Recent Developments of ROMS at UCLA

Alexander Shchepetkin
Jeroen Molemaker
Francois Colas
Evan Mason
Yusuke Uchiyama
Jaison Kurian
Jim C. McWilliams

Chicago, IL, June 2009

What do we do:

- grids, configurations, resolutions
- scientific goals
- downscaling
- process studies, submesosale dynamics
- sediment transport
- parameterizations

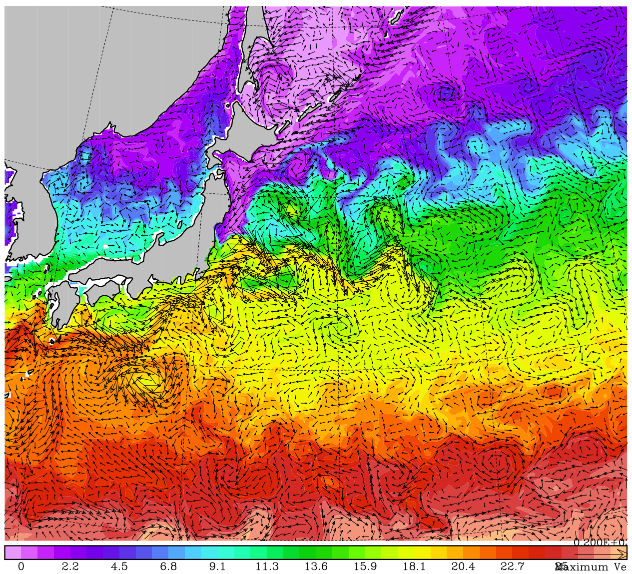
What does it take to make it happen:

- changing requirements for codes to meet scientific goal
- **changes in ROMS kernel algorithms**
- sub-models and parameterizations, interference with kernel
- changing modeling practices
- model logistics and tools: coping with large data
- computational performance: adaptation to multi-core CPUs

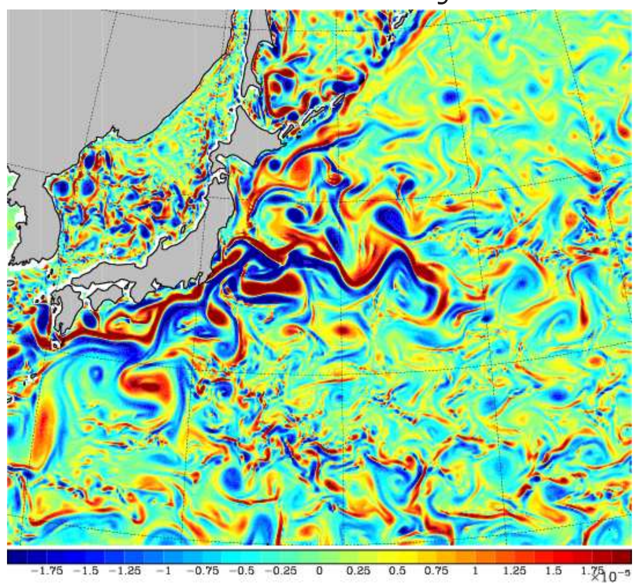
Pacific simulations

Grids: 0.45 deg; 0.22 deg at Equator, isotropic $\Delta\phi = \cos\phi \cdot \delta\lambda$, $976 \times 720 \times 40$
12km, isotropic, centered off Equator, $1840 \times 960 \times 32$
posed as a regional configuration: side boundaries from SODA (a global POP solution)

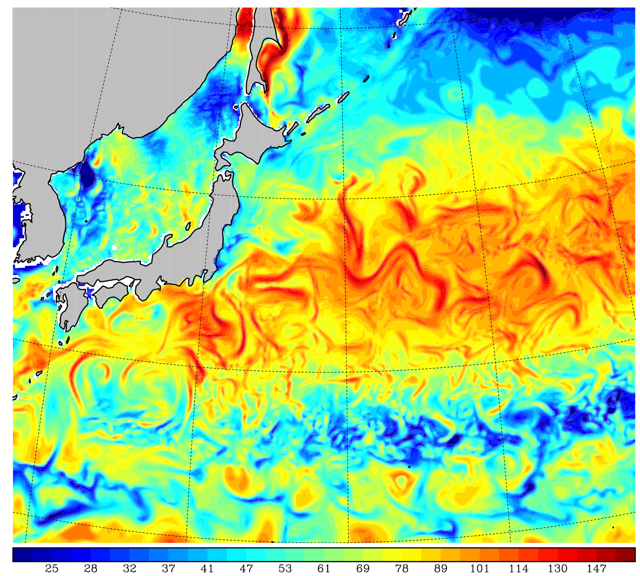
SST + surface velocity, 12km



surface vorticity



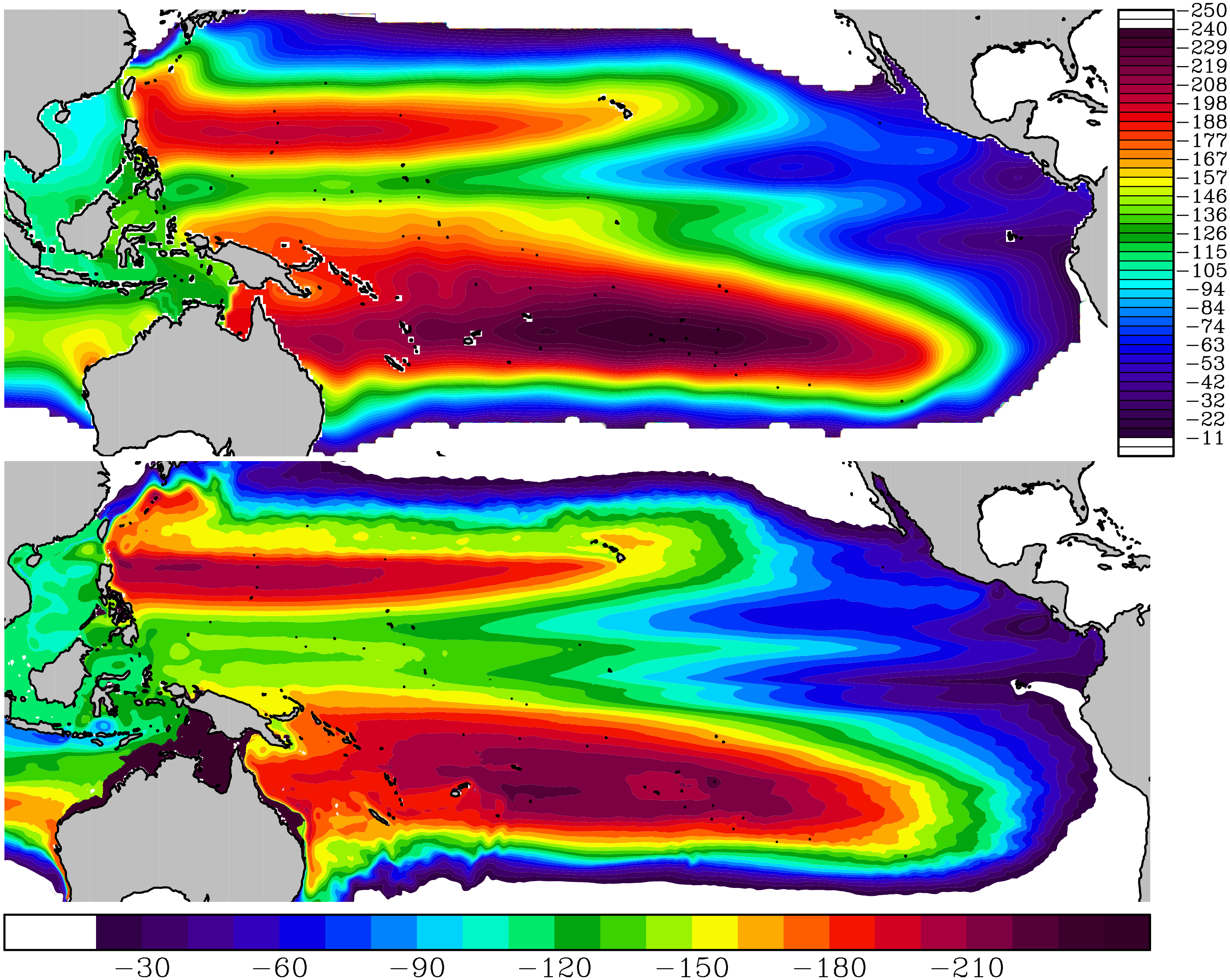
KPP PBL depth, December



Interests: large scale dynamics
focus vertical parameterization schemes, KPP
focus on long-term performance and conservation properties of the code
comparison of different wind/heat forcing products
air-sea interaction
modeling sensitivities and safe practices (mainly due to topographic effects)
standard test platform for Kernel algorithms (time stepping, coupling, parameterization)
generate side boundaries for other models

Pacific simulations, continued...

Depth of 20^0C isoterm, annual



Levitus

vs.

ROMS,

12km

32 levels

QCOW winds

Risien &

Chelton, 2008

SOC 1.1a

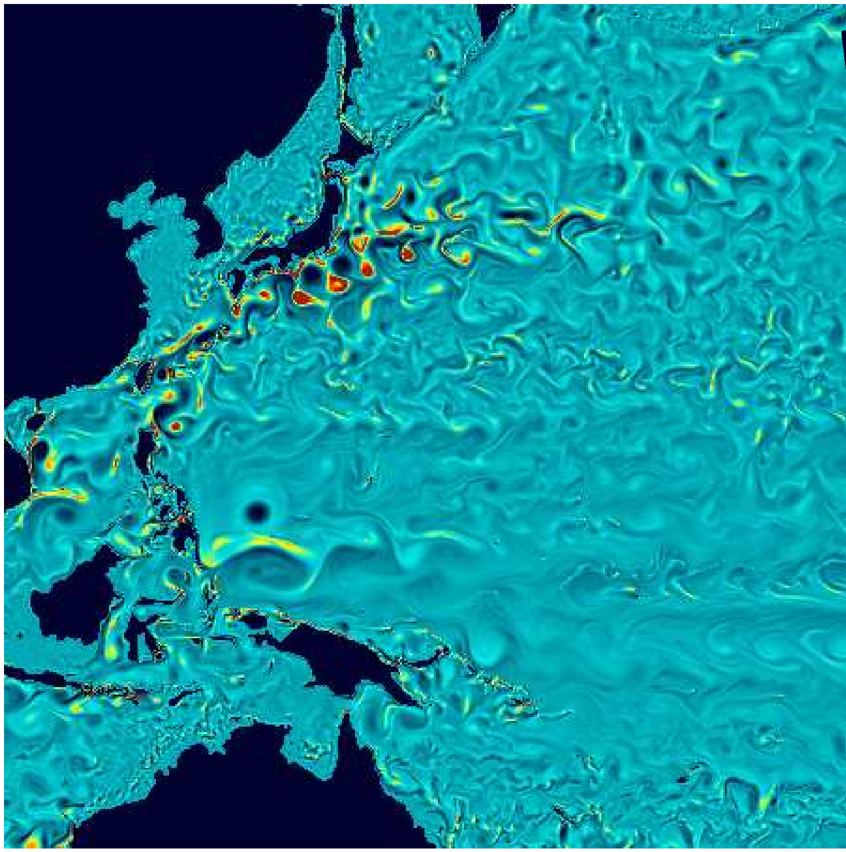
heat/freshwater

Grist & Josey,

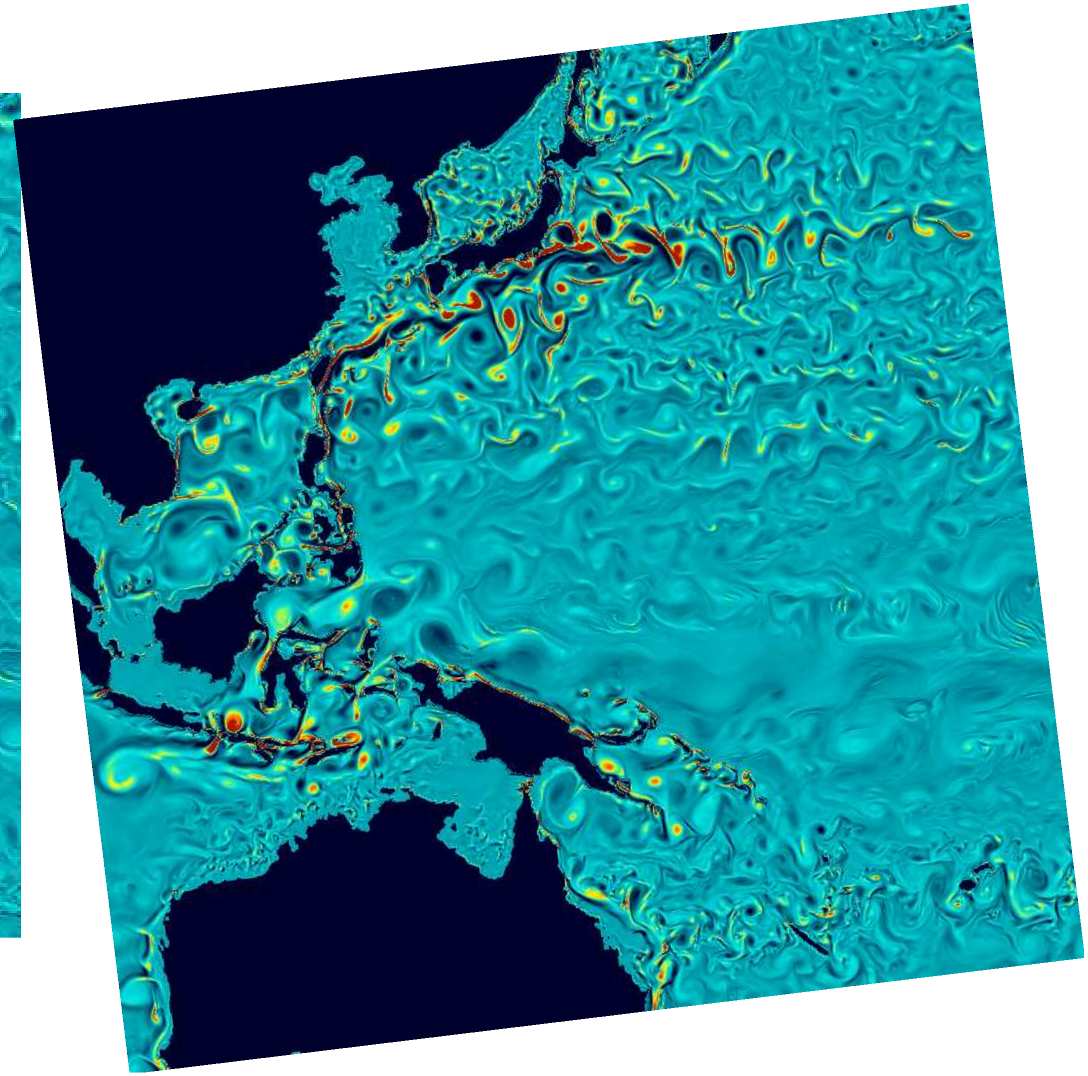
2003

Pacific simulations, continued...

Mesoscale, effect of resolution



0.22-degree

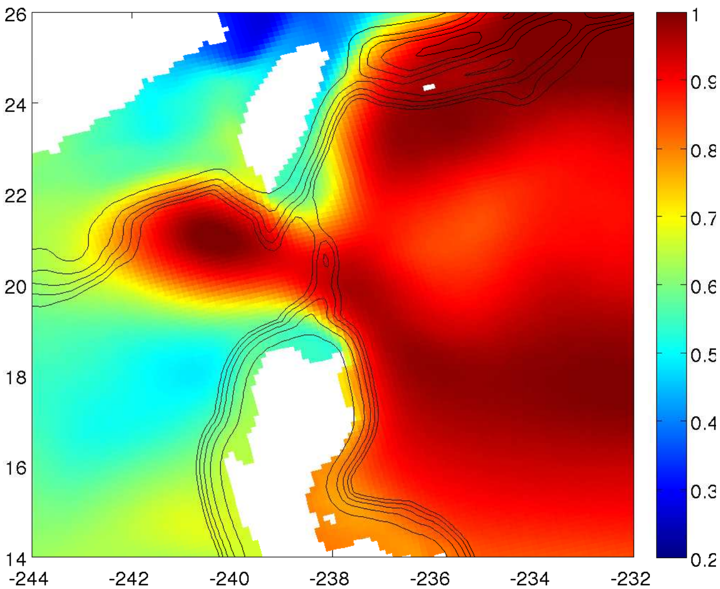


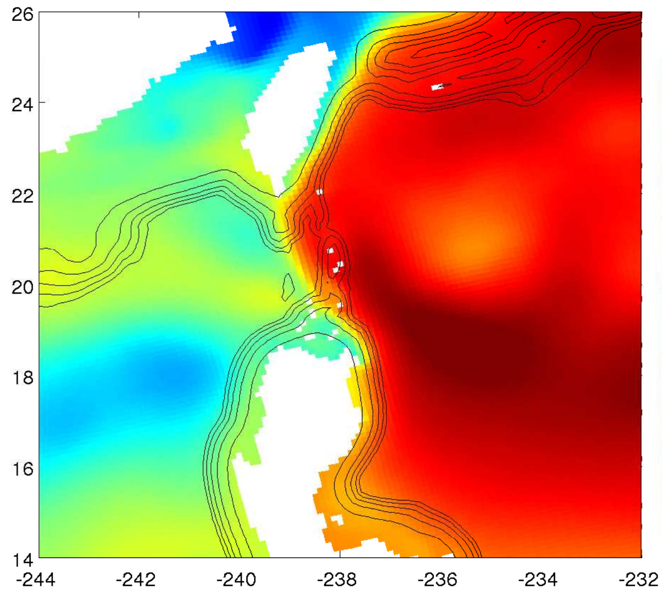
12 km

near-surface relative vorticity in Western Pacific
from ROMS Pacific simulations
min/max saturated colors correspond to $\pm 0.25 \times 10^{-4} sec^{-1}$

Pacific simulations, continued...

Topography and land mask sensitivity





Mean annual SSH from two 12km ROMS Pacific simulations, Luzon Strait

topographic contours at 400, 800, 1200, 1600, 2000 m

Note several extra 1-point islands on the right

- Comparable sensitivity was observed by Gille, Metzger, & Tokmakian (2004) in NLOM model
- in line with the experience of **not to neglect small islands**, even thought poorly resolved, e.g., Galapagos (barrier for Equatorial undercurrent, Eden & Timmermann, 2004): Carribean (Gulf Stream separation), Kuril, Aleutian, etc
- sigma-modelers tend to over-smooth topography, because of fear of Haney (1991) criterion. Our current practices lean toward $rx_1 \sim 10$

Luzon Strait map from `http://en.wikipedia.org/wiki/Luzon_Strait`

Pacific simulations, continued...

Coastal upwelling off Peru-Chile region extracted from 12km Pacific solution.

SST, middle of December

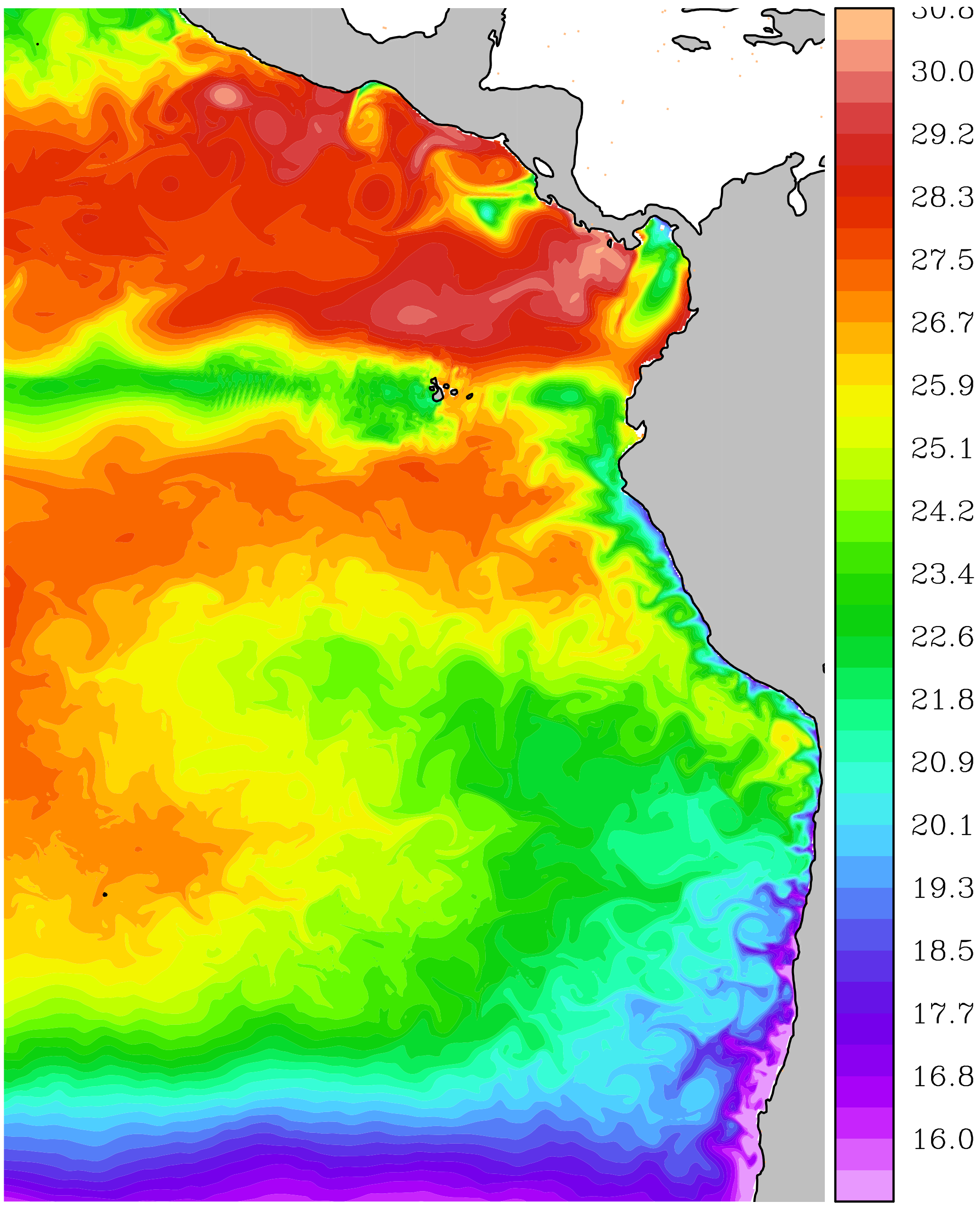
Interests:

air-sea interaction,
VOCALS project: coarse-resolution climate models lack this upwelling

sub-mesoscale activity
(possible in finer nested configuration)

gap wind events

...another "US West Coast" configuration, but with stronger Equatorial link



New US West Coast solutions

USW4: $\Delta x = 4$**km**
375 × 625
42 levels

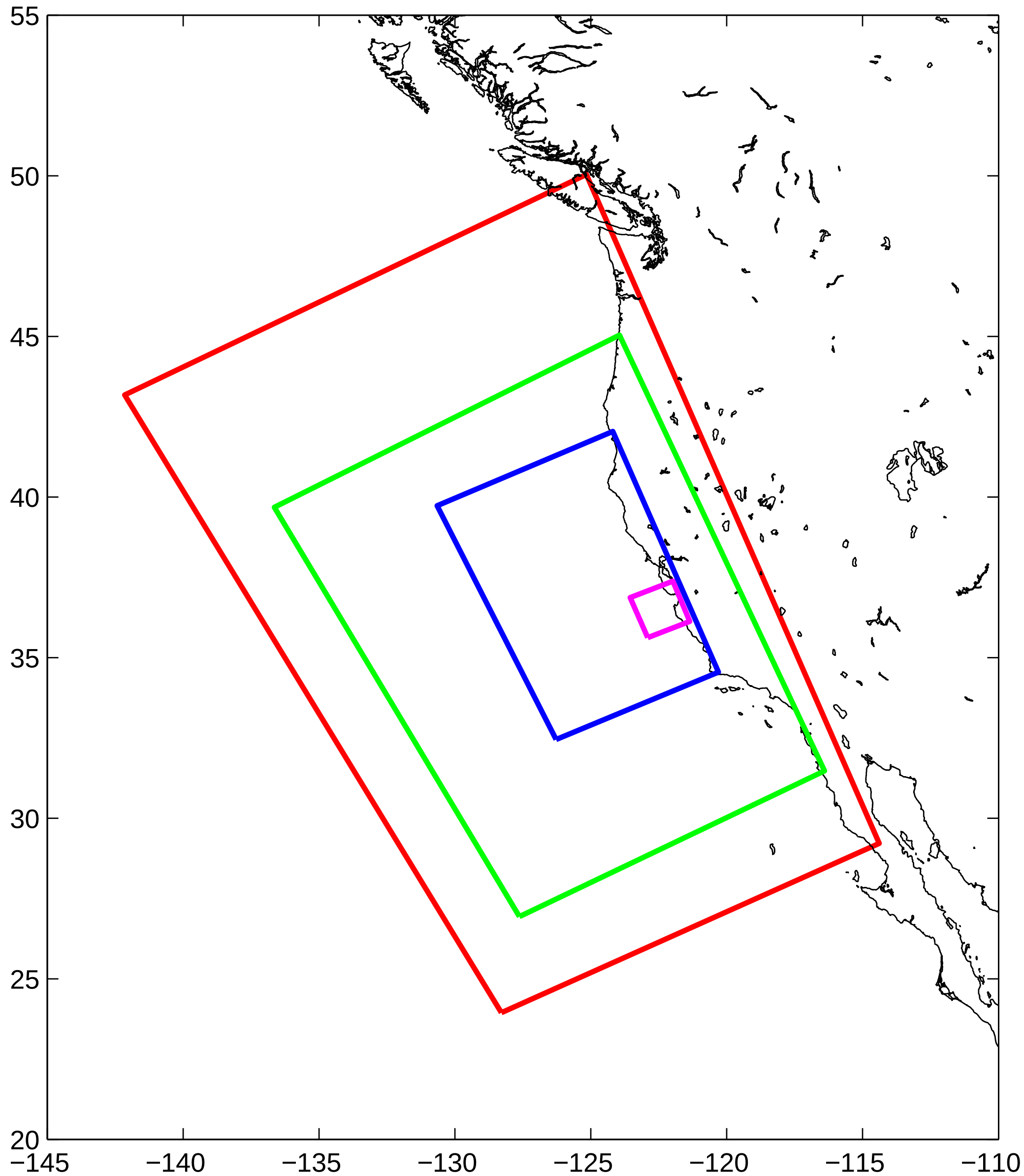
SMCAL: $\Delta x = 1.5$**km**
800 × 1100
42 levels

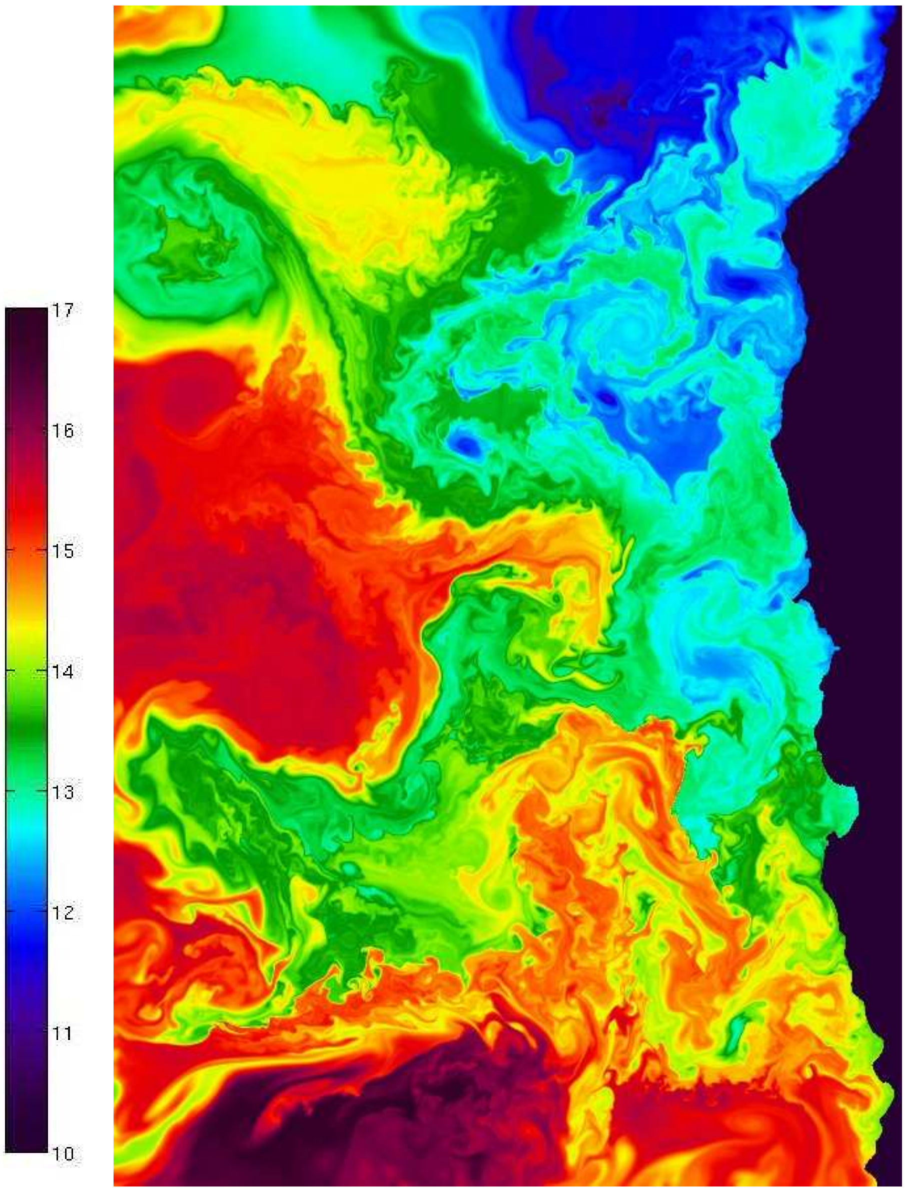
SMCC: $\Delta x = 0.5$**km**
1200 × 1800
60 levels

CUC: $\Delta x = 150$**m**
900 × 900
80 levels

goals and interests: submesoscale dynamics; process studies; downscaling techniques; interaction of flow with bottom topography; generate side-boundary conditions for other grids.

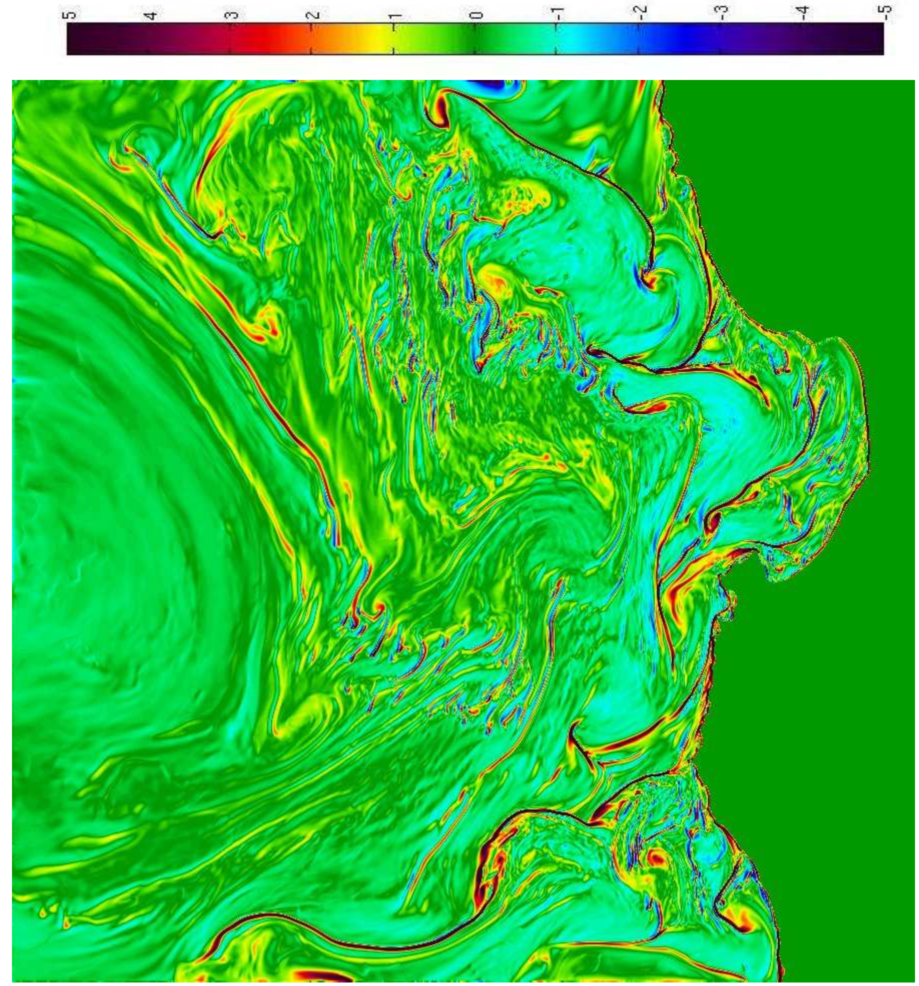
cf., Capet et. al., 2008, but now without idealization, and with updated codes, better techniques, and newer machinery.

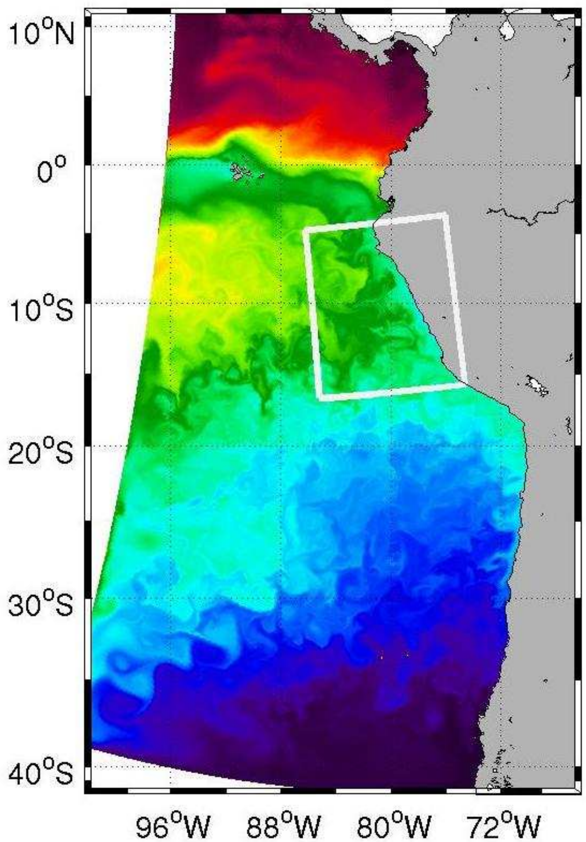




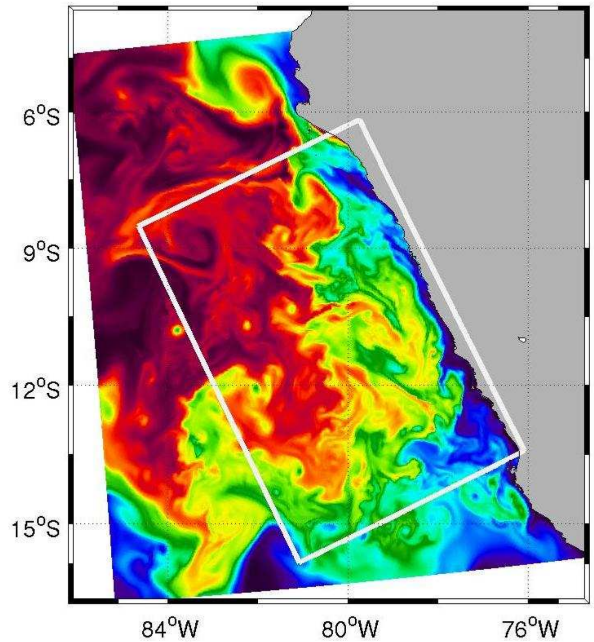
left: SMCC, $\Delta x = 0.5$km, SST

below: CUC, $\Delta x = 150$m, surface vorticity scaled by f



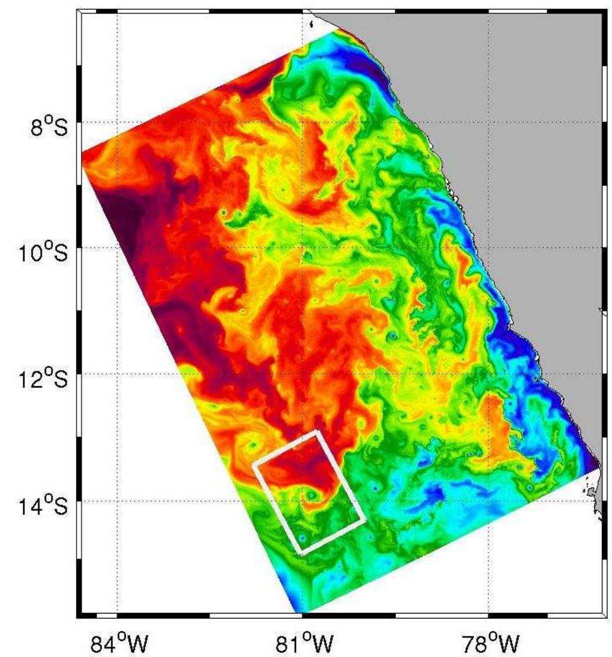


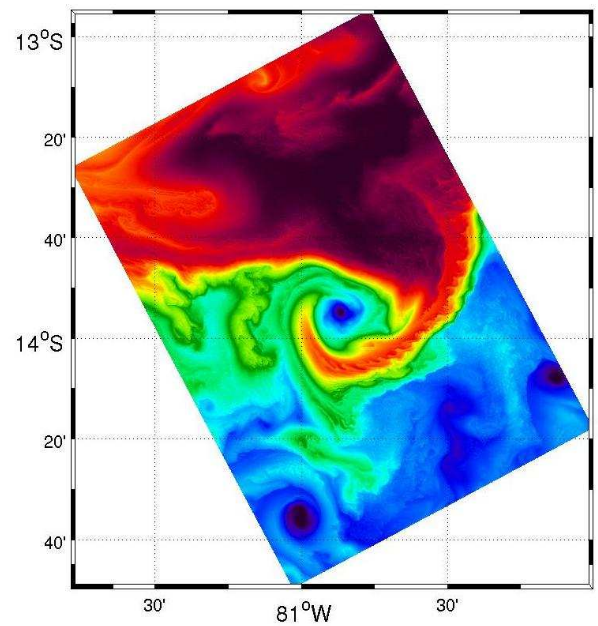
←7.5 km



2.2 km

500 m





180 m

Peru-Chile upwelling system: 4-stage downscaling

7.5 km	384 × 800 × 30	[1992-1999] POP/CCSM ERS COADS Pathfinder
2.2 km	468 × 560 × 30	[1994-1998] receives 3-day averages, outputs daily
500 m	1200 × 1800 × 42	[April 1995 - July 1996] receives daily, outputs every 12h
180 m	694 × 972 × 84	[1st-30th September 1995] receives 12h outputs every 3h

Peru-Chile continued...

Spiral eddy

$$\Delta x = 180m$$

top left: SST

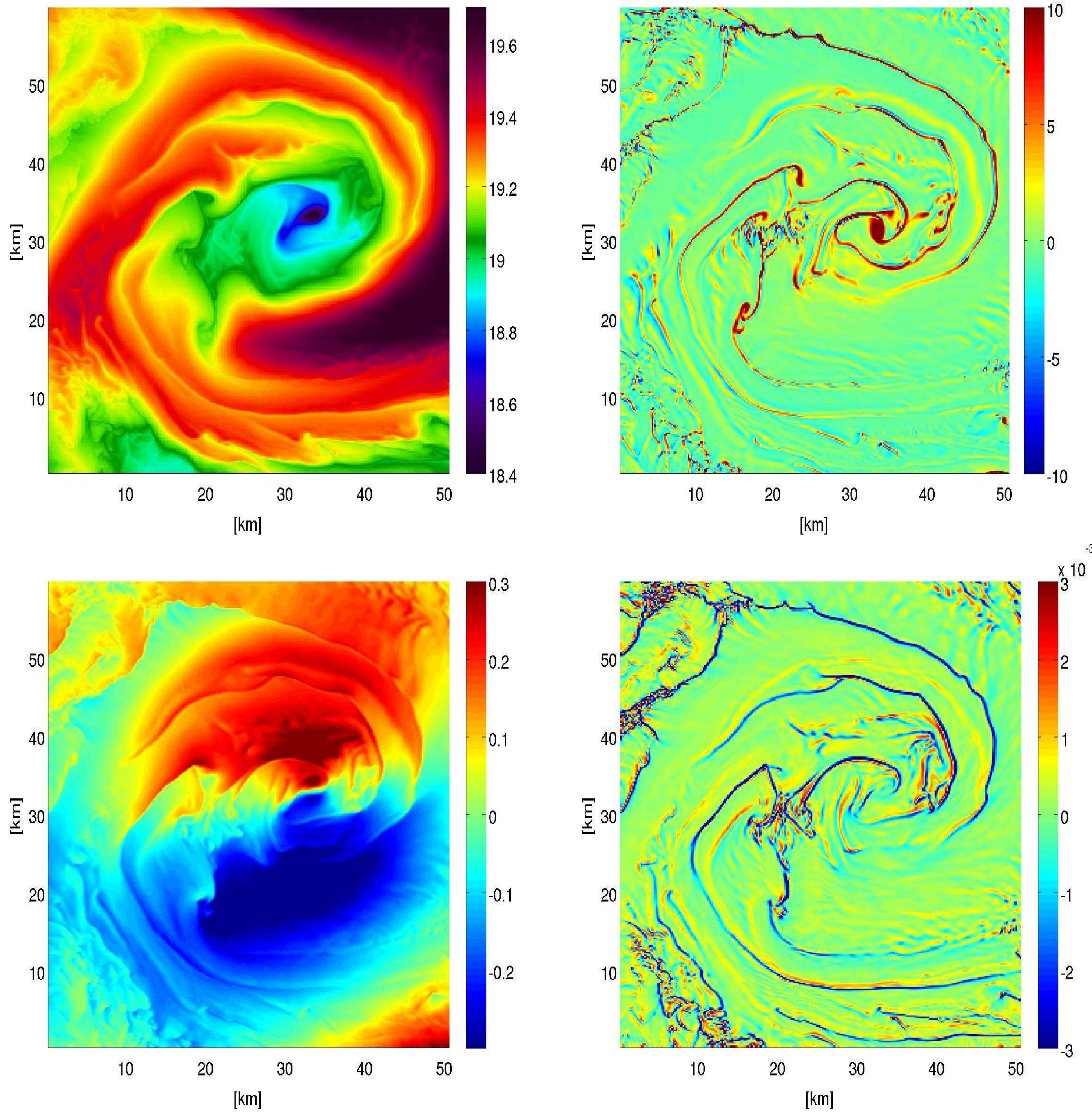
right: $\frac{\text{vorticity}}{f}$

bottom left: **u**

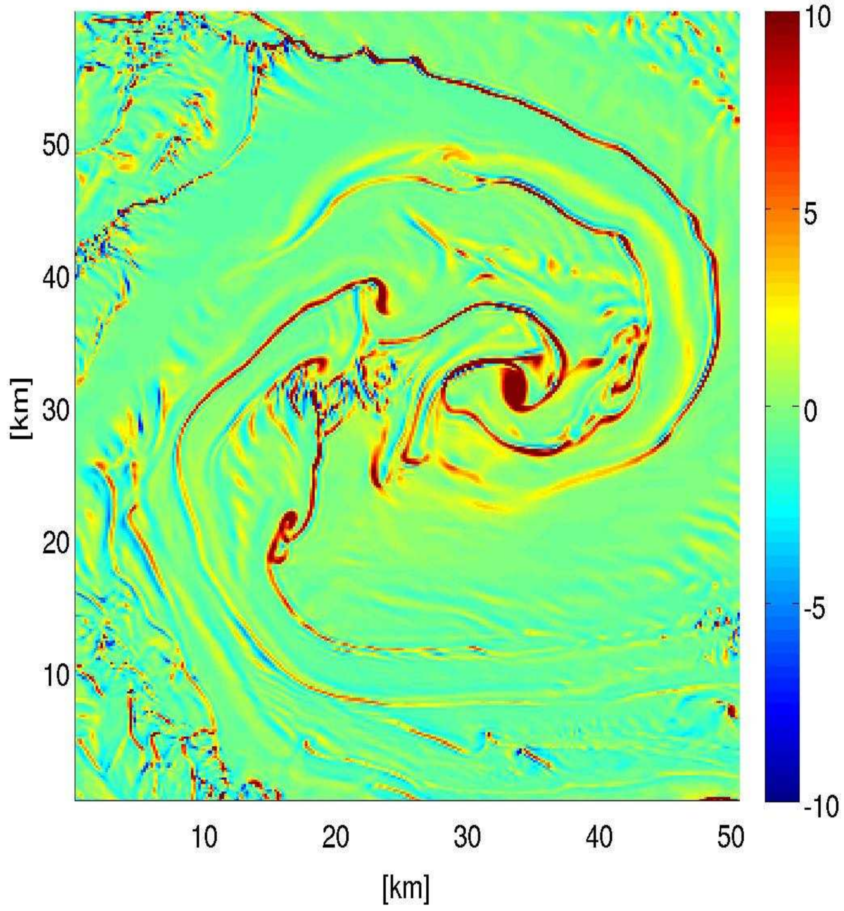
right: **w** at -10m

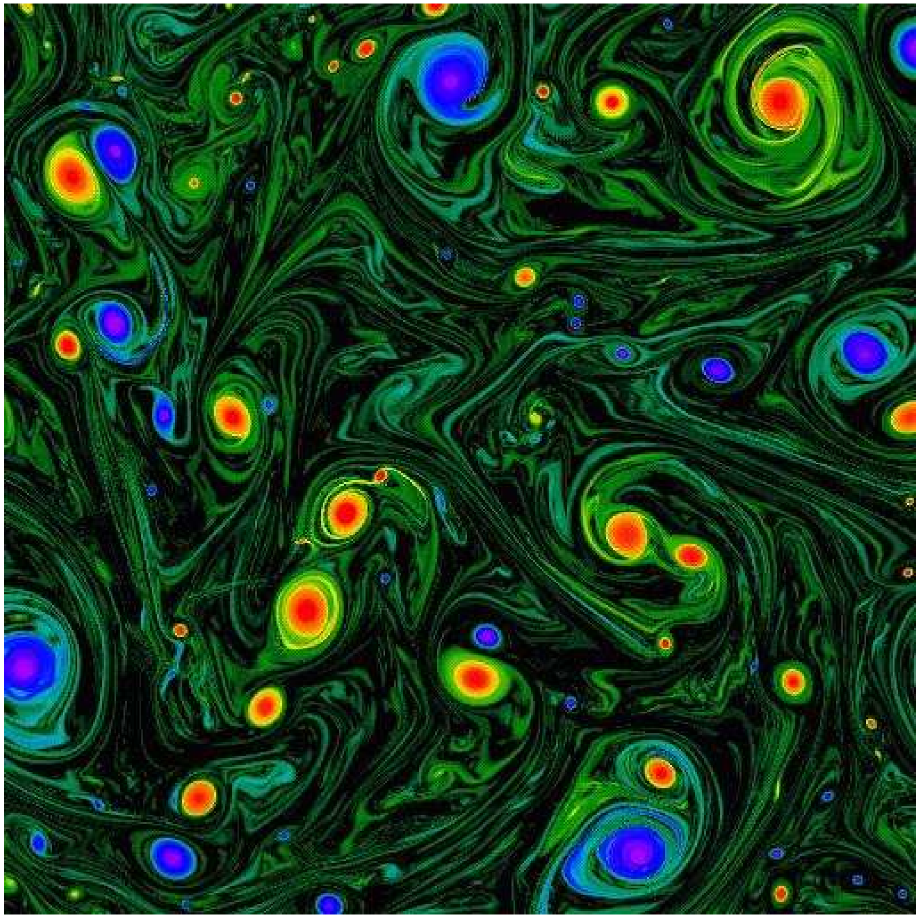
cyclones only

southern hemisphere



Peru-Chile continued...





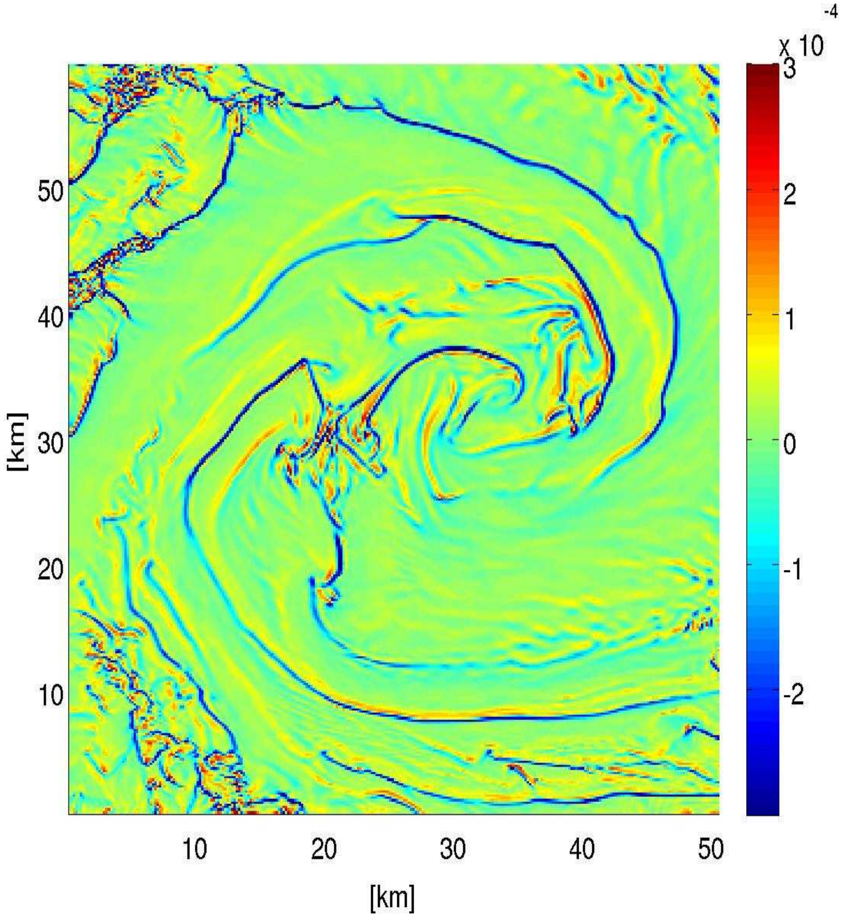
spiral eddy, **vorticity/f**

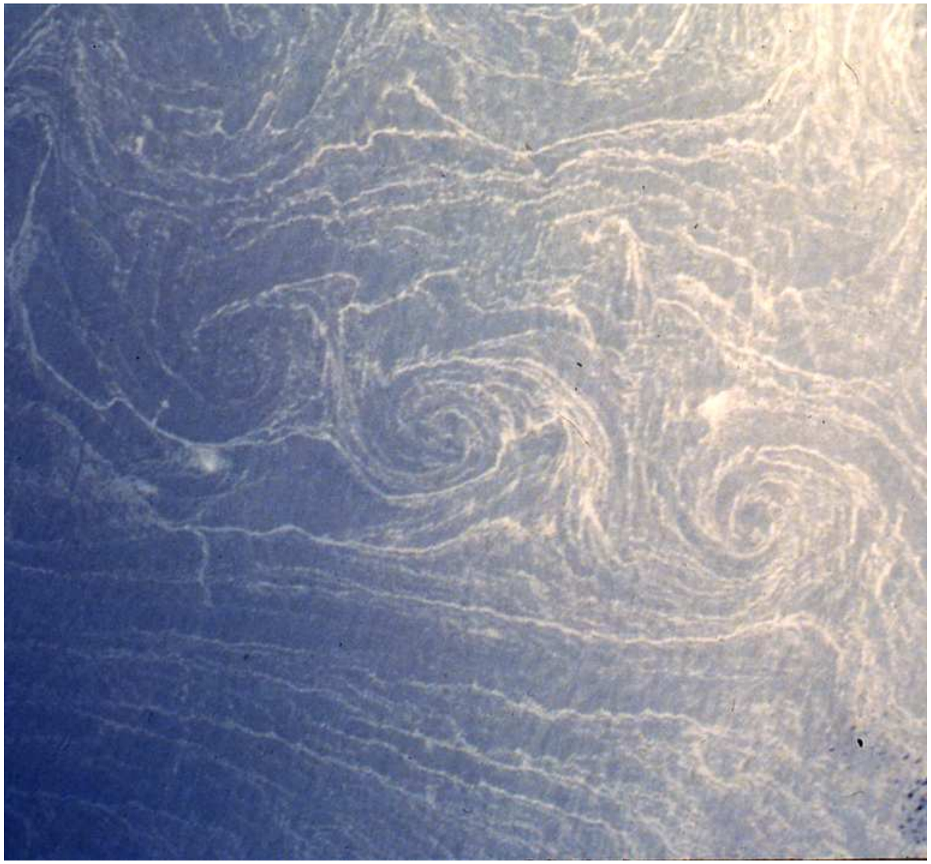
cyclones only (southern hemisphere)

2D freely decaying turbulence

1024×1024-grid 6th-order+ELAD dissipation

Peru-Chile continued...





spiral eddy, **w** at 10 m
180m grid

Mediterranean Sea, Shear Wall Spiral Eddies

STS-41G, October 1984. Picture #17-35-094

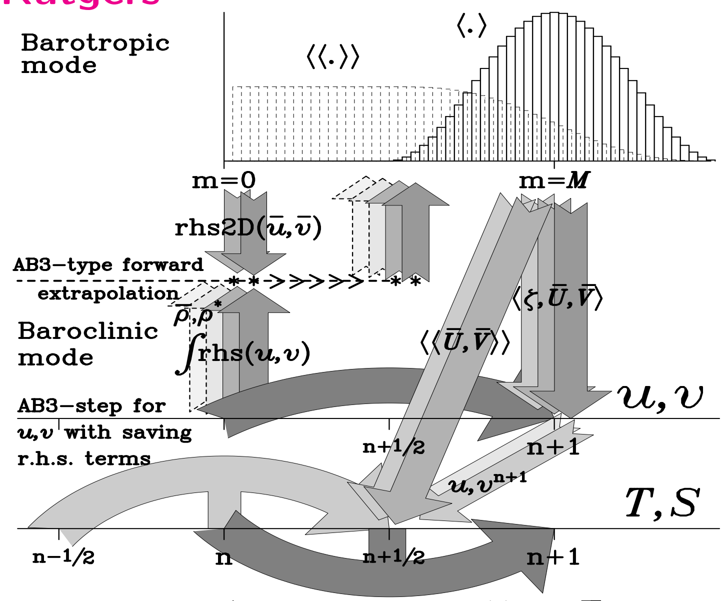
image on the right from

http://www.lpi.usra.edu/publications/slidesets/oceans/oceanviews/slide_03.html

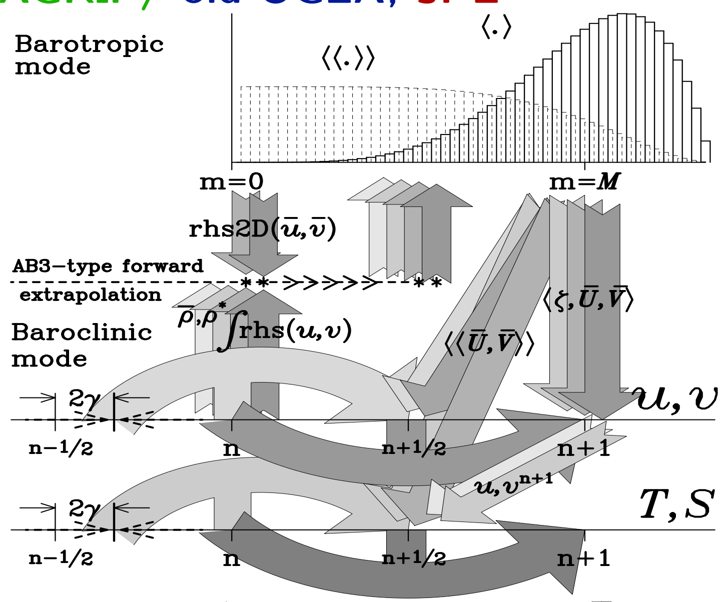
What does it take to make it happen

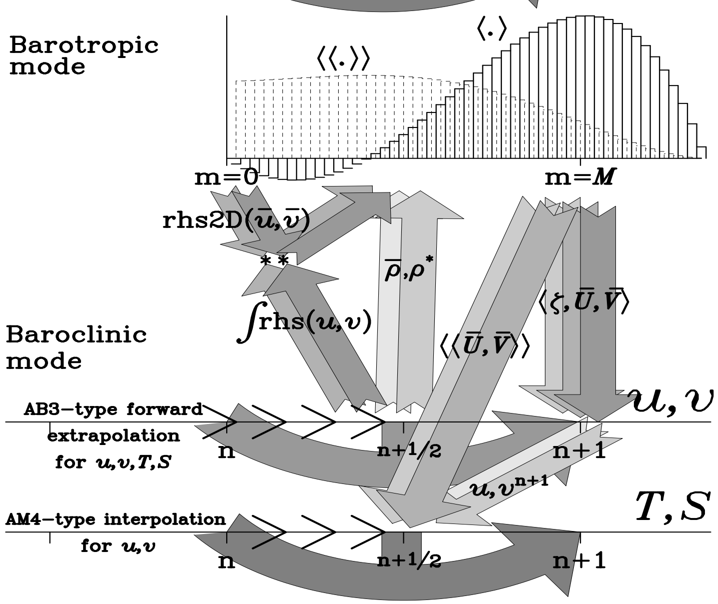
Status Quo: Four variants of ROMS kernel in existence

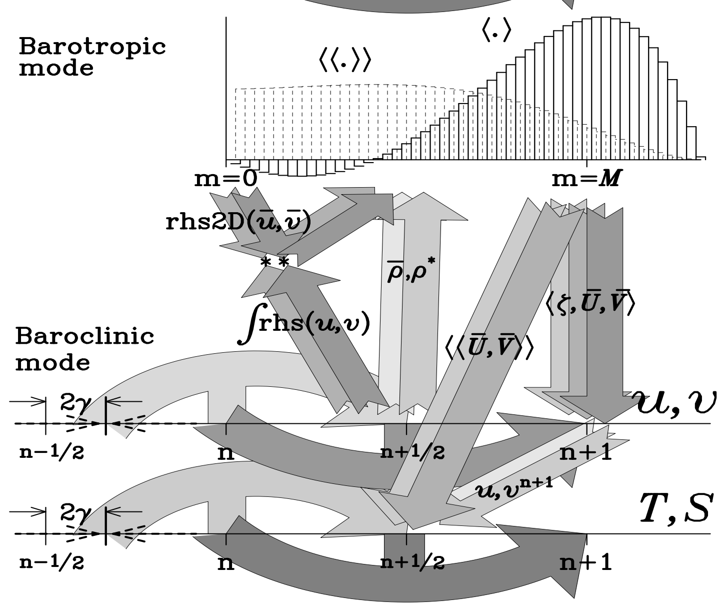
Rutgers



AGRIF/ old UCLA; JPL







Non-hydrostatic code prototype

UCLA (current)

Algorithmic Features of ROMS

- **vertical coordinate:** loosely $\in$ σ-class models, but the code stores $z(x,y,s)$ as an array $\Rightarrow$ a rather general vertical coordinate. Currently settling for "z-sigma":

$$z = z^{(0)} + \zeta\left(1 + \frac{z^0}{h}\right) \qquad z^{(0)} = h \cdot \frac{h_c \cdot s + h \cdot C(s)}{h + h_c} \qquad h = h(x,y) \qquad h_c = \text{const}$$

$$C(s) \equiv C[S(s)] \quad \text{where} \quad S(s) = \frac{1 - \cosh(\theta_s s)}{\cosh(\theta_s) - 1} \qquad C(S) = \frac{\exp(\theta_b (S+1)) - 1}{\exp(\theta_b) - 1} - 1$$

note $\lim_{s \to 0} \frac{\partial C(s)}{\partial s} = 0$ and **there is no restriction $h_c < h_{\min}$!**

- orthogonal curvilinear grid in horizontal directions

- **time stepping engine:** free-surface, split-explicit, barotropic-baroclinic mode splitting; built around new time-stepping algorithms for hyperbolic system equations; always use forward-backward feedback; **exact** finite-time-step, finite-volume conservation and constancy for tracers; **higher-than-second-order** accuracy for critical terms: advection, pressure-gradient, etc; **exact restart capability**

- Boussinesq approximation, with EOS stiffening (not all)

- intended for limited-area modeling $\Rightarrow$ focus on open boundary conditions; grid nesting, `RomsTools` (more than one branch)

- **adjoint**

- coupled with sub-models (biology, sediment transport, wave effects, etc...)

- non-hydrostatic extension

- parallel via 2D domain decomposition: threaded/OpenMP, or MPI, or both; multiple architecture support; **exact, verifiable single/multi CPU matching**

- *poor man's* computing, ground-up design philosophy, focusing on inter-component algorithm interference; code infrastructure is distinct from *modular* (like in MOM/POP) design

- code architecture decisions involve optimization in multidimensional space, including model physics, numerical algorithms, computational performance and cost

- loose, but talking to each other community >10 years

- inter-modeling communication

Selected topics:

- Anomalously stable modified RK2 stepping for wave system
- Updated nesting techniques in ROMS
- Implicit bottom drag ... at last

Modified RK2 algorithm of SM2005

$\partial_t \zeta = -\omega \cdot u \qquad \partial_t u = -\omega \cdot \zeta \qquad \alpha \equiv \omega \Delta t$

predictor

$$\zeta^{n+1,*} = \zeta^n - i\alpha \cdot u^n$$
$$u^{n+1,*} = u^n - i\alpha \cdot \left[\beta \zeta^{n+1,*} + (1-\beta)\zeta^n\right]$$

corrector

$$\zeta^{n+1} = \zeta^n - \frac{i\alpha}{2} \cdot \left(u^{n+1,*} + u^n\right)$$
$$u^{n+1} = u^n - \frac{i\alpha}{2} \cdot \left[\epsilon \zeta^{n+1} + (1-\epsilon)\zeta^{n+1,*} + \zeta^n\right]$$

stable, the original RK2 is weakly unstable for non-dissipative wave system

can be made **third-order** accurate for phase speed

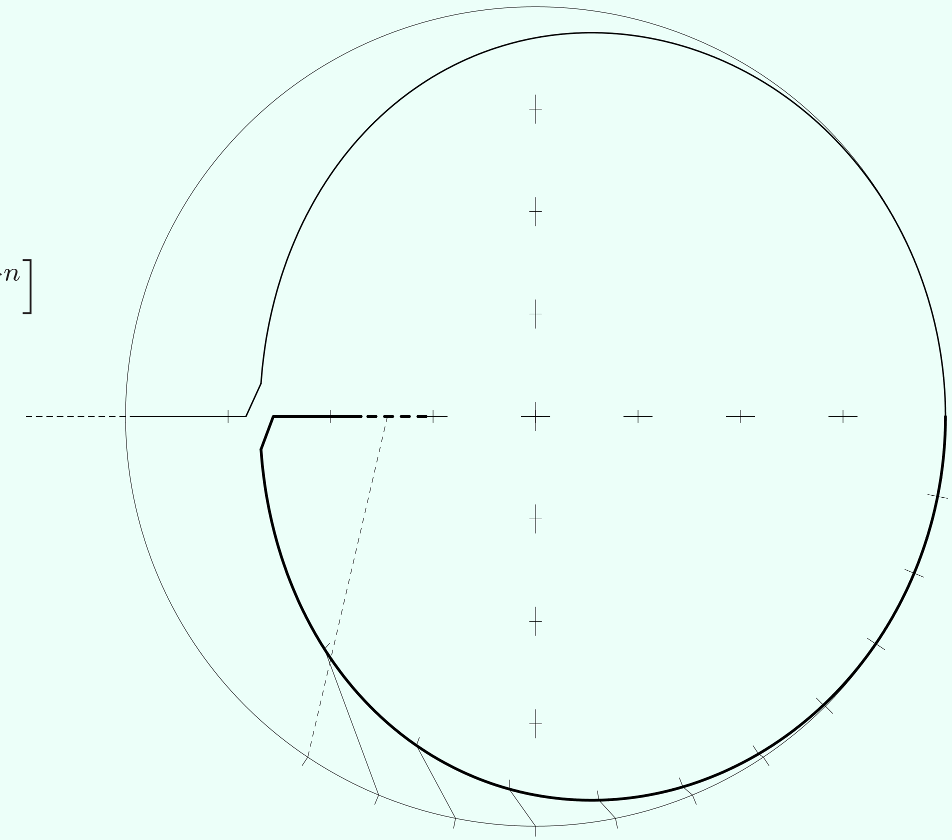
the best choice of coefficients $\beta = 1/3 \quad \epsilon = 2/3$

$\alpha_{\text{max}} = \sqrt{6\left(3 - \sqrt{5}\right)} = 2.14093$

the best accuracy β, ϵ settings also lead to maximum stability

used for startup in ROMS kernels

twice as much work relatively to the classical Forward-Backward scheme, but with only insignificant gain in α_{max}



$\beta = 1/3 \qquad \epsilon = 2/3 \qquad \alpha_{\text{max}} = 2.14093$

Rueda, Sanmiguel-Rojas, and Hodges (2007)

predictor

$$\zeta^{n+1,*} = \zeta^n - i\alpha\cdot u^n$$
$$u^{n+1,*} = u^n - i\alpha\cdot\left[\beta\zeta^{n+1,*} + (1-\beta)\zeta^n\right]$$

corrector

$$\zeta^{n+1} = \zeta^n - i\alpha\cdot\left[\gamma u^{n+1,*} + (1-\gamma)u^n\right]$$
$$u^{n+1} = u^n - i\alpha\cdot\left[\theta\zeta^{n+1} + (1-\theta)\zeta^n\right]$$

driven by desire to make TRIM of Casulli & Cheng (1992) stable with respect to baroclinic waves

"balance rule" $\gamma+\theta\equiv 1$ must be respected to maintain second-order accuracy

can be made **third-order** accurate

the best choice $\beta=1/6,\quad \gamma=\theta=1/2$

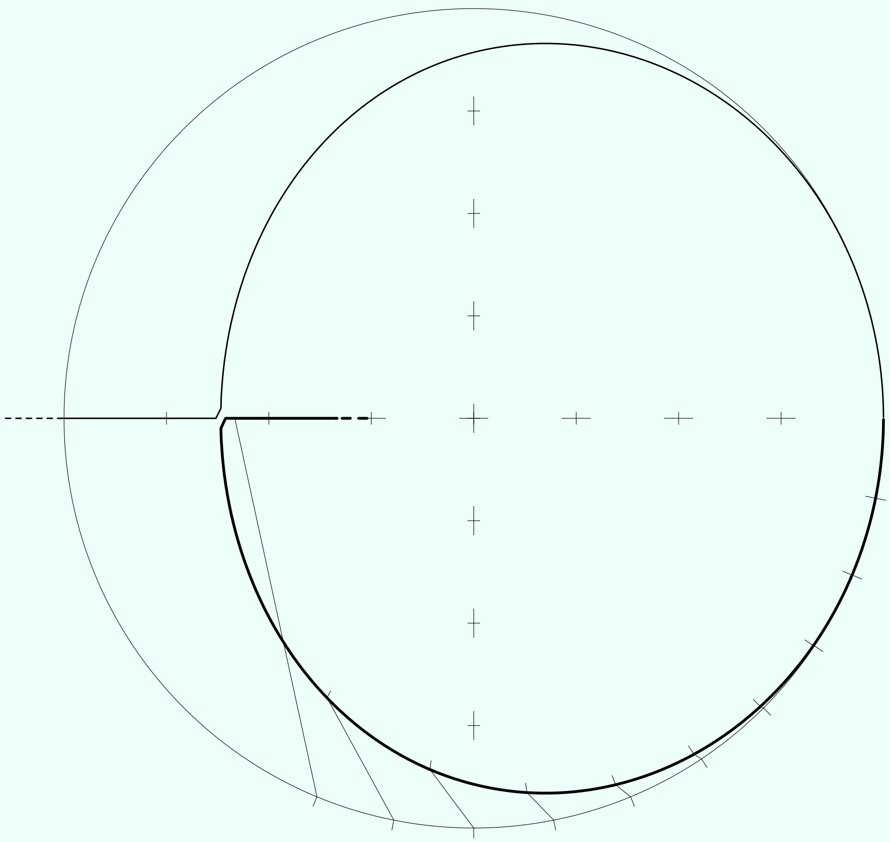
$\alpha_{\mathrm{max}}=2$ independent of β

[same as $\theta_{\mathrm{p}}=1/6$, $\theta_{\mathrm{b}}=\theta=1/2$ in their original notation; inspired by "theta"-method of Casulli ⇒ use of θ-s for all coefficients. Also somewhat by SM2005]

does not revert to the standard RK2 if $\beta=0$, $\gamma=\theta=1/2$, but becomes Higdon (2002) instead

intersects SM2005 if $\epsilon\equiv 1$ there, $\gamma=\theta=1/2$

...just another modified Runge-Kutta scheme



$\beta=1/6,\quad \gamma=\theta=1/2,\qquad \alpha_{\mathrm{max}}=2$

Temptation: For all "sane" choice of coefficients the range of stability in **both cases** is limited by one of the branches leaving the unit circle along negative real axis.

However, at some point **after** the scheme is becomes unstable ($\alpha > \alpha_{\mathsf{max}}$) the branch **reverses its direction and re-enters** the unit circle.

Can the point of reversal be brought into the unit circle?

The answer is **NO for both** SM2005 and Rueda et. al. (2007): there are *not enough free coefficients left to play with.*

But, what if combine them ...

mod. RK2 of SM2005 + Rueda et. al. (2007) combined

predictor

$$\begin{aligned} \zeta^{n+1,*} &= \zeta^n - i\alpha \cdot u^n \\ u^{n+1,*} &= u^n - i\alpha \cdot \left[\beta\zeta^{n+1,*} + (1-\beta)\zeta^n\right] \end{aligned}$$

corrector

$$\begin{aligned} \zeta^{n+1} &= \zeta^n - i\alpha \cdot \left[(1-\theta)u^{n+1,*} + \theta u^n\right] \\ u^{n+1} &= u^n - i\alpha \cdot \left[\theta\left(\epsilon\zeta^{n+1} + (1-\epsilon)\zeta^{n+1,*}\right) + (1-\theta)\zeta^n\right] \end{aligned}$$

γ is replaced by $1/2-\theta$ because of "balance rule" $\Rightarrow$ at least second-order $\forall\ \beta, \theta, \epsilon$

Characteristic equation

$$\lambda^2 - \lambda\left[2 - \alpha^2 + \alpha^4 A\right] + 1 - \alpha^4 B = 0 \qquad \text{where} \qquad \begin{cases} A = \beta\epsilon\theta(1-\theta) \\ B = (1-\theta)(\beta - \beta\epsilon\theta + \epsilon\theta - \theta) \end{cases}$$

substitute $\lambda = e^{\pm i\alpha}$ and Taylor expansion

$$\alpha^4\left(\frac{1}{12} - A - B\right) \pm i\alpha^5 B + \alpha^6\left(\frac{B}{2} - \frac{1}{360}\right) + \mathcal{O}\left(\alpha^7\right) = 0$$

third-order accuracy $A + B = \frac{1}{12}$ or $\epsilon = 1 + \frac{1}{12\,\theta(1-\theta)} - \frac{\beta}{\theta}$ still leaves β, θ free

mod. RK2 of SM2005 + Rueda et. al. (2007) continued ...

Eliminating ϵ leaves us with

$$\left.\begin{aligned} A &= (1-\theta)\left(C^2 - (\beta - C)^2\right) \\ B &= (1-\theta)\left((\beta - C)^2 - C^2 + \frac{1}{12(1-\theta)}\right) \end{aligned}\right\} \quad \text{where} \quad C = \frac{\theta}{2} + \frac{1}{24(1-\theta)}$$

Can eliminate B as well (fourth-order accuracy), but it is actually a bad idea ...

Instability occurs when leaving unit circle through $\lambda = \pm 1$, so substitute it into Char. Eqn.

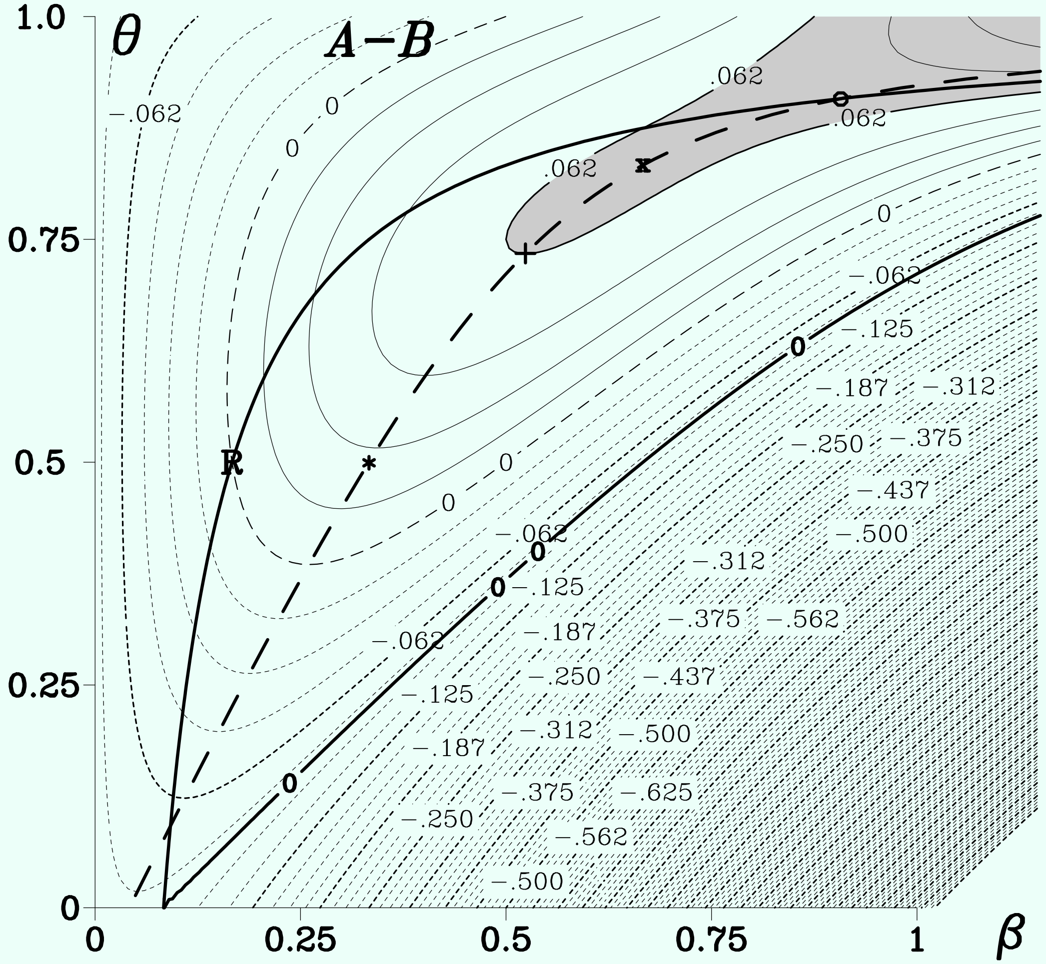
$$\lambda = -1 \quad : \qquad 4 - \alpha^2 + \alpha^4 (A - B) = 0$$

$$\lambda = +1 \quad : \qquad \alpha^2 \left[1 - \alpha^2 (A + B)\right] = 0$$

- Case of $\lambda = -1$ yields $\alpha^2_{\text{max}} = \left(1 \pm \sqrt{1 + 16(A-B)}\right) / [2(A-B)]$ where the sign $\pm$ must be chosen to be the same as the sign of $(A - B)$. The solution **exists only if** $A - B < 1/16$. As $A - B \to 1/16$, then $\alpha_{\text{max}} \to \sqrt{8}$, which is the largest stability limit when this limitation applies.
  [Note that $\alpha_{\text{max}} = 2$ in the case of $A - B = 0$, and changes continuously when $A - B$ changes sign.]
- Case $\lambda = +1$ yields $\alpha^2_{\text{max}} = 1/(A + B)$, which with leads to a **less restrictive $\alpha_{\text{max}} = \sqrt{12}$ for the entire subset of third-order algorithms.**

mod. RK2 of SM2005 + Rueda et. al. (2007) continued ...

...so it is all about $A - B = A(\beta, \theta) - B(\beta, \theta)$



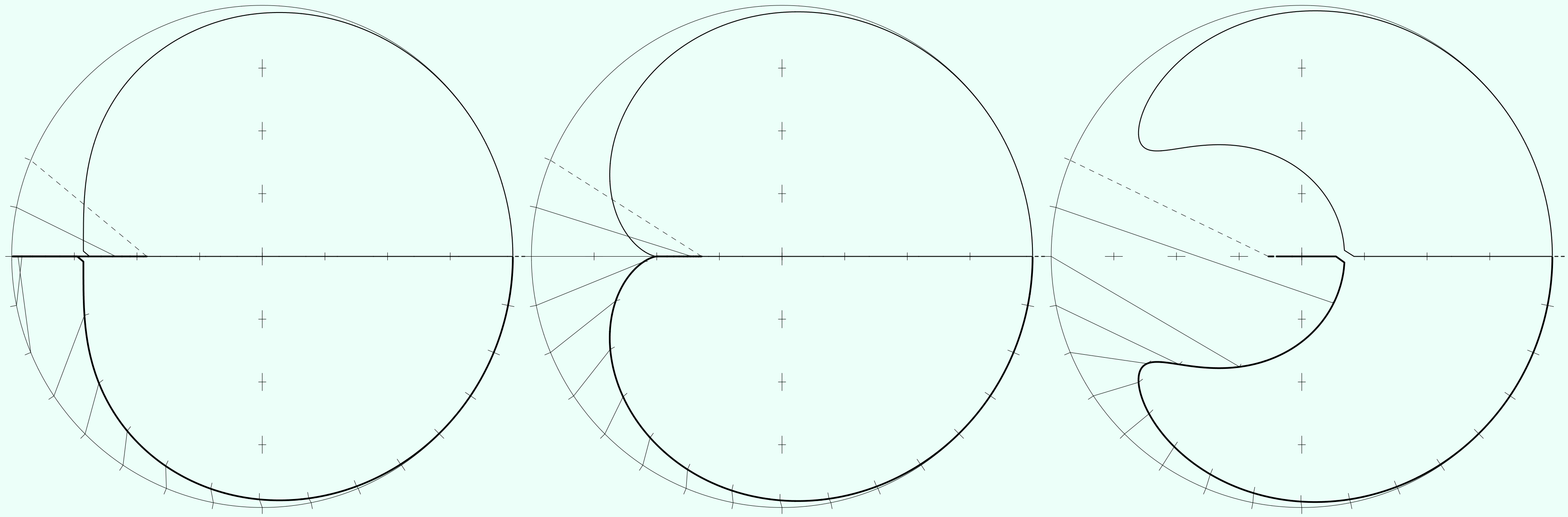
shaded is the region where $\lambda \to -1$ branch does not leave the unit circle

R corresponds to Rueda et. al. (2007), ***** to SM2005;

solid bold curves $\epsilon = 0$ *upper* and $\epsilon = 0$ *lower*; **dashed bold** minimum dissipation curve

mod. RK2 of SM2005 + Rueda et. al. (2007) continued ...

Examples of anomalously stable mod. RK2 time stepping



$\theta = 0.734$
$\beta = 0.523641604$
$\epsilon = 0.71340818$
[just entered shaded area]

$\theta = \frac{5}{6}\ \beta = \frac{2}{3}\ \epsilon = \frac{4}{5}$

$\theta = \beta = \frac{1}{2} + \sqrt{\frac{1}{6}} \approx 0.9082482$

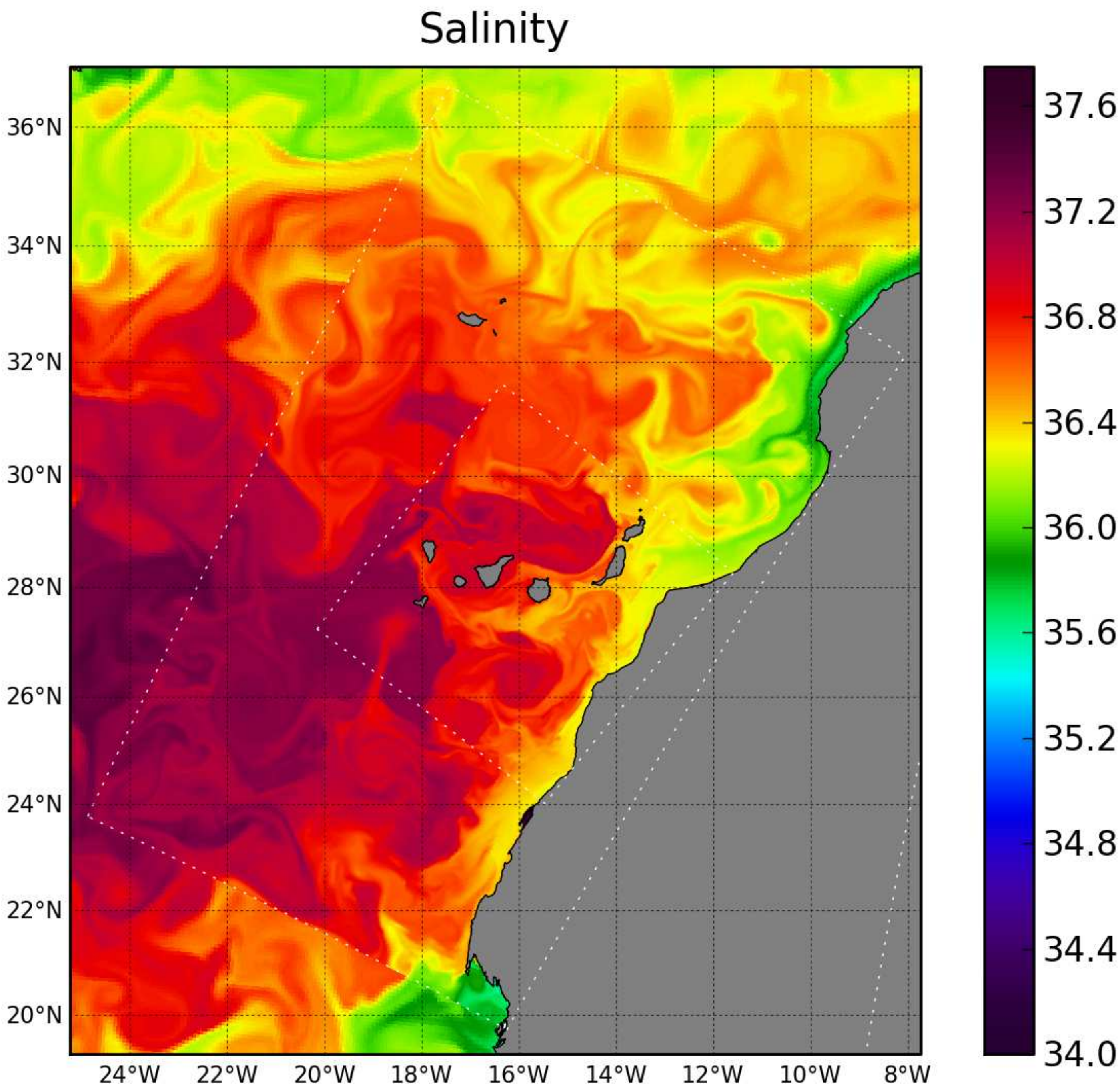
$\epsilon = 1$
[Rueda et.al.(2007) can be tuned to this without changing code]

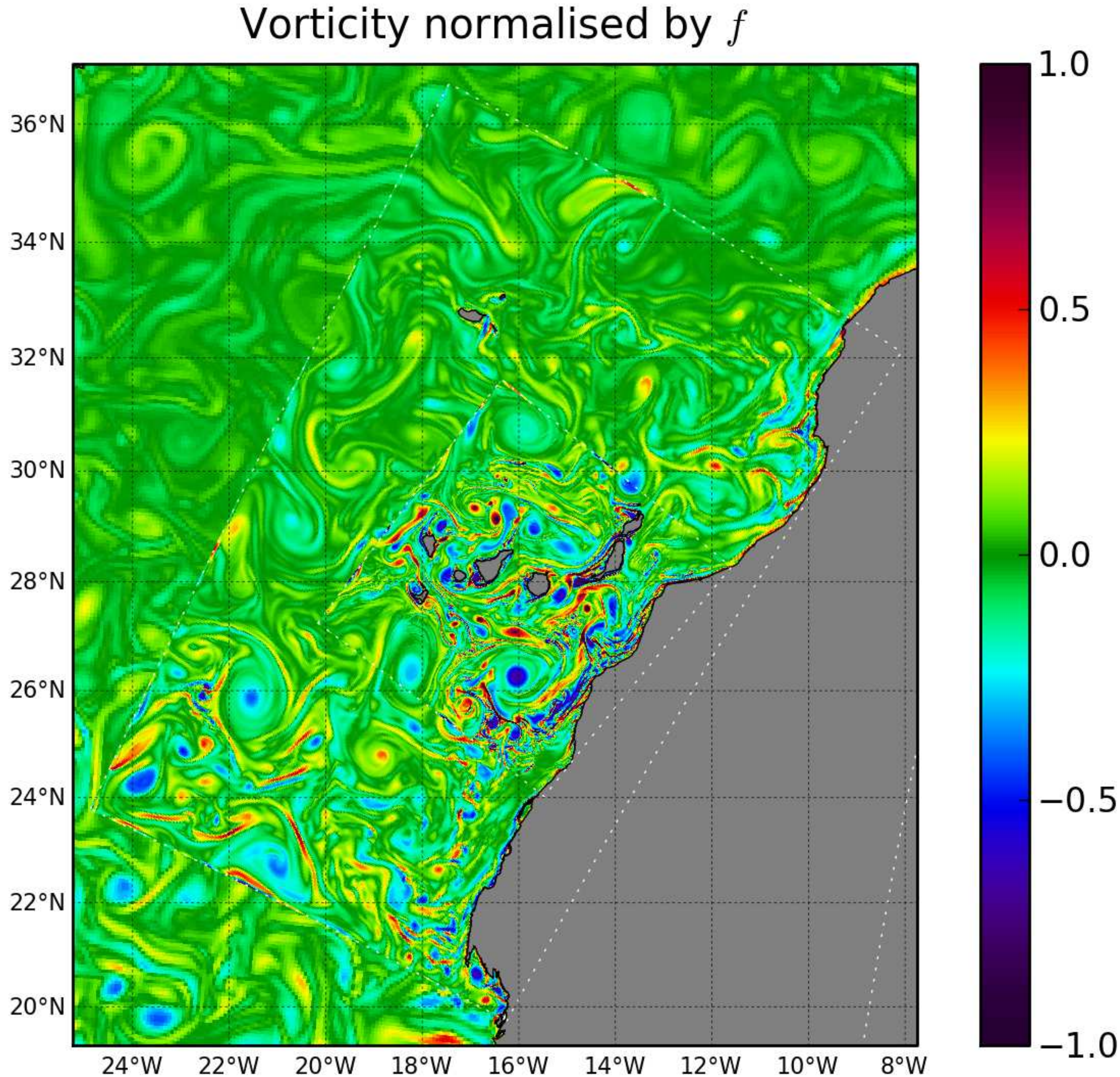
$\alpha_{\max} = 2\sqrt{3} \approx 3.4641$ in all cases

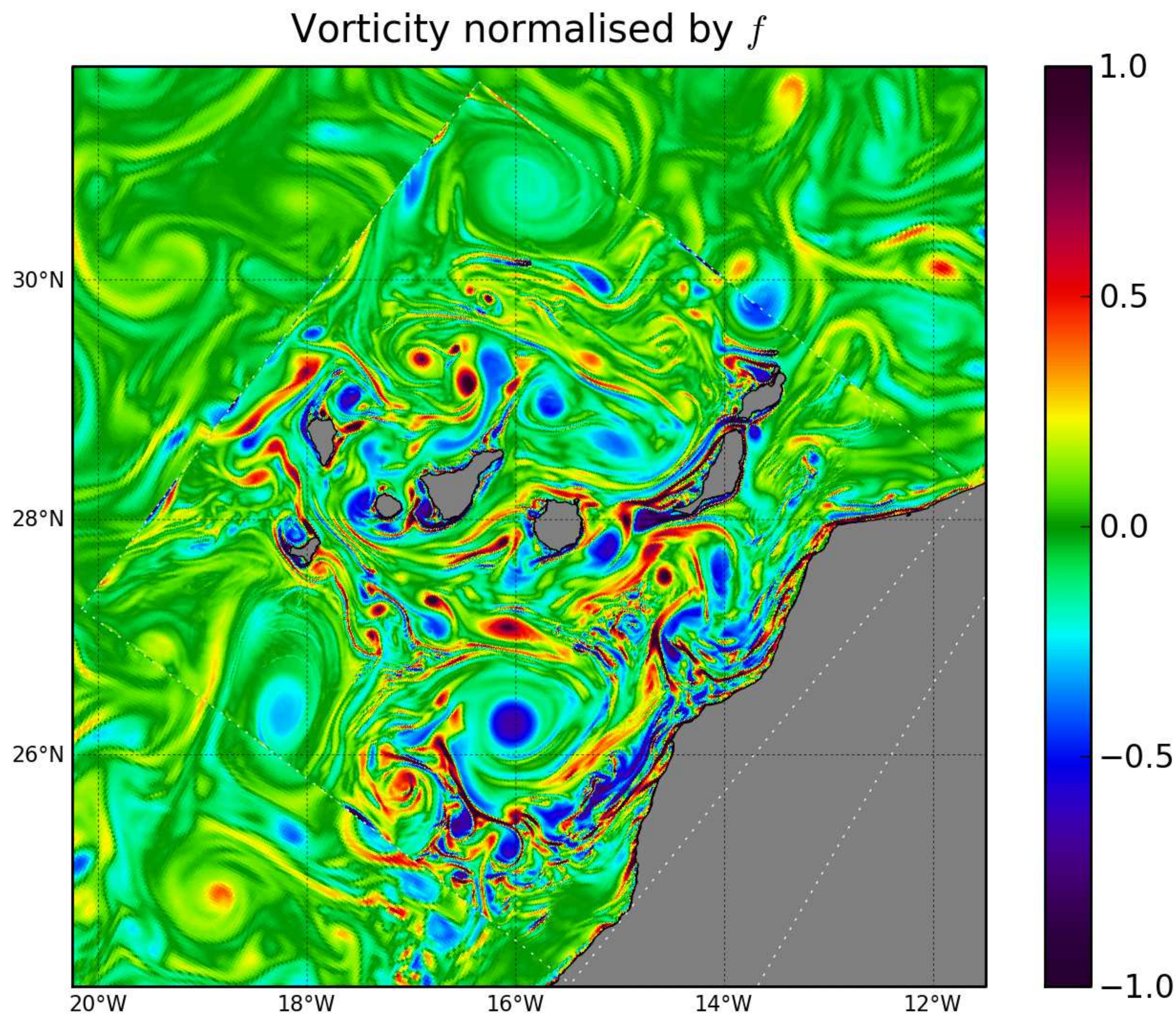
- **2-time-level, all-positive coefficients** *valuable for engineering codes*
- $\approx 1.7\times$ **gain in** $\alpha_{\max}$ **relatively to SM2005**
- efficiency comparable with forward-backward schemes

Grid nesting techniques

- downscaling
- 1-way
- new Flather-type characteristic BC for normal barotropic velocity
- radiation for normal 3D velocities
- upstream advection BC for tangential velocities
- upstream advection for tracers
- forced by 1 row of boundary points
- rotated grids
- **new "tools"**




Vorticity normalised by f
36°N
34°N
32°N
30°N
28°N
26°N
24°N
22°N
20°N
24°W
22°W
20°W
18°W
16°W
14°W
12°W
10°W
8°W
1.0
0.5
0.0
−0.5
−1.0



Vorticity normalised by f
30°N
28°N
26°N
20°W
18°W
16°W
14°W
12°W
1.0
0.5
0.0
−0.5
−1.0

Bottom drag: a modeler prospective

model needs $\quad \Delta z_1 \cdot \dfrac{u_1^{n+1} - u_1^{n}}{\Delta t} = A_{3/2} \cdot \dfrac{u_2^{n+1} - u_1^{n+1}}{\Delta z_{3/2}} - r_D \cdot u_1^{n?} \qquad r_D = ?$

where $u_1 \equiv u_{k=1}$ is understood in finite-volume sense $u_1 = \dfrac{1}{\Delta z_1} \displaystyle\int_{\text{bottom}}^{\text{bottom}+\Delta z_1} u\left(z'\right) \mathrm{d}z'$

from physics $\quad \text{STRESS} = F(u), \qquad F = ?$

duality of u_*: it controls **both** bottom stress and vertical viscosity profile

$$\text{STRESS} = u_*^2, \qquad \text{and} \qquad A = A(z) = \kappa u_* \cdot (z_0 + z) \qquad z \to 0$$

roughness length z_0 = statistically averaged scale of unresolved of topography

constant-stress boundary layer $\quad A(z) \cdot \partial_z u = \text{STRESS} = const = u_*^2$

$$\kappa u_* \left(z_0 + z\right) \partial_z u = u_*^2 \qquad \text{hence} \qquad u(z) = \frac{u_*}{\kappa} \ln\left(1 + \frac{z}{z_0}\right)$$

$$u_1 = \frac{u_*}{\kappa}\left[\left(\frac{z_0}{\Delta z_1} + 1\right) \ln\left(1 + \frac{\Delta z_1}{z_0}\right) - 1\right] \qquad \text{hence} \qquad u_* = \kappa \cdot u_1 / [...]$$

$$-r_D \cdot u_1 = -\kappa^2 \left|u_1\right| \cdot \left[\left(\frac{z_0}{\Delta z_1} + 1\right) \ln\left(1 + \frac{\Delta z_1}{z_0}\right) - 1\right]^{-2} \cdot u_1$$

$$r_D = \kappa^2 |u_1| \Bigg/ \left[\left(\frac{z_0}{\Delta z_1} + 1 \right) \ln \left(1 + \frac{\Delta z_1}{z_0} \right) - 1 \right]^2$$

well-resolved asymptotic limit for $\Delta z_1 / z_0 \ll 1$ is $\quad r_D \sim 4\kappa^2 |u_1| \cdot \dfrac{z_0^2}{\Delta z_1^2}$

however in this case $\quad u(z) = \dfrac{u_*}{\kappa} \ln \left(1 + \dfrac{z}{z_0} \right) \sim \dfrac{u_*}{\kappa} \cdot \dfrac{z}{z_0} \qquad$ hence $\quad u_1 = \dfrac{u_*}{\kappa} \cdot \dfrac{\Delta z_1}{2 z_0}$

resulting $\quad r_D \sim \kappa^2 u_* \cdot \dfrac{2 z_0}{\Delta z_1} = \dfrac{A_{\text{bottom}}}{\Delta z_1 / 2} \qquad$ in line with no-slip with laminar viscosity

unresolved $\Delta z_1 / z_0 \gg 1$ limit $\quad r_D \sim \kappa^2 |u_1| \Big/ \ln^2 \left(\dfrac{\Delta z_1}{z_0} \right) \qquad$ known as "log-layer"

- overall there is nothing unexpected
- smooth transition between resolved and unresolved
- avoids introduction of *ad hoc* "reference height" z_a, e.g., Soulsby (1995) formula $\quad \mathrm{STRESS} = [\kappa / \ln(\mathrm{z_a}/\mathrm{z_0})]^2 \cdot \mathrm{u}^2|_{\mathrm{z}=\mathrm{z_a}} \quad$ where $u|_{z=z_a}$ is hard (or impossible) to estimate from discrete variables
- in practice this differs by a factor of 2 from published formulas, e.g., Blaas (2007), with $z_a = \Delta z_1 / 2$, due to finite-volume vs. finite-difference interpretation of discrete model variables
- near-bottom vertical grid-box height Δz_1 is an inherent control parameter of r_D, making it impossible to specify "physical" quadratic drag coefficient, $r_D = C_{\mathrm{D}} \cdot |u|$

How large is $\frac{\Delta t \cdot r_D}{\Delta z_1}$?

$$\frac{\Delta t \cdot r_D}{\Delta z_1} = \underbrace{\frac{\Delta t \cdot |u_1|}{\Delta x}}_{\text{advective Courant number}} \cdot \kappa^2 \cdot \underbrace{\frac{\Delta x}{\Delta z_1} \bigg/ \left[\left(\frac{z_0}{\Delta z_1}+1\right)\ln\left(1+\frac{\Delta z_1}{z_0}\right)-1\right]^2}_{\text{purely geometric criterion}}$$

in unresolved case $\frac{\Delta x}{\Delta z_1} \cdot \left[\kappa \Big/ \ln\left(\frac{\Delta z_1}{z_0}\right)\right]^2$

Typical high-resolution ROMS practice $h_{\min} \sim 25m$, $N = 30...50$, hence $\Delta z \sim 1m$, $\Delta x = 1km$, and $z_0 = 0.01m$, $\kappa = 0.4$ estimates the above as 7.5.

- **$\sim 50...100$ in Bering Sea in our $\Delta x = 12.5$km Pacific simulation, even more in a coarser 1/5-degree**

It is mitigated by the bottom-most velocity Courant number ~ 0.1 but, still exceeds the limit of what explicit treatment can handle

- sigma-models are the most affected, but they are the ones which are mostly used when bottom drag matters
- vertical grid refinement toward the bottom makes this condition stiffer

Implicit treatment of $-\Delta t \cdot r_D \cdot u_1^{n+1}$ term: include it into implicit solver for vertical viscosity terms, however this interferes with Barotropic Mode (BM) splitting:

- Bottom drag can be computed only from full 3D velocity, but not from the vertically averaged velocities alone.
- Barotropic Mode must know the bottom drag term **in advance** as a part of 3D$\rightarrow$2D forcing for consistency of splitting. This places computing vertical viscosity before BM, however, later when BM corrects the vertical mean of 3D velocities, it *destroys* the consistency of (no-slip like) bottom boundary condition.
- If BM receives bottom drag based on the most recent state of 3D velocity **before** BM, but the implicit vertical viscosity terms along with (the final) bottom drag are computed **after** BM is complete (hence accurately respecting the bottom boundary condition), this changes the state of vertical integrals of 3D velocities, interfering with BM in keeping the vertically integrated velocities in nearly non-divergent state.
- Current ROMS practice is to split bottom drag term from the rest of vertical viscosity computation. This limits the time step (or r_D itself) by the explicit stability constraint.

Ekman layer in shallow water: $h = 10m$, $u_* = 6 \times 10^{-2} m/s$ ($\approx 5m/s$ wind), $f = 10^{-4}$, $A_v = 2 \times 10^{-3} m^2/s$, non-slip at $z = -h$

Top: Explicit, CFL-limited, bottom drag **before** Barotropic Mode (BM) for **both** r.h.s. 3D and for BM forcing ($\Rightarrow$ no splitting error); implicit step for vertical viscosity **after** with bottom drag excluded ($\Rightarrow$ undisturbed coupling of 2D and 3D); need $r_D < \Delta z_{\text{bottom}}/\Delta t_{\text{3D}}$ for stability

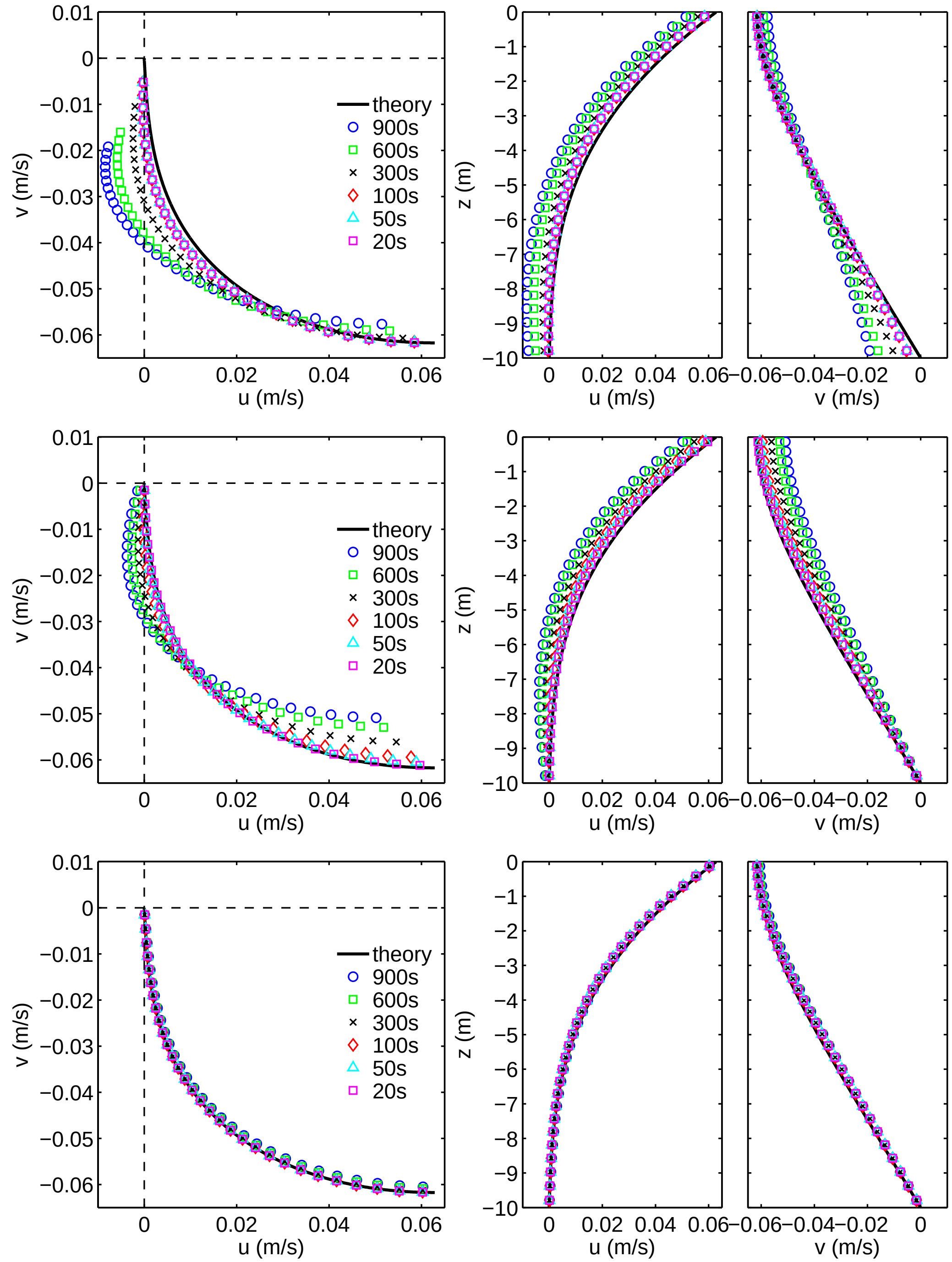
Middle: Unlimited drag **before** BM applies for BM forcing **only**; implicit vertical viscosity **after** with drag included into implicit solver (i.e., the drag is recomputed relative to what BM got before $\Rightarrow$ splitting error)

Bottom: Bottom drag is computed as a part of implicit vertical viscosity step **before** and for **both** 3D and BM forcing

In all cases BM has bottom drag term which captures its tendency in fast time

$$\partial_t \overline{\mathbf{U}} = \ldots \underbrace{\left[\underbrace{-r_D \cdot \mathbf{u}_{\text{bottom}}}_{\text{drag from 3D mode}} + r_D \cdot \overline{\mathbf{u}}^{m=0} \right]}_{\text{3D}\rightarrow\text{BM forcing}} - r_D \cdot \overline{\mathbf{u}}$$

so when $\mathbf{u}_{\text{bottom}}$ is updated/corrected by BM, so does the $-r_D \cdot \mathbf{u}_{\text{bottom}}$ term computed from it; above $\overline{\mathbf{U}} = (h + \zeta)\overline{\mathbf{u}}$



- Classical operator splitting dilemma

$$\partial_t \mathbf{u} = \mathcal{R}(\mathbf{u}) \qquad \text{where} \qquad \mathcal{R}(\mathbf{u}) = \mathcal{R}_1(\mathbf{u}) + \mathcal{R}_2(\mathbf{u}) \qquad \text{both are stiff, but}$$

$$\mathbf{u}^{n+1} = \mathbf{u}^n + \Delta t \cdot \mathcal{R}\left(\mathbf{u}^{n:n+1}\right)$$

is not practical because of complexity (implicitness), so instead

$$\mathbf{u}' = \mathbf{u}^n + \Delta t \cdot \mathcal{R}_1(\mathbf{u}^{n:\prime}) \qquad \text{followed by} \qquad \mathbf{u}^{n+1} = \mathbf{u}' + \Delta t \cdot \mathcal{R}_2(\mathbf{u}^{\prime:n+1})$$

$$\mathbf{u}^{n+1} = [1 + \Delta t \cdot \mathcal{R}_2(.)] \cdot [1 + \Delta t \cdot \mathcal{R}_1(.)]\, \mathbf{u}^n$$

$$[1 + \Delta t \cdot \mathcal{R}_2(.)] \cdot [1 + \Delta t \cdot \mathcal{R}_1(.)] \neq [1 + \Delta t \cdot \mathcal{R}_1(.)] \cdot [1 + \Delta t \cdot \mathcal{R}_2(.)]$$

resulting in $\mathcal{O}(\Delta t)$ operator splitting error. Especially inaccurate in near cancellation $R_1 \approx -R_2$ situation (balance).

- reminiscent of implicit no-slip boundaries + pressure-Poisson projection method for incompressible flows
- Requires substantial redesign of ROMS kernel
- somewhat encourages **anti-modular** code design
- Possible only in corrector-coupled and Generalized FB variants of ROMS kernels
- Incompatible (or at least hard to implement) in Rutgers kernel because of forward extrapolation of r.h.s. terms for 3D momenta (AB3 stepping) and extrapolation of 3D→BM forcing terms which is not compatible with having stiff terms there
- Incompatible with predictor-coupled kernel (currently used by AGRIF), because of extrapolation of 3D→BM forcing, and because overall having BM too early the computing sequence (implicit vertical viscosity step is done only after predictor step for tracers which is after BM)
- **Must have, long overdue**

Other models? POM? GETM?