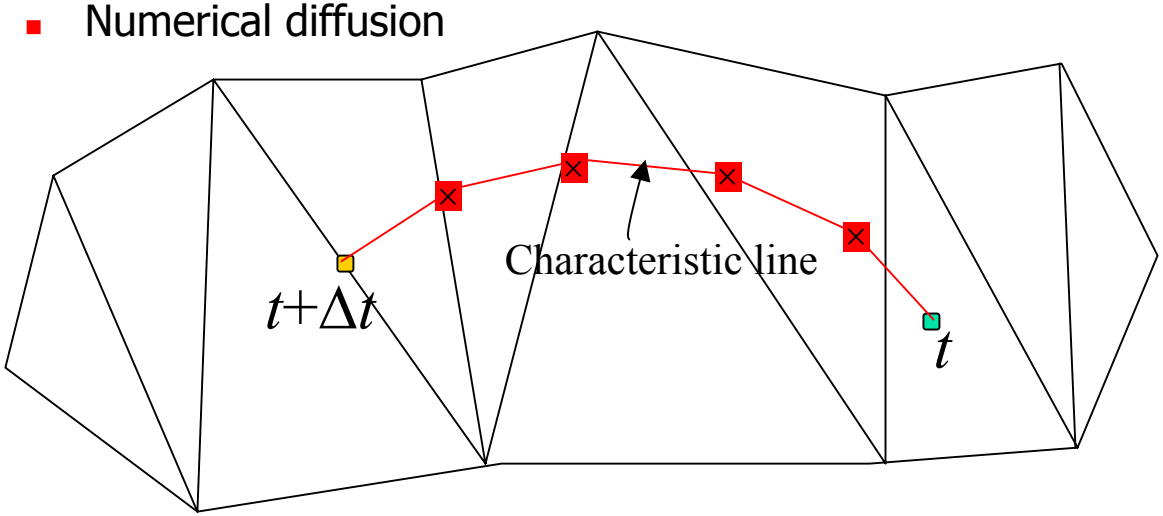
$$\int_{\Omega_i}\left(\frac{\partial\eta}{\partial t}+\nabla\cdot\int_{-h}^{\eta}\mathbf{u}dz\right)d\Omega_i=0\ \Rightarrow\ \int_{\Omega_i}\frac{\partial\eta}{\partial t}d\Omega_i+\int_{\Gamma_i}d\Gamma_i\int_{-h}^{\eta}u_n dz=0$$

$$P_i\frac{\eta_i^{n+1}-\eta_i^n}{\Delta t}+\alpha\sum_{l=1}^{3}s_{i,l}\lambda_{jsl}\sum_{k=m_{jsl}}^{m_{jsl}}\Delta z_{jsl,k}^n u_{jsl,k}^{n+1}+(1-\alpha)\sum_{l=1}^{3}s_{i,l}\lambda_{jsl}\sum_{k=m_{jsl}}^{m_{jsl}}\Delta z_{jsl,k}^n u_{jsl,k}^n=0$$

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

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
  - 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:

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

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

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

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

$\boxed{\mathbf{A}_j, \mathbf{G}_j}$

- Substitution of the first to third equation leads to:

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

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

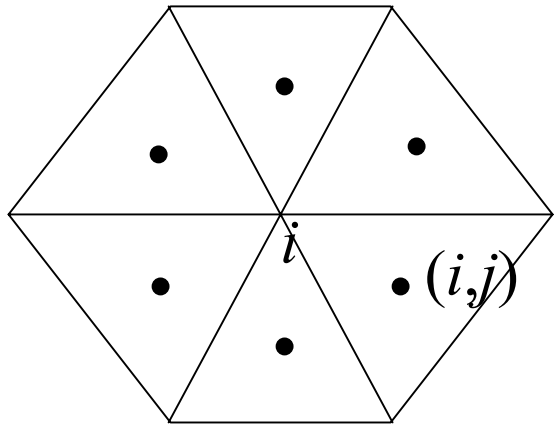
$$\eta_i^n - \frac{(1-\alpha)\Delta t}{P_i}\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\left[\mathbf{\Delta Z}_{jsl}^n\right]^T\mathbf{U}_{jsl}^n - \frac{\alpha\Delta t}{P_i}\sum_{l=1}^{3} s_{i,l}\lambda_{jsl}\left[\mathbf{\Delta Z}_{jsl}^n\right]^T\mathbf{A}_{jsl}^{-1}\mathbf{G}_{jsl}^n,\ \ (i = 1,\cdots,N_e)$$

Tangential velocity

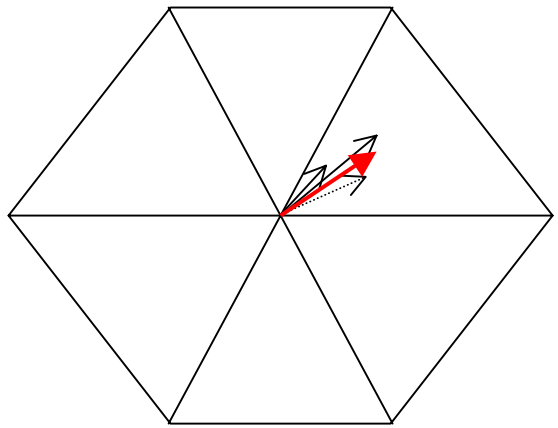
- Elevations at nodes:

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

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



- Averaging around the ball

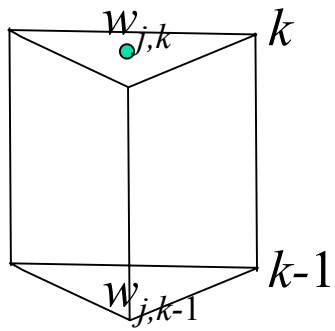


Vertical velocity

- 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:

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

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

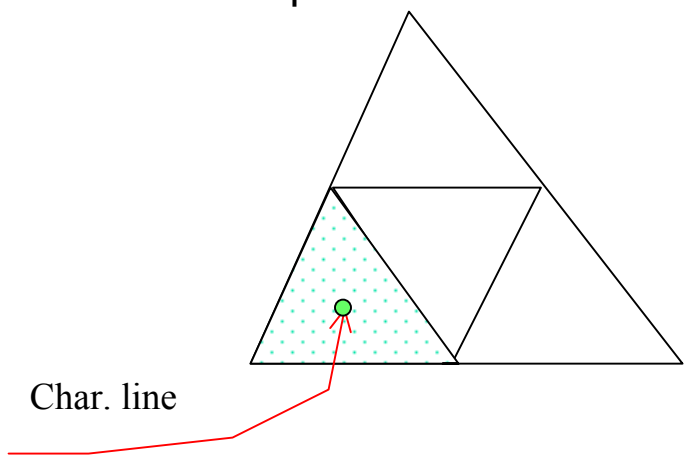


Transport equation

- 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

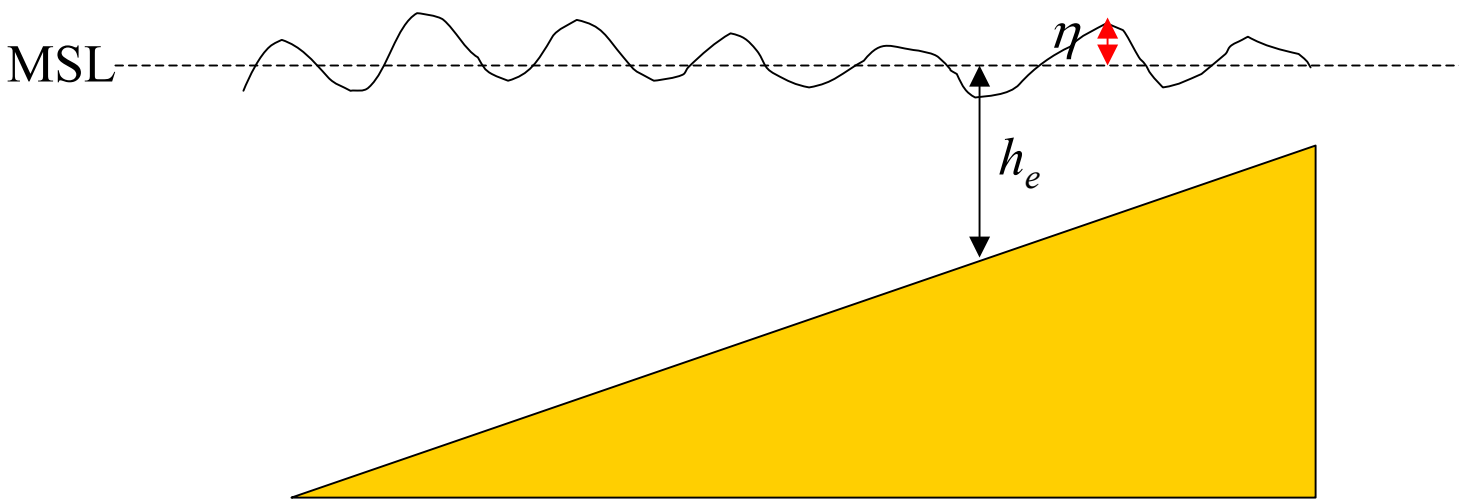


Turbulence closure

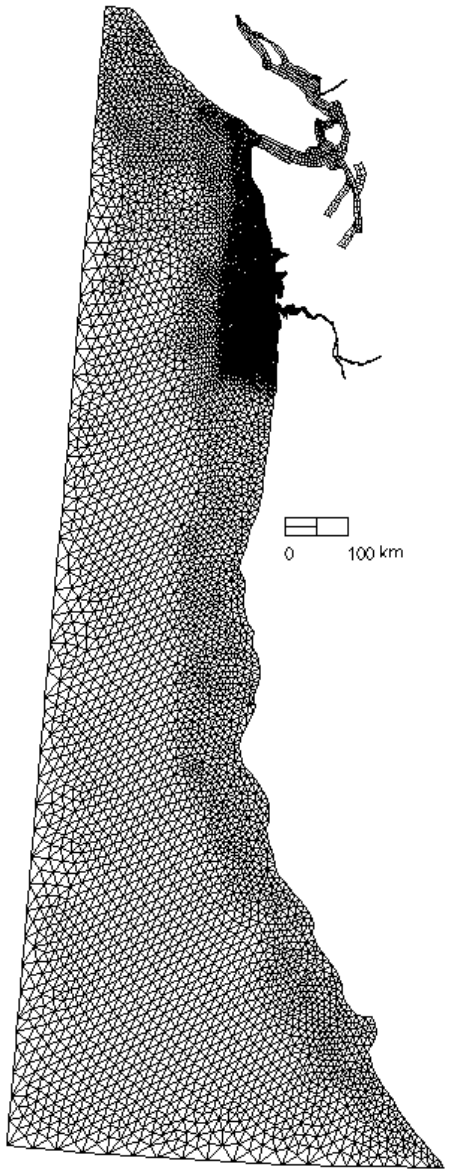
- Finite difference method
  - k and ℓ are defined at side centers and half levels, and diffusivities at whole levels.
  - k and ℓ are then interpolated back onto whole levels to evaluate diffusivities.
  - Turn off the advection
- Because of the oscillatory nature of the closure eqs., terms are treated implicitly whenever possible
  - The production term (i.e., buoyancy + shear) is treated implicitly when negative, or explicitly when positive.
- Details, details, details……
  - Initial condition
  - Bounds & clippings

Adjustment of free surface

- Free-surface indices are adjusted with the newly computed elevations
- If the total depth of an element $h_e + \eta < h_0$, it is dried; a dry element can be re-wetted at a later time step when the total depth becomes positive again



Computational performance

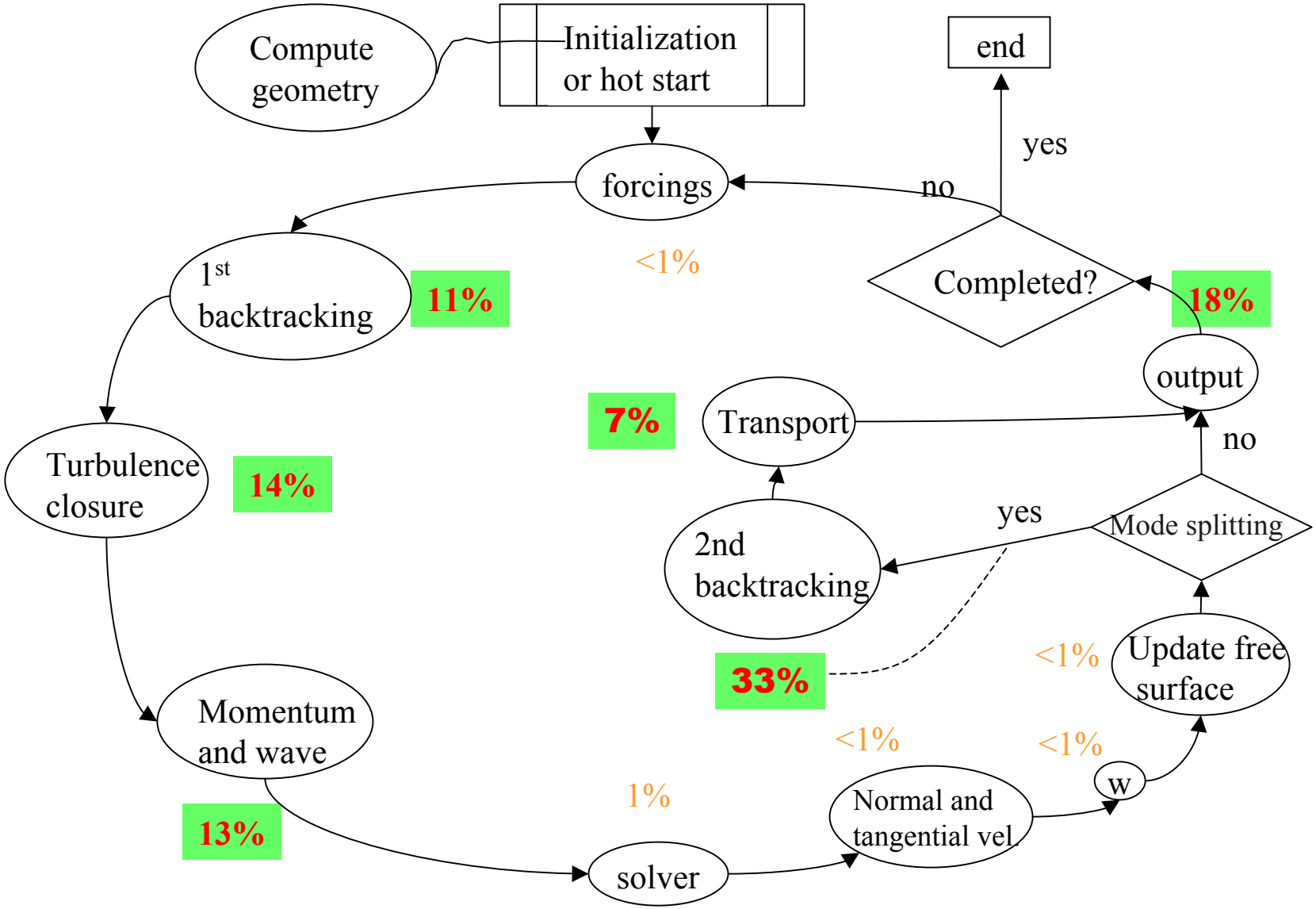


Grid specifications:

- **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



Forcings and preparations

- Compute bottom drag coefficient (N_s)
- Read new wind and heat fluxes (N_p)
- Compute wind stress (N_s)
- Read in time series from *.th
- Compute # of subdivisions in btrack (N_p)

```
do i=1,np
  do k=1,nvrt
    .......
  enddo !k
enddo !i
```

Backtracking routines (1)

- Main loop:

```
do i=1,ns (or np)
  do k=kbs(i),kfs(i)
     initialize (x0,y0,z0), (u0,v0,z0) and jlev
     call btrack
      record btracked values (vnbt, vtbt,  tsdbt, ssdbt etc.)
  enddo !k
enddo !i
```

- Routine “btrack”

```
do idt=1,ndelt
  xt=x0-uu*dtb
  call quicksearch
  interpolate vel. at (xt,yt,zt)
  xt=x0 !copy end point to starting point
enddo !idt
```

Backtracking routines (2)

- Routine "quicksearch"

(1) check if the end point is inside the starting element;

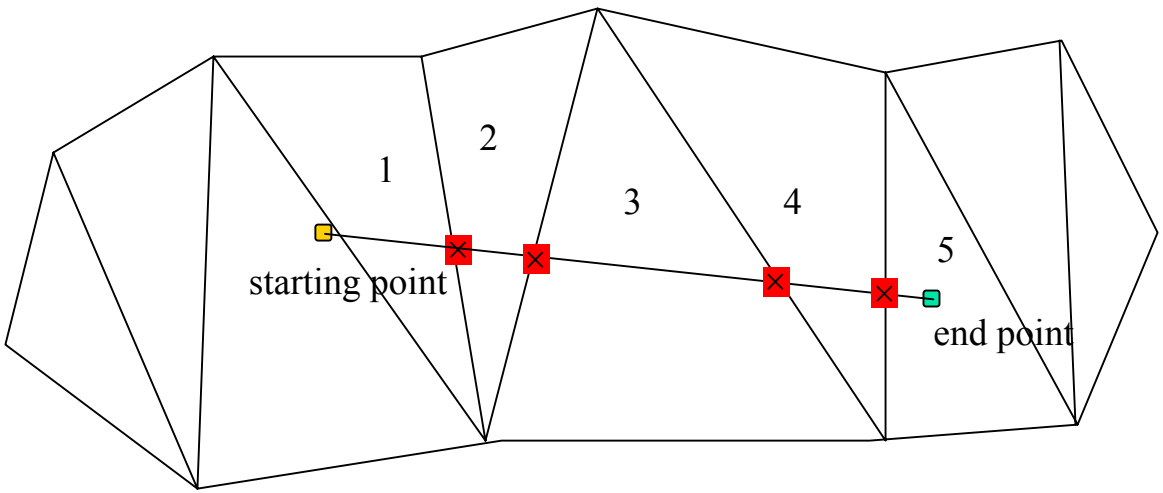
(2) find 1st intersecting side;

(3) proceed to the next element;

(4) if dry or horizontal boundary, slide using the tangential vel., and update the end point;

(5) check if the end point is inside the element; if not, find next intersecting side and go to (4);

(6) compute the vertical level index for the end point and exit.



Momentum and wave-continuity equations

- Momentum eq:

```
do i=1,ns
   construct matrices for each vertical;
   invert matrices;
   compute r.h.s.;
   compute products of some matrices (for wave-continuity equations);
enddo !i
```

- Wave-continuity eq:

```
do i=1,ne
   compute and store non-zero entries of a sparse matrix using previous info;
enddo !i
```

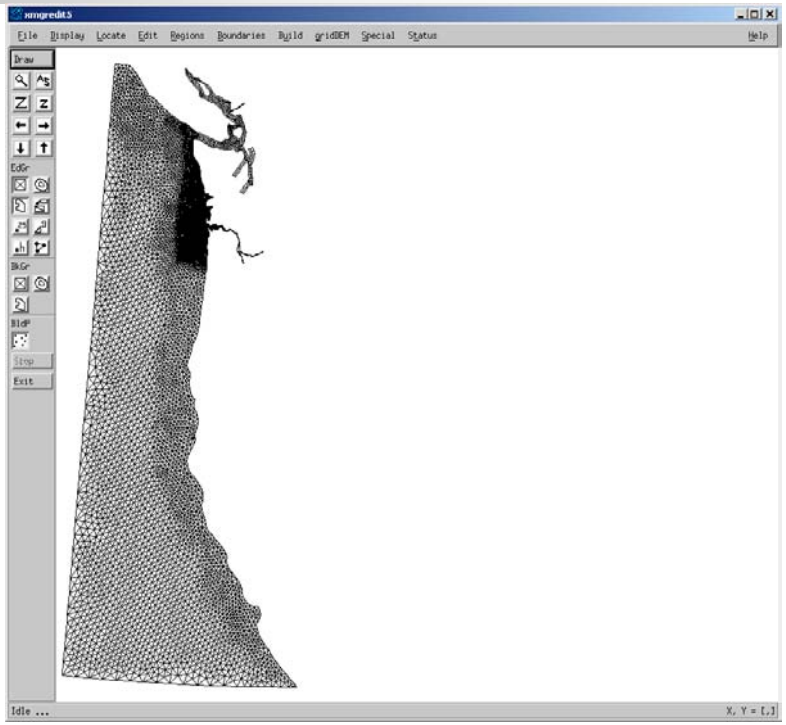
- Sparse matrix solver:
  - From ITPACK;
  - Uses standard iterative schemes (*pre-conditioner*) (e.g., Jacobi) together with *accelerators* like Conjugate Gradient;
  - Little else is known inside.

Preparing an Elcirc run

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

- 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 (no longer needed);
- 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 hgrid.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

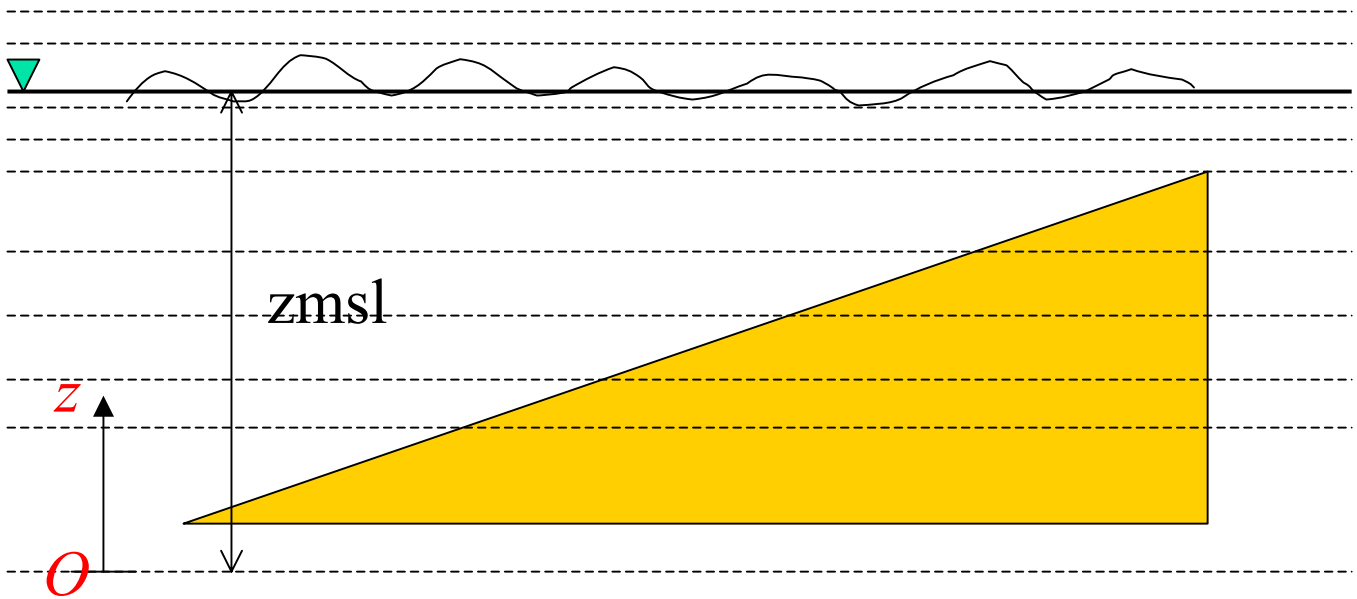
.....

59 0.80 4826.60

60 1.00 4827.60

61 2.00 4829.60

62 36.40 4866.00

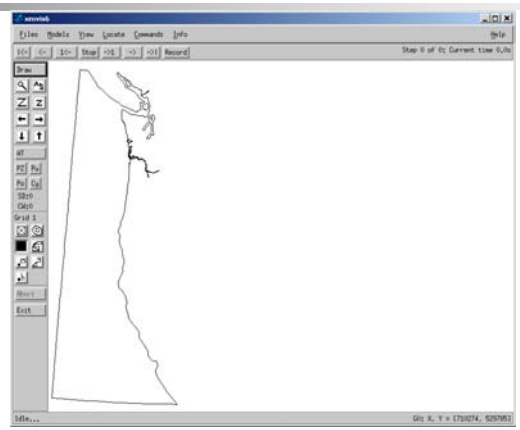


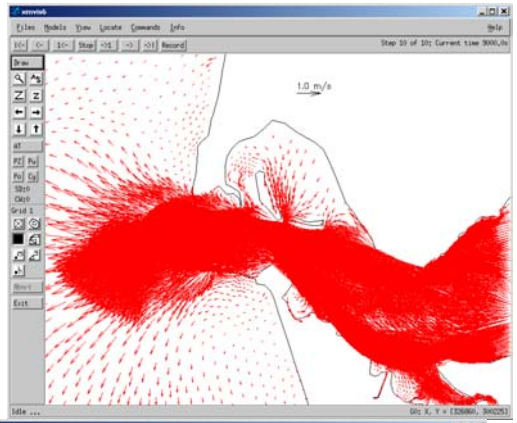
Param.in

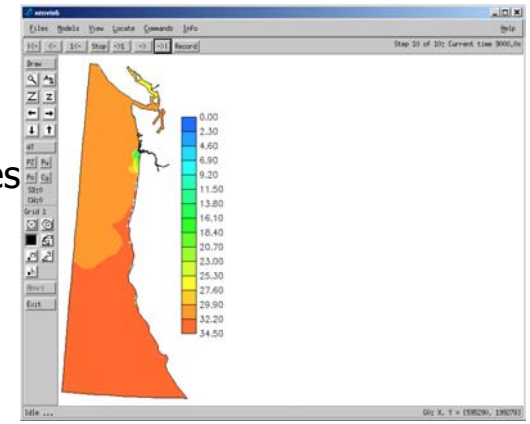
```
QUARTER ANNULAR TEST EXAMPLE 1
ELCIRC
1 NSCREEN
0 iforecast
0 IHOT
1  ICS
0.0 0.0  SLAM0,SFEA0
1.0
1 0    baroclinic/barotropic
4. 30. 0. 33.
5.   RNDAY
1 2.
2095.872   1047.936 Dt
2  nsubfl
5 90 NDELT
1 nadv
0.01 h0
0 ntau
0.  Cd
0   NCOR
0.0 CORI
2 3600.  NWS ← hdf
1 0.5
1 0 heat
0   turbulence 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
 .....
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
```

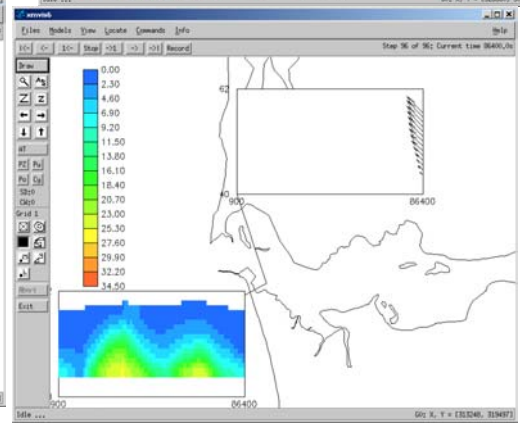
Visualization: XMvis6 (Turner et al.)

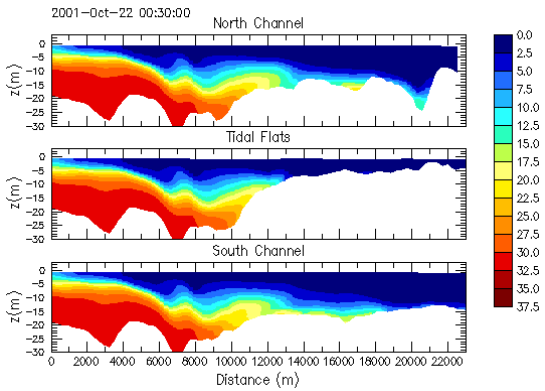
- Main global binary outputs:
  - *.61: 2D scalars (1_elev.61)
  - *.62: 2D vectors (3_wind.62)
  - *.63: 3D scalars (2_salt.63)
  - *.64: 3D vectors (7_hvel.64)
- Most useful functions:
  - Files
    - ELCIRC slabs ← horizontal levels
    - ELCIRC samples ← vertical profiles
    - ELCIRC surface/bottom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

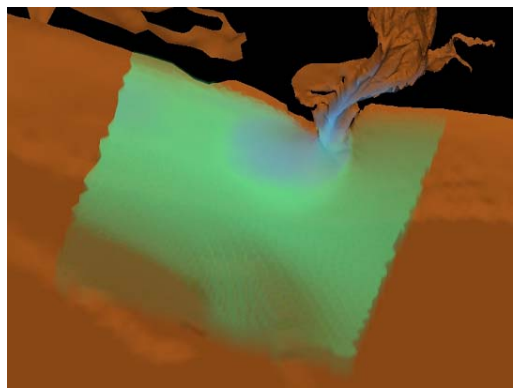








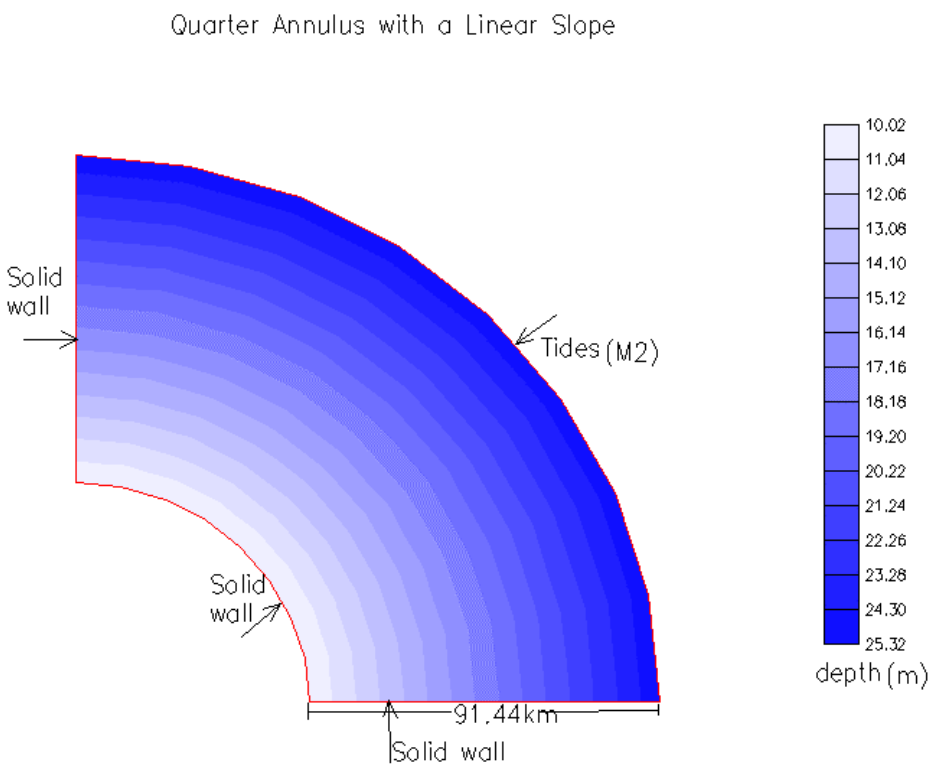




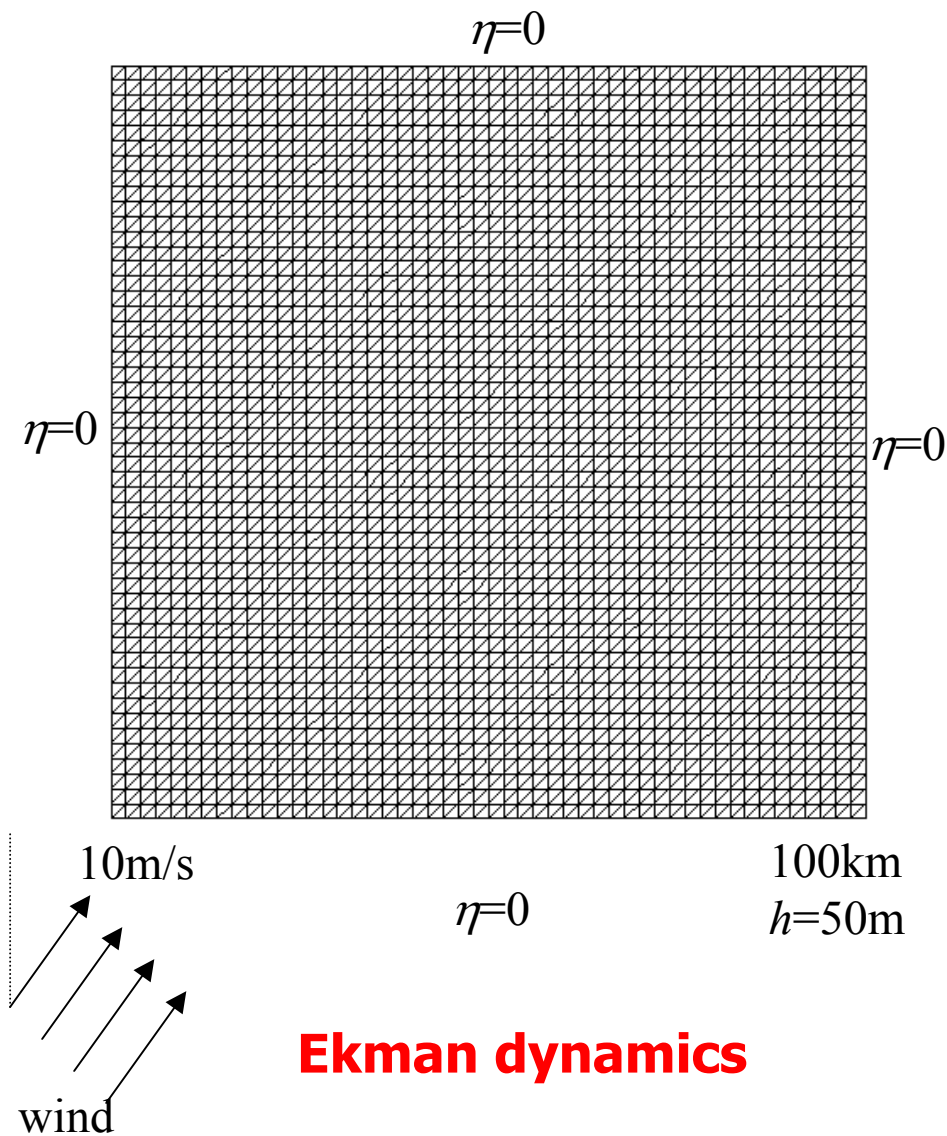
Practical issues

- Horizontal grid
  - Orthogonal vs. non-orthogonal elements
  - Triangles vs quads
  - Use uniform quads and general triangles
- Vertical grid
  - Adequate resolution for baroclinic applications
- Parameters
  - Baroclinic time step: optimal near $\Delta t = \Delta x / \sqrt{g'h}$
  - 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.)

Benchmarks



Quarter annulus

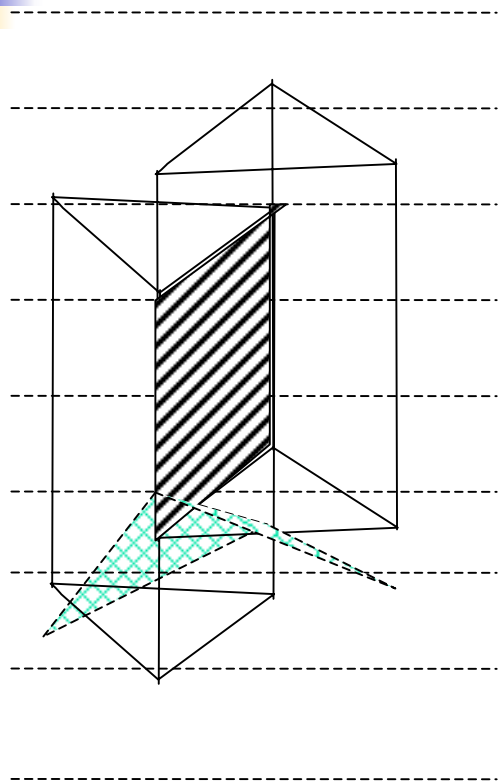


Ekman dynamics

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
  - Staircase bottom; orthogonality; low order method; non mass conservative transport; numerical diffusion
- The ugly
  - Inconsistencies and ambiguities of indices

Side and element indices



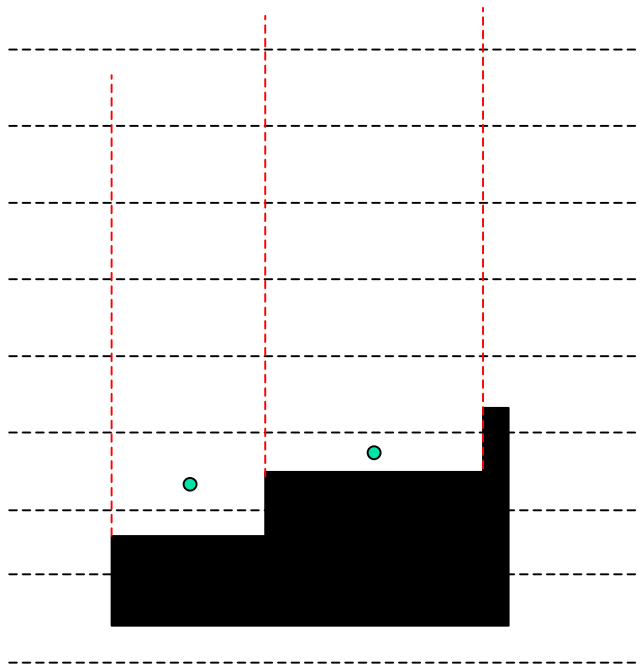
For a wet side j

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

For a dry side j

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

Baroclinic gradient



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

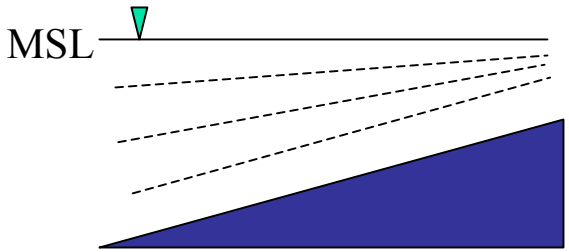
$$\mathbf{A}_j = \begin{pmatrix} \Delta z_{j,M_j} + a_{j,M_j-1/2} & -a_{j,M_j-1/2} & \cdots & 0 \\ -a_{j,M_j-1/2} & \Delta z_{j,M_j-1} + a_{j,M_j-1/2} + a_{j,M_j-3/2} & -a_{j,M_j-3/2} \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & \cdots & -a_{j,m_j+1/2} & \Delta z_{j,m_j-1} + a_{j,m_j+1/2} +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

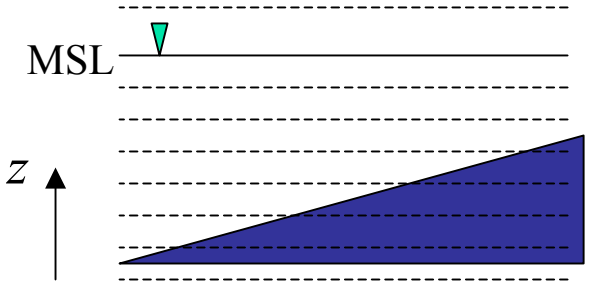
$$a_{j,k\pm 1/2} = E^v_{k\pm 1/2-1/2} \frac{\Delta t}{\Delta z^n_{j,k\pm 1/2}}$$

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

σ- and z-coordinate

$$\sigma = \frac{z-\eta}{h+\eta} \quad (-1 \le \sigma \le 0)$$



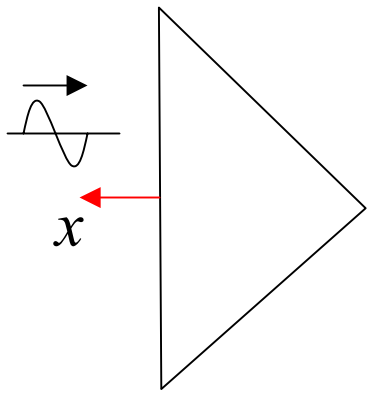


- Follows the bottom naturally
- Vertical domain is "uniform"
- Has problem when the depth is very shallow

- Deals with wetting and drying better
- A number of levels will be wasted

Radiation boundary condition

$$\frac{\partial \eta}{\partial t} + \sqrt{gh}\frac{\partial \eta}{\partial x} = 0$$



SELFE: Semi-implicit Eulerian-Lagrangian Finite Element

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

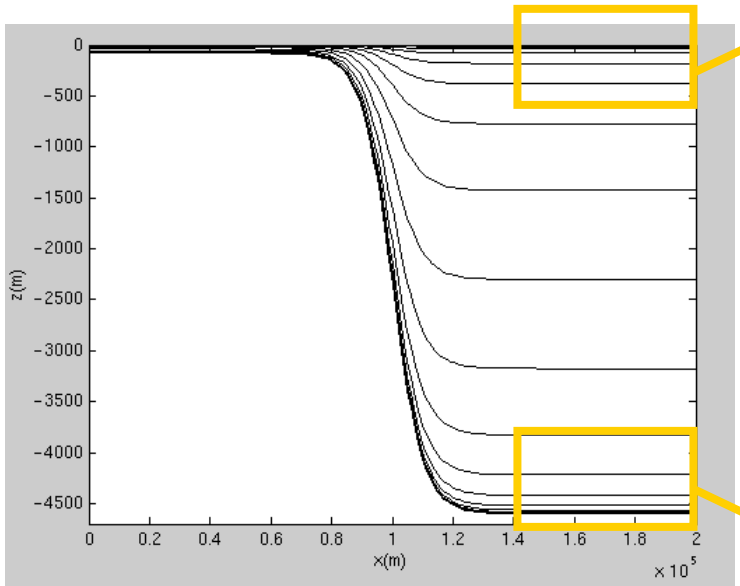
ELCIRC User Group Meeting

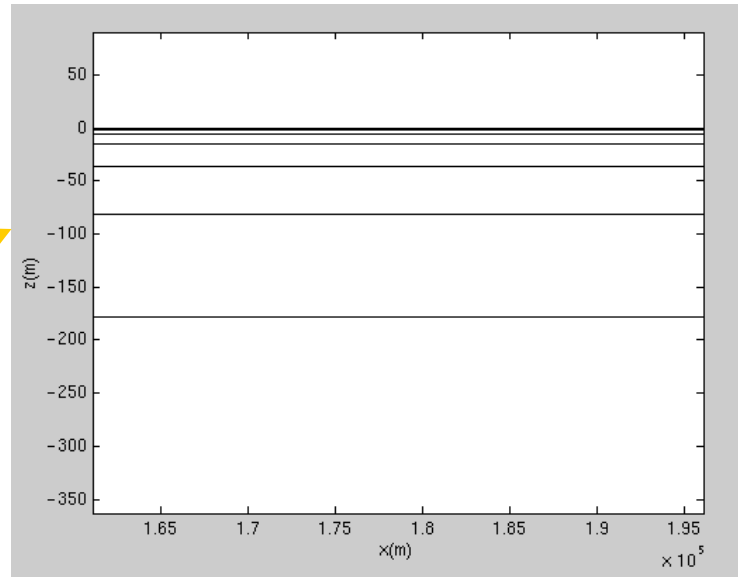
OGI SCHOOL OF SCIENCE & ENGINEERING

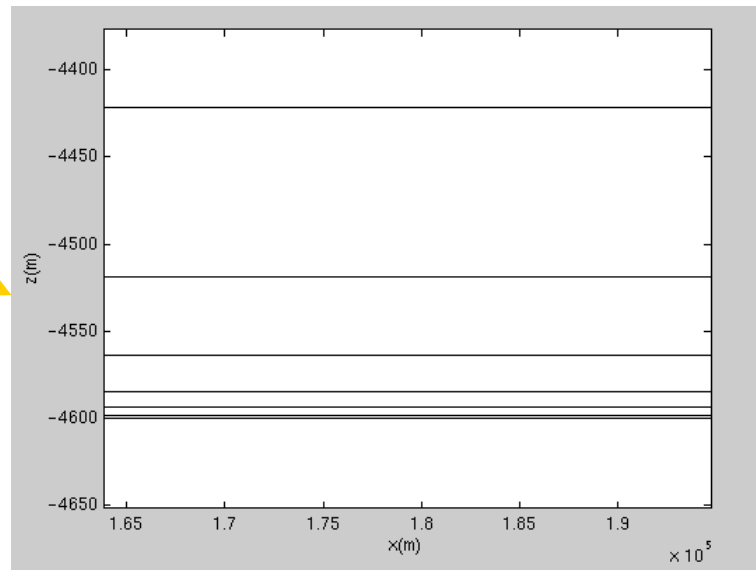
Comparison

Model name	**ELCIRC**	**SELFE**
Numerical method	Semi-implicit FD/FV	Semi-implicit FE/FV
Shape function (barotropic)	Constant	Linear
Advection	ELM	ELM with optional sub-division of grids
CFL restriction	No	No
Horizontal grid	Orthogonal unstructured	Unstructured
Vertical grid	*z*-coordinate	*S*-coordinate
Volume conservation	Numerically exact	Numerically not exact
Baroclinicity	FD with trapezoidal integration	Hybrid method with 4th- or 6th order integration
o.b.c. for elevation	Empirical	Natural
Transport eq	▪ FD ▪ FCT (in progress)	▪ FE ▪ FCT (in progress)

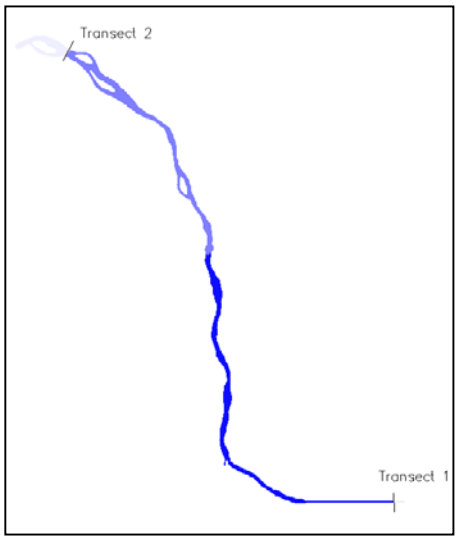
S-coordinates (Song & Haidvogel 1994)

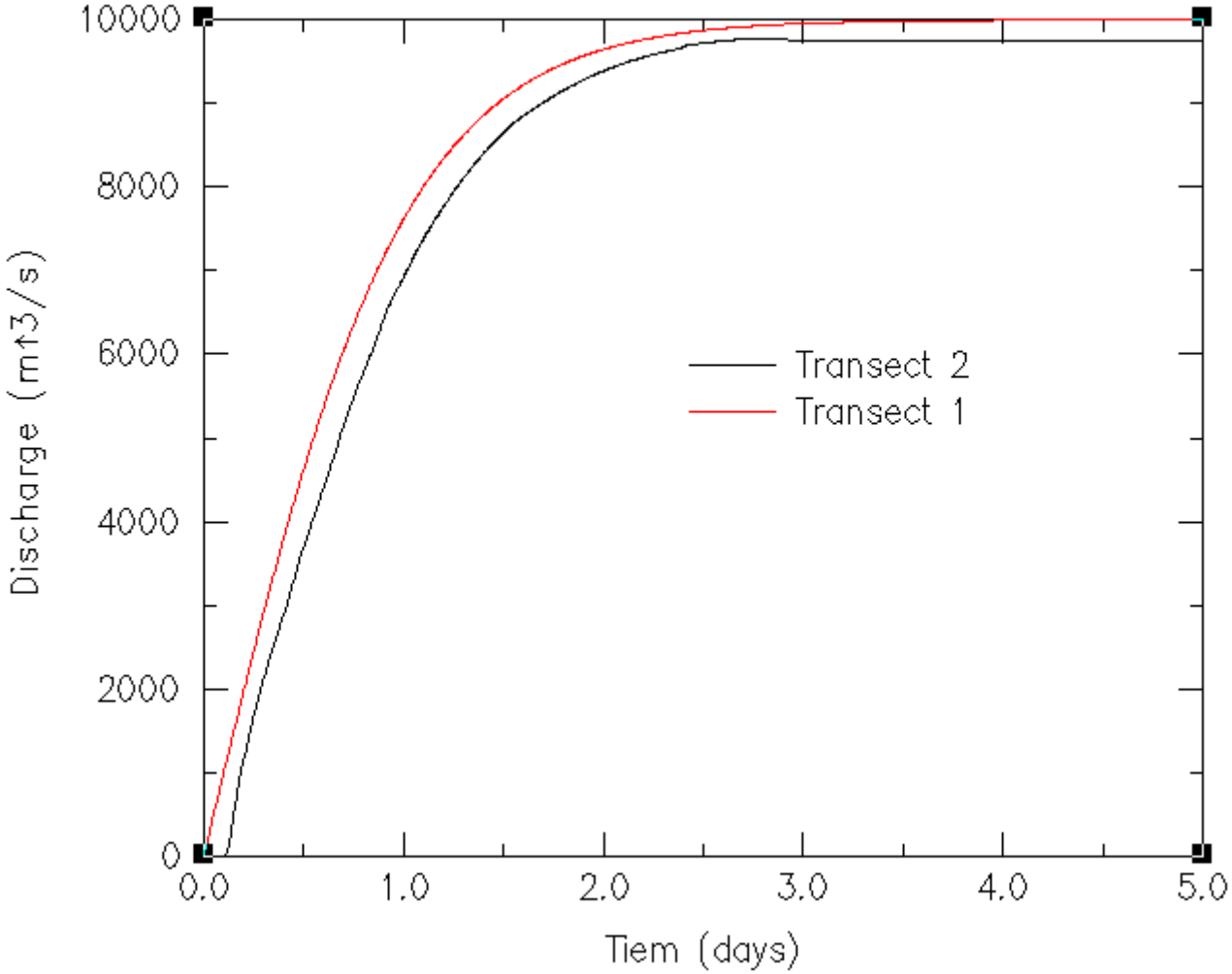






Volume conservation test





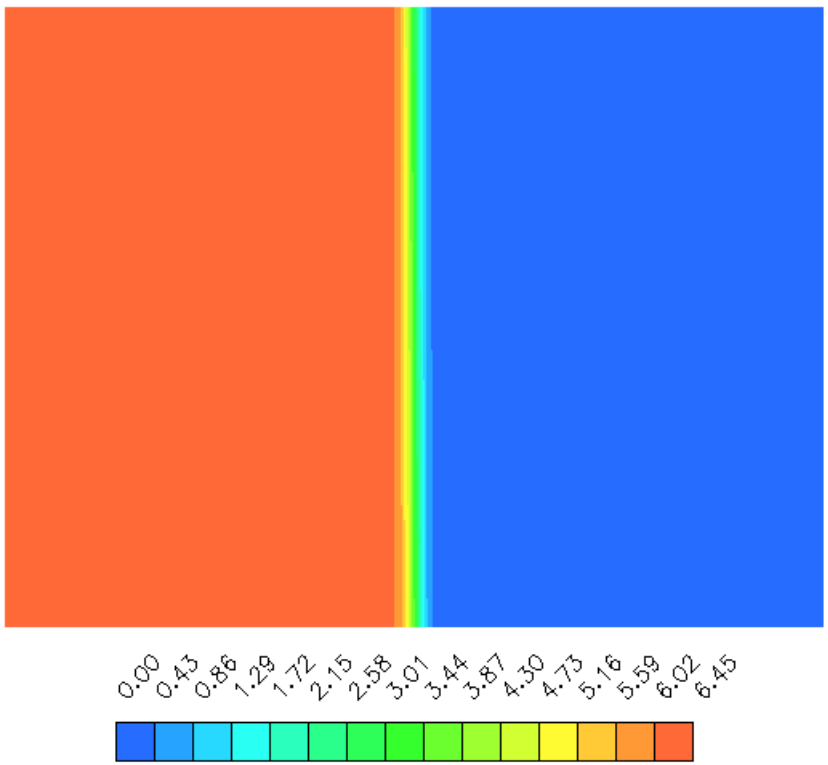
Error:

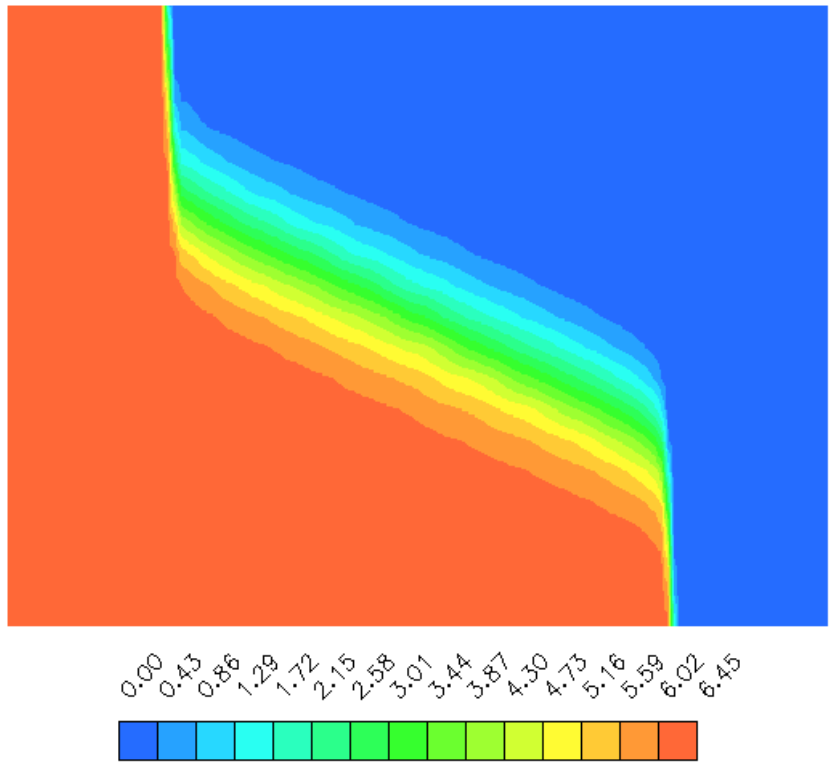
SELFE: ~2.5%

ELCIRC: <1%

Adjustment under gravity

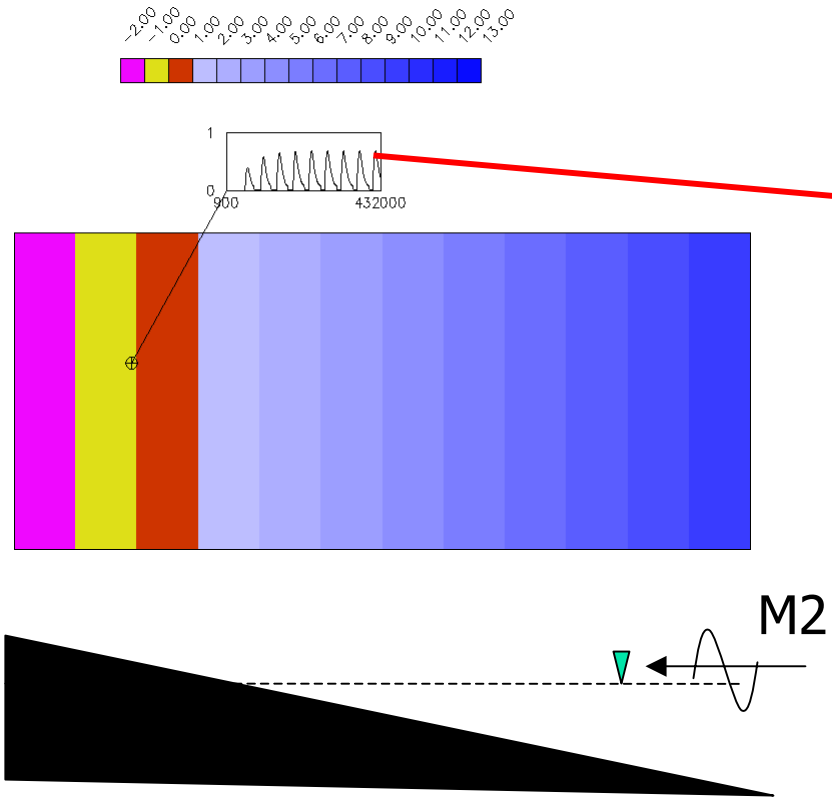
t=0hr *k-kl* *t*=12hr

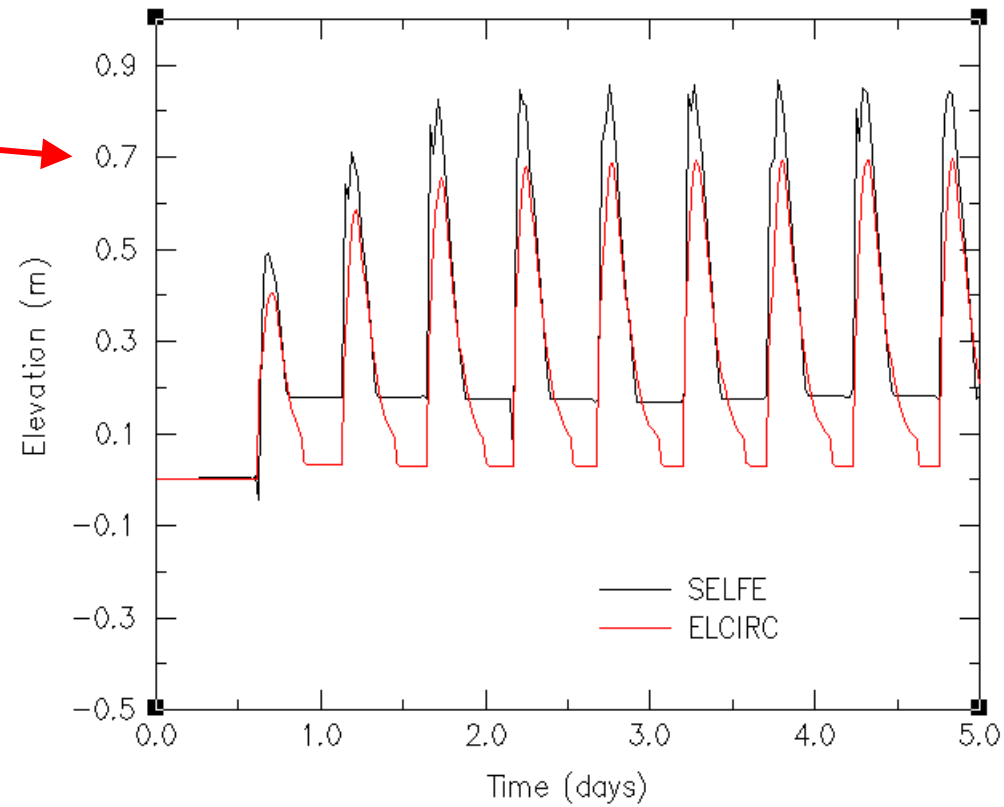




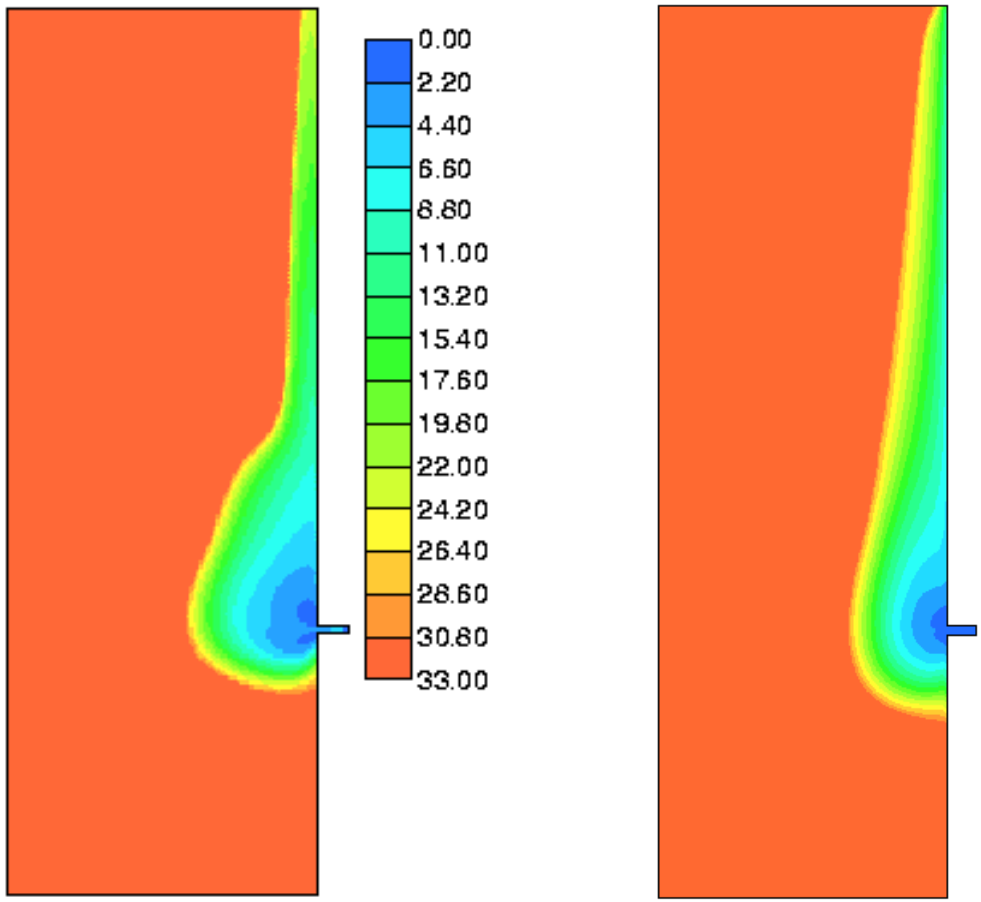
Internal wave speed:
SELFE: 92% (linear) or 93% (quadratic);
ELCIRC: 82% (best result: 88% with 4x resolution)
ROMS: 90-94%

Wetting and drying





Plume test



ELCIRC

SELFE (preliminary)

List of things to do and discussions

- *A σ-coordinates version of ELCIRC
- * An alternative, non-ELM, scalar transport algorithm, to seek strict mass conservation and to reduce numerical diffusion.
- * SELFE
- Non-hydrstatic ELCIRC
- Version control
- Organization of web site

3D community ocean models

	ADCIRC[1]	POM[2]/ROMS[3]	FVCOM[4]	QUODDY[5]	UnTRIM[6]/ ELCIRC
Wetting and drying	2D only	No	Yes	No	Yes
Horizontal grid	Unstr.	Stru.	Unstr.	Unstr.	Unstr.
Vertical representation	σ-coord	σ-coord	σ-coord	σ-coord	z-coord
Numerical algorithm	FE	FD	FV	FE	FD/FV
Continuity wave or primitive equations	GWCE	PE	PE	GWCE	PE
Mode splitting	Yes	Yes	Yes	Yes	No
Advection treatment	Eul	Eul	Eul	Eul	ELM

1. Advanced Circulation, Luettich *et al.* (Univ. of UNC, Waterway Experiment Station of Army Corp of Engineers):
2. Princeton Ocean Model (POM) (Mellor and Blumberg)
3. Regional Ocean Modeling System, Haidvogel et al. (Rutgers Univ.)
4. Finite Volume Community Ocean Model, C. Chen (University of Massacusetts)
5. QUODDY, Lynch *et al.* (Dartmouth College)
6. Unstructured Tidal River Inter-tidal Mudflat, Casulli (Univ. of Trento, Italy)