Numerically Accurate treatment of bottom drag in ocean models with mode and time splitting

Alexander F. Shchepetkin

Salt Lake Sity, UT, February 2012

Sec. 006 Advances in Coastal Ocean Modeling, Analysis, and Prediction. rm 251; 8:45am

- While representation of bottom drag term is essential for coastal modeling, the basic mathematical dilemma of handling an implicit no-slip bottom boundary condition in a mode-split model (whether split-explicit or implicit free-surface) is not satisfactorily solved within the oceanic modeling community, and, moreover, most current oceanic codes do not even allow this without a major algorithmic redesign.

- The essence of the problem is the splitting of two stiff operators – one is associated with the Barotropic Mode splitting, the other is due to the implicit no-slip boundary condition at the bottom – a situation reminiscent to the classical dilemma in incompressible flows, e.g., Dukowicz & Dvinsky, 1992.

- Vertical grid refinement toward the bottom – a standard modeling practice motivated by the need to resolve turbulent bottom boundary layer – exacerbates the splitting dilemma resulting in poor convergence.

- This presentation overviews the current modeling practices and proposes self-consistent algorithms to address both the errors due to time splitting and handling of the discrete of no-slip bottom boundary condition in the turbulent case where regularization length (viscous sublayer) is only marginally resolved or not resolved at all.

Bottom drag: Physics and Discretization Issues

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

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

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

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 $A(z) \cdot \partial_z u = \text{STRESS} = const = u_*^2$

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

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

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

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

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

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

unresolved $\Delta z_1/z_0 \gg 1$ limit $\quad r_D \sim \kappa^2\,|u_1| \Big/ \ln^2\left(\dfrac{\Delta z_1}{z_0}\right) \quad$ 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 $\mathrm{STRESS} = [\kappa/\ln(\mathrm{z_a}/\mathrm{z_0})]^2\cdot \mathrm{u}^2\big|_{\mathrm{z=z_a}}$ where $u\big|_{z=z_a}$ 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 $\dfrac{\Delta t \cdot r_D}{\Delta z_1}$?

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

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

Typical high-resolution ROMS practice $h_{\text{min}} \sim 25m$, $N = 30...50$, 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→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:

$h = 10m$, $f = 10^{-4}$, $A_v = 2 \times 10^{-3} m^2/s$
$u_* = 6 \times 10^{-2} m/s$ ($\approx 5m/s$ wind),
non-slip at $z = -h$, $N = 30$

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 $r_D < \Delta z_{\text{bottom}} / \Delta t_{3D}$ for stability

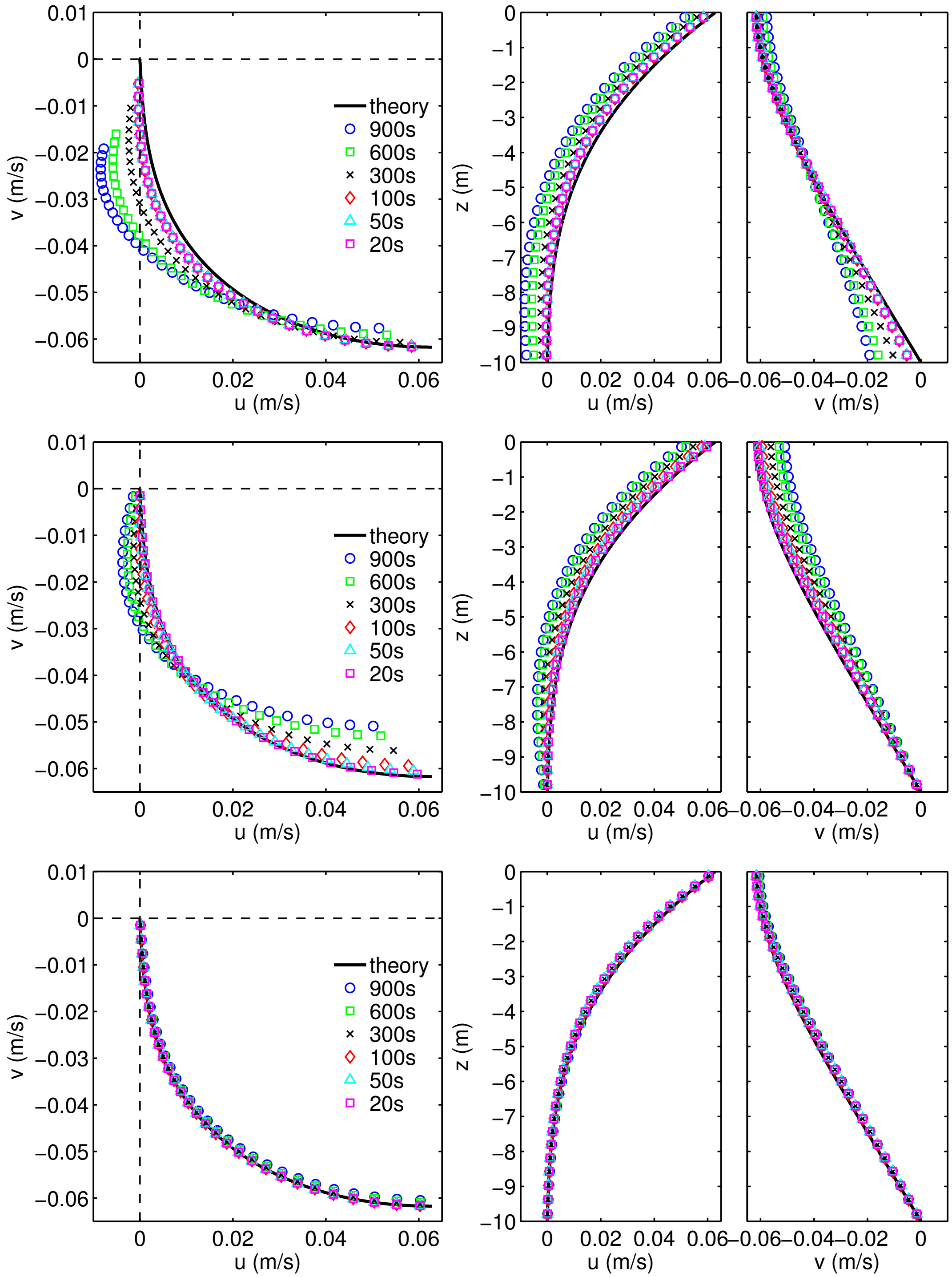
Middle: Unlimited drag **before** BM applies for BM forcing **only**; implicit vertical viscosity **after** with drag included into implicit solver **(i.e., the drag term is recomputed relative to what BM got before $\Rightarrow$ splitting error)**

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

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

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 $\overline{\mathbf{U}} = (h + \zeta)\overline{\mathbf{u}}$



Key components for small splitting error:

- No-slip B.C. and bottom drag term **must be included** into implicit vertical viscous solver
- *Both* the total bottom stress term and bottom drag coefficient must be available to BM (i.e. must be precomputed before BM)
- As BM advances vertical integrals of $\overline{u}, \overline{v}$, it should also take into account the incremental changes to bottom drag term, so after u, v^{n+1} are are adjusted by BM, the resultant adjusted bottom drag term is still in balance with the remaining terms as it was before BM run

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

Note: $-r_D \cdot \overline{\mathbf{u}}$ **cannot** simulate bottom drag by itself. *The sole purpose* of its presence is to make BM "feel" the incremental change in $-r_D \cdot \delta \overline{\mathbf{u}} = -r_D \cdot \left(\overline{\mathbf{u}} - \overline{\mathbf{u}}^{m=0} \right)$, so the subsequent correction of 3D u, v's by BM (which unavoidably changes the bottom drag if recomputed from the *updated* u, v's) nevertheless is able to predict the change in $-r_D \cdot \mathbf{u}_{\text{bottom}}$.

Flowchart of POM code: `pom2k.f`

Mellor, 2004 POM User's Manual

bottom friction coefficient

$$\mathtt{cbc}_{i,j} = \kappa^2 \Big/ \left[\ln \left(\frac{\Delta z_{i,j,k_b}}{z_{0b}} \right) \right]^2$$

restricted to $\quad 0.0025 < \mathtt{cbcmin} < \mathtt{cbc}_{i,j} < \mathtt{cbcmax} = 1.0$

`profu,v` compute quadratic drag coefficients,

$$\mathtt{tps}_{i+1/2,j} = \overline{\mathtt{cbc}}^x_{i+1/2,j} \sqrt{(u^n)^2 + \left(\left(\overline{\overline{v}}^{x,y} \right)^n \right)^2} \Bigg|_{i+1/2,j,k_b}$$

at u-location $i+1/2, j$ (similarly at $i, j+1/2$) and solve implicit vertical viscosity problem **together with no-slip bottom B.C.**

momentum fluxes at the bottom $\langle wu \rangle$, $\langle wv \rangle$

$$\mathtt{wubot}_{i+1/2,j} = -\mathtt{tps}_{i+1/2,j} \cdot u^{n+1}_{i+1/2,j,k_b}$$

are computed at the very end of **`profu,v`**

$\mathtt{wubot}_{i+1/2,j}$ and $\mathtt{wvbot}_{i,j+1/2}$ are applied to the r.h.s. of BM **during the next** time step, where they are kept **constant** in fast time (no adjustment to bottom drag term within BM)

- built-in delay between bottom drag and BM
- **subject to splitting error** (BM disturbs bottom B.C. after enforcing vertical integrals u, v at next step)
- can be *partially* repaired by introducing adjustment of bottom drag term into BM

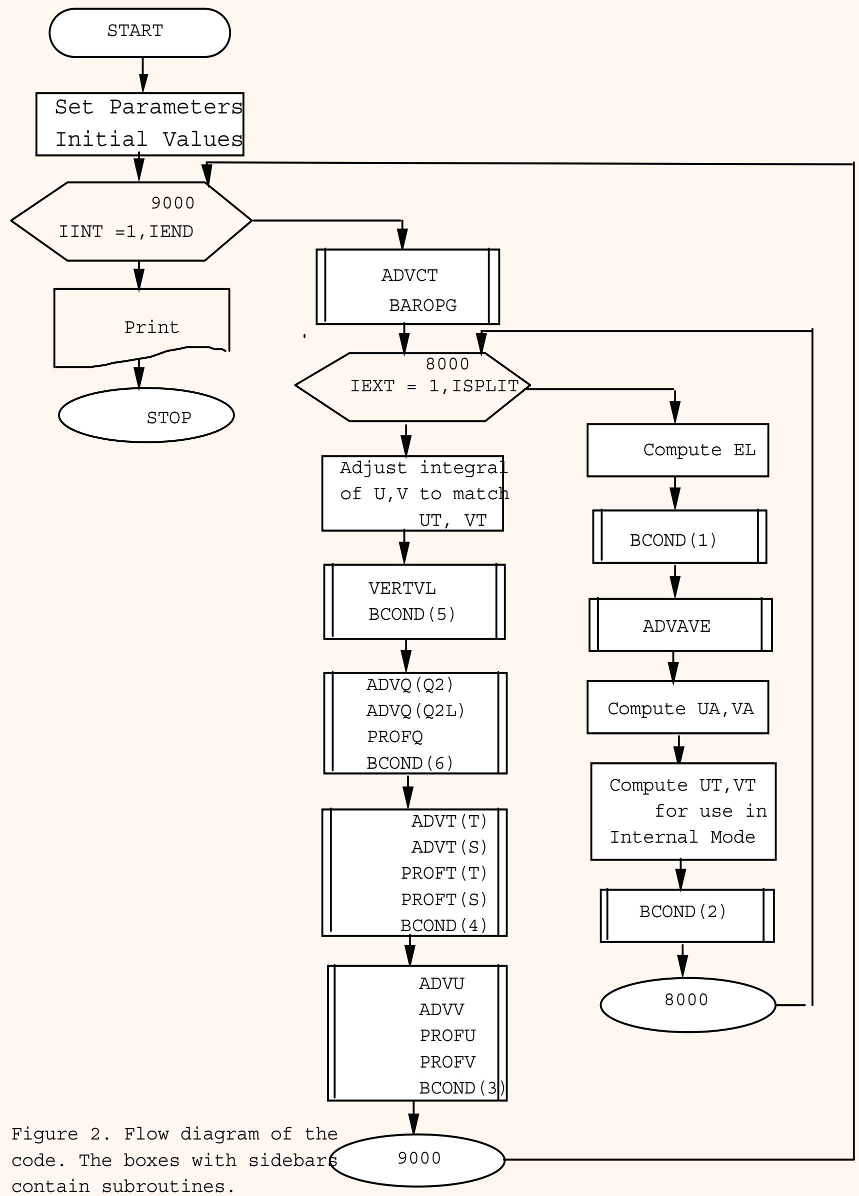


Figure 2. Flow diagram of the code. The boxes with sidebars contain subroutines.

Test problem: Upwelling response

Based on traditional ROMS "Upwelling test" configuration:

- EW periodic channel, 80 km wide;
- $h_{\text{max}} = 25m$ $h_{\text{max}} = 150m$;
- f-plane, $f = -8.26 \times 10^{-5}$ *southern* hemisphere;
- initially flat stratification in T;
- spatially uniform wind, 0.1 N/m^2 stress,
  modulated by $\sin\left(\frac{\pi}{4} \cdot t_{\text{[days]}}\right)$ for $0 < t < 2$ days;
  thereafter constant
- "salinity" is a passive tracer just to illustrate flow, and

either

- analytical vertical viscosity profile, constant in time

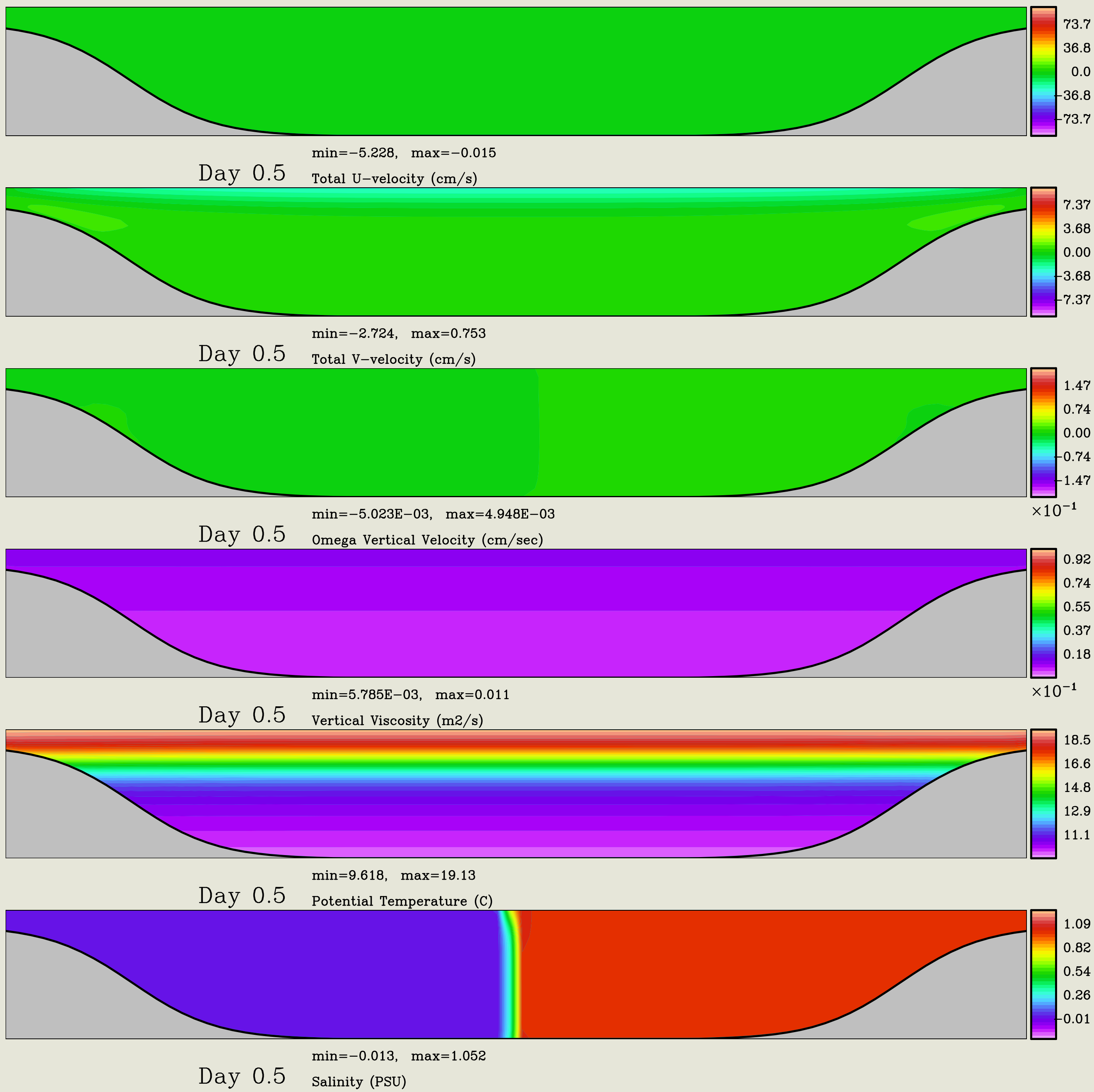
or

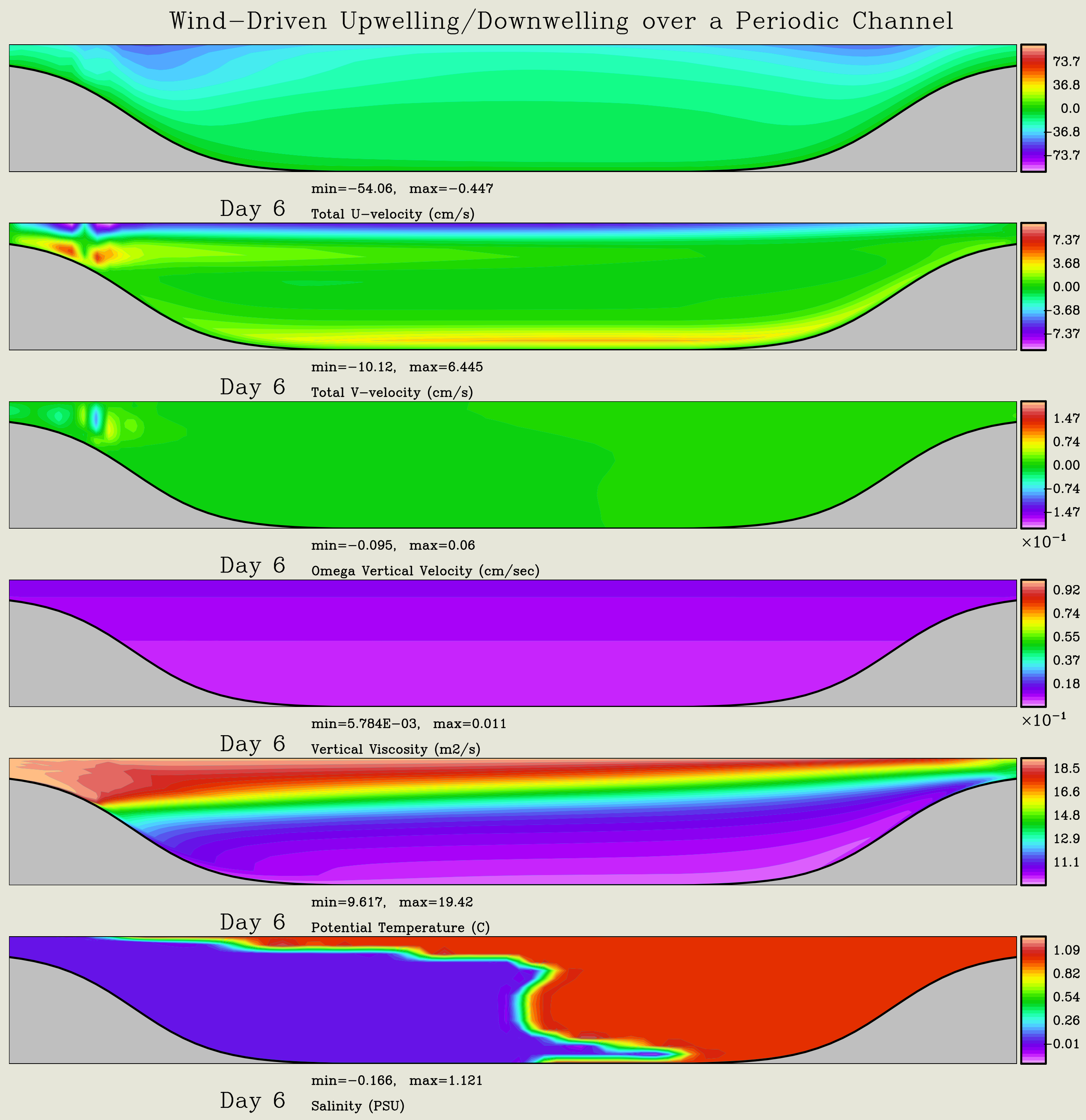
- KPP, both top and bottom, dynamically changing

Wind–Driven Upwelling/Downwelling over a Periodic Channel

Analytical

A_{kv} profile

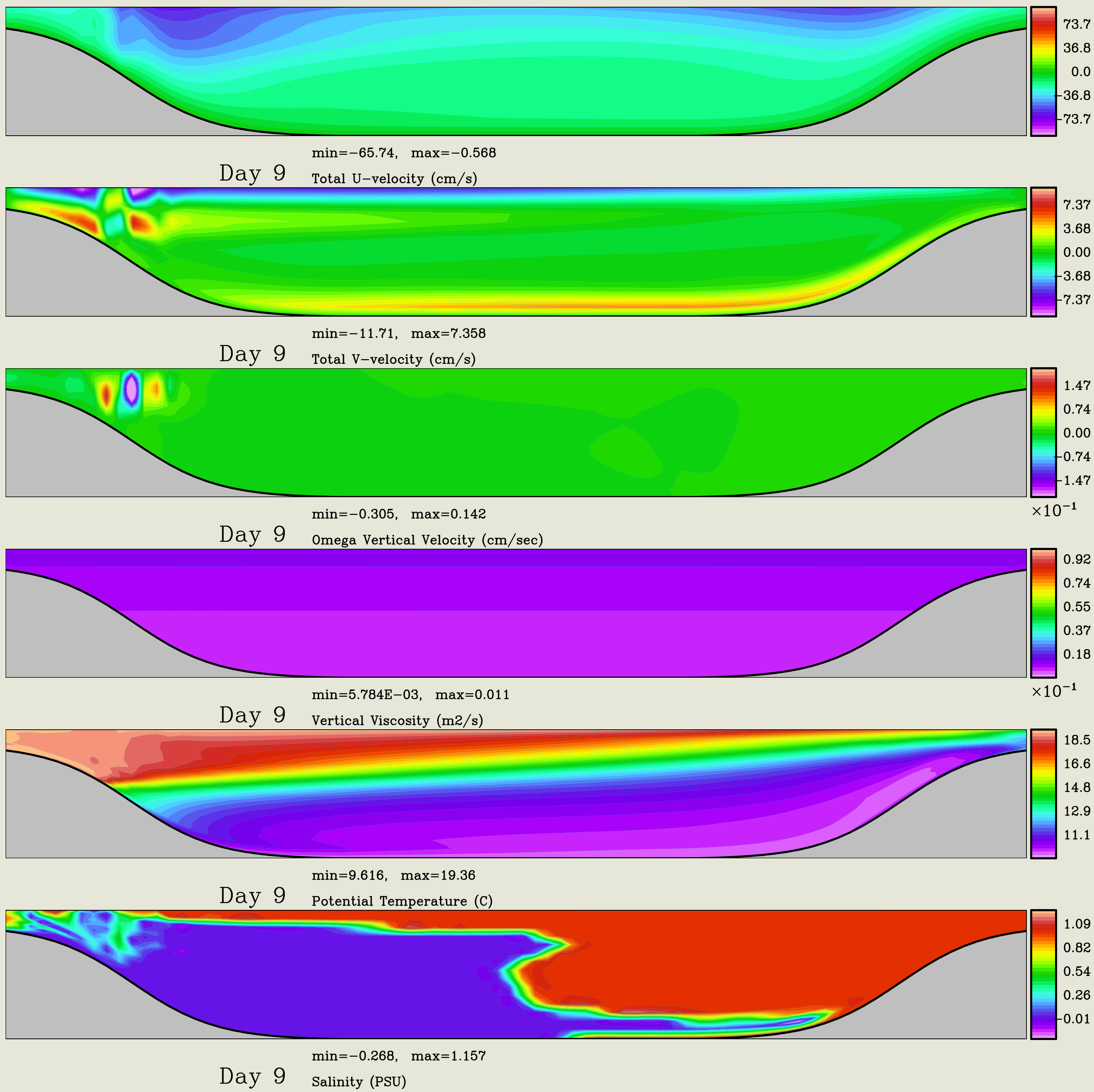




Analytical

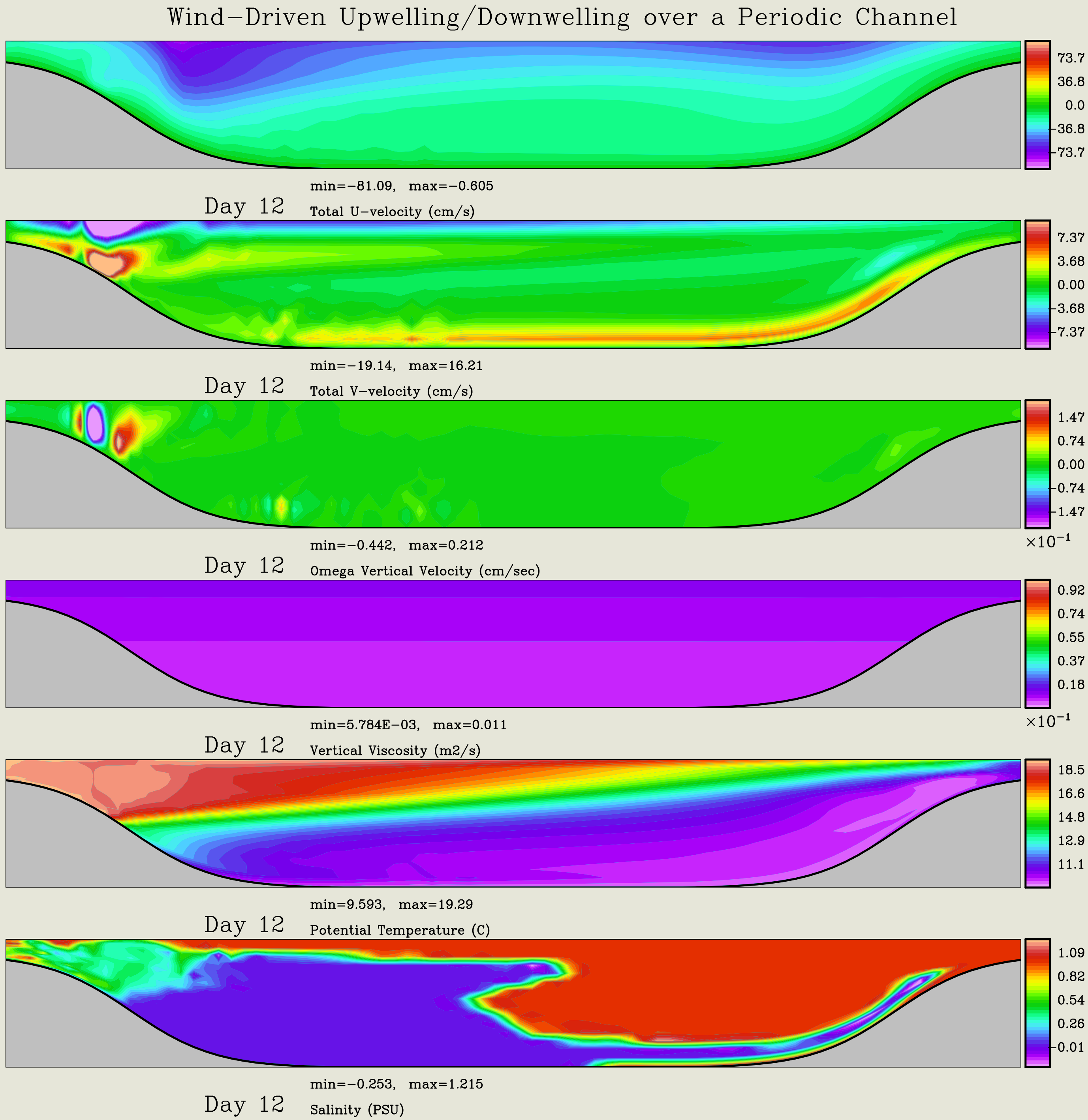
A_{kv} profile

Wind−Driven Upwelling/Downwelling over a Periodic Channel



Analytical

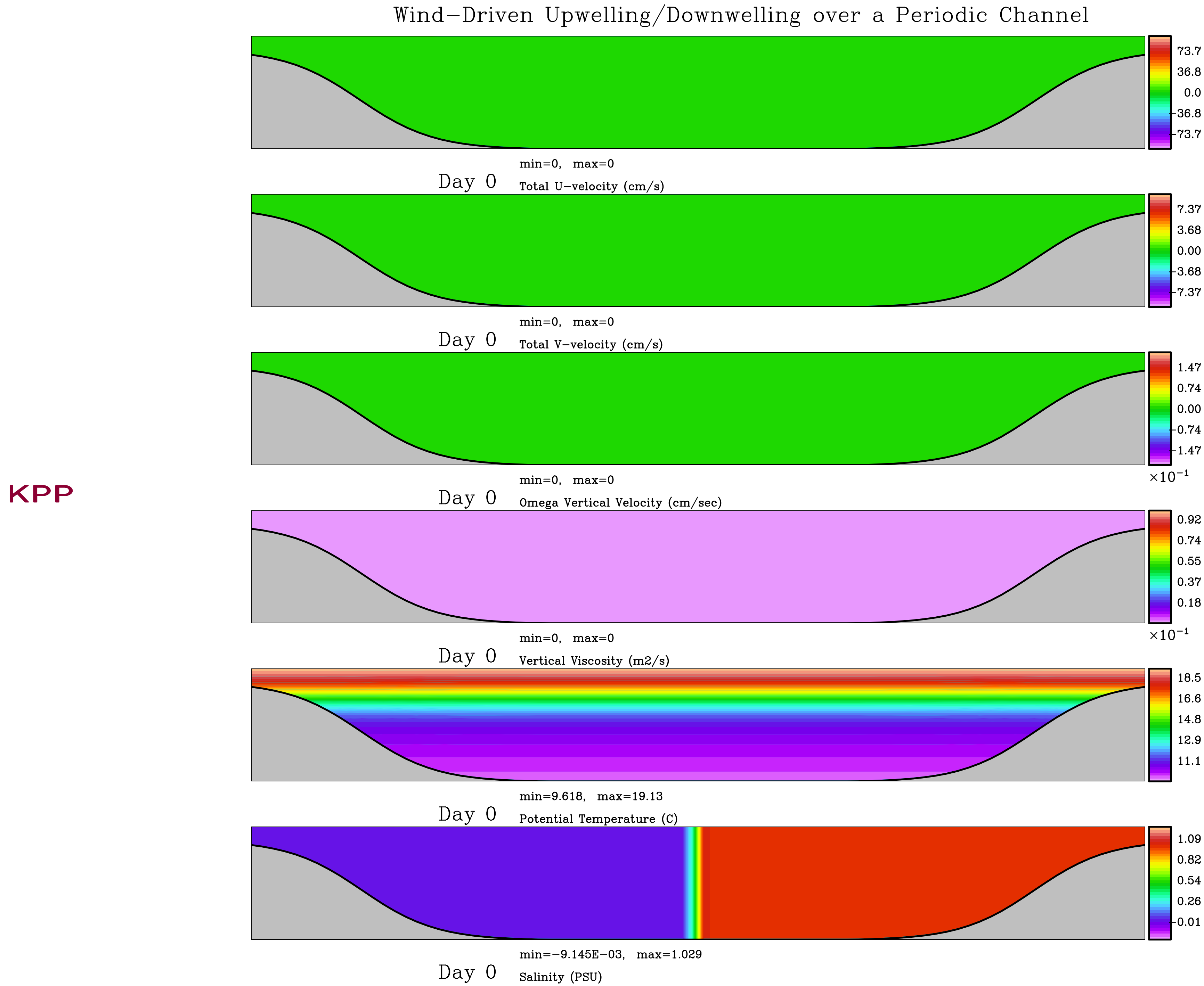
A_{kv} profile



Analytical

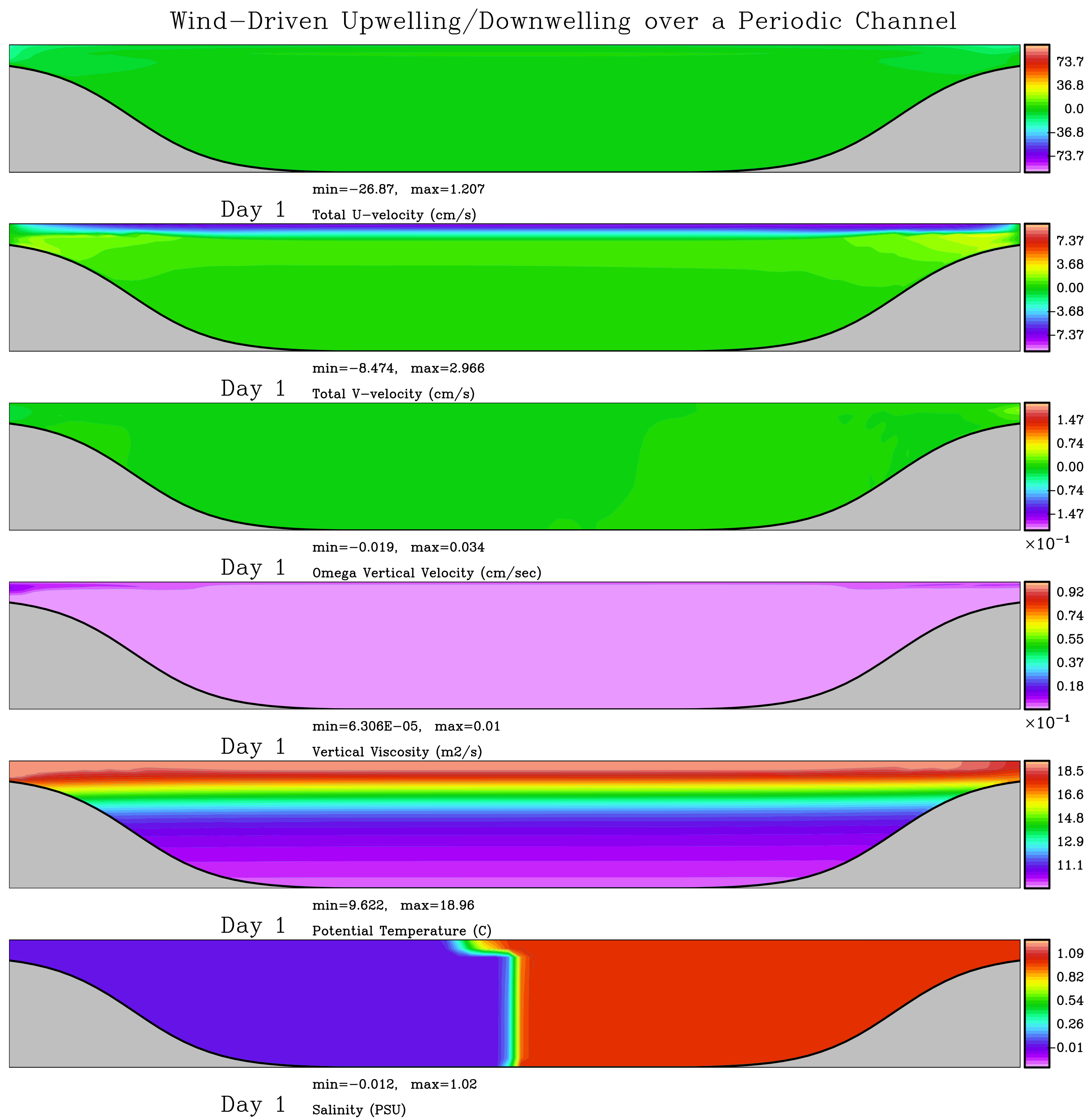
A_{kv} profile

- **same, but with KPP, top and bottom**

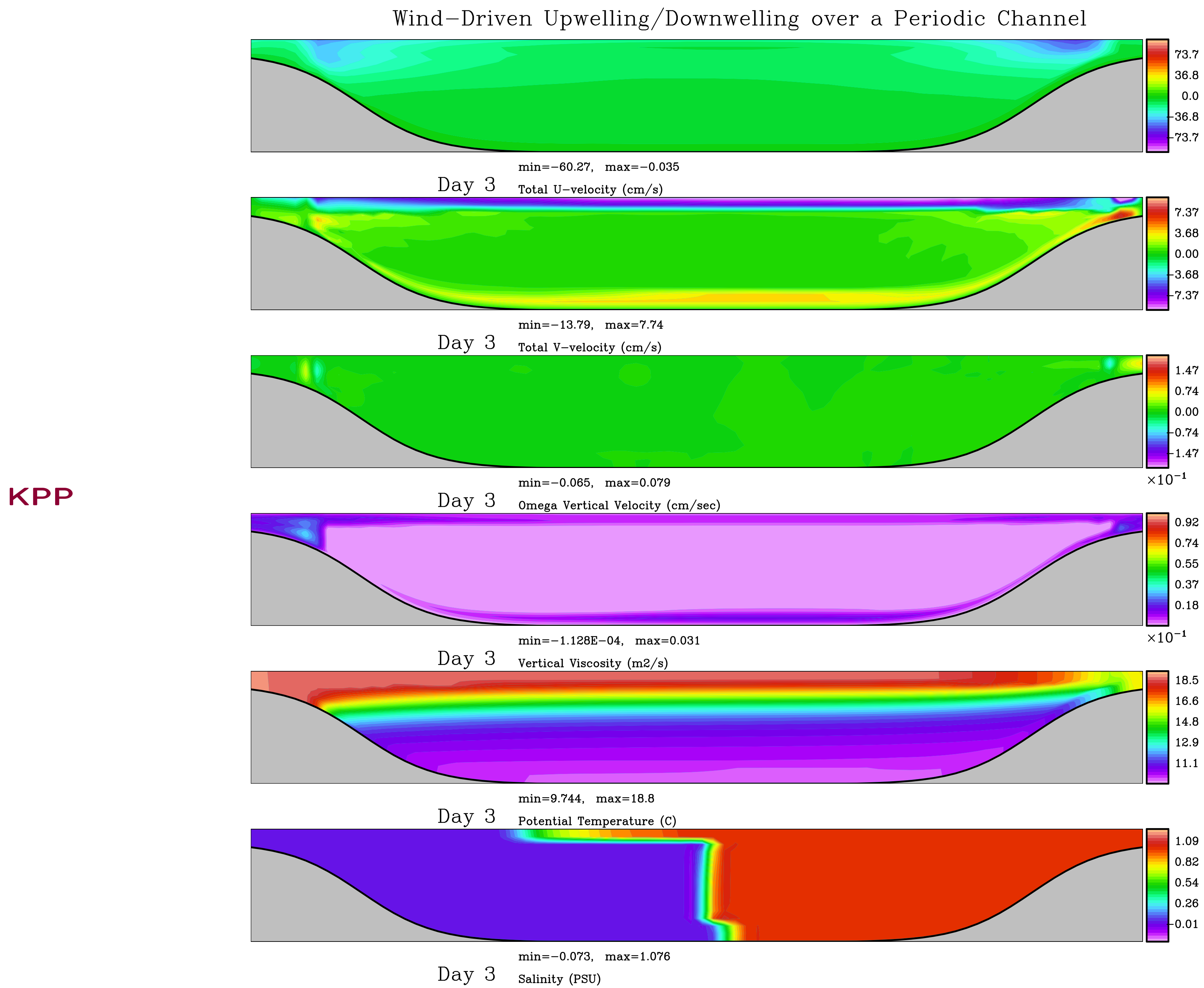


Wind−Driven Upwelling/Downwelling over a Periodic Channel
KPP
73.7
36.8
0.0
-36.8
-73.7
min=0, max=0
Day 0
Total U−velocity (cm/s)
7.37
3.68
0.00
-3.68
-7.37
min=0, max=0
Day 0
Total V−velocity (cm/s)
1.47
0.74
0.00
-0.74
-1.47
×10⁻¹
min=0, max=0
Day 0
Omega Vertical Velocity (cm/sec)
0.92
0.74
0.55
0.37
0.18
×10⁻¹
min=0, max=0
Day 0
Vertical Viscosity (m2/s)
18.5
16.6
14.8
12.9
11.1
min=9.618, max=19.13
Day 0
Potential Temperature (C)
1.09
0.82
0.54
0.26
0.01
min=−9.145E−03, max=1.029
Day 0
Salinity (PSU)

KPP

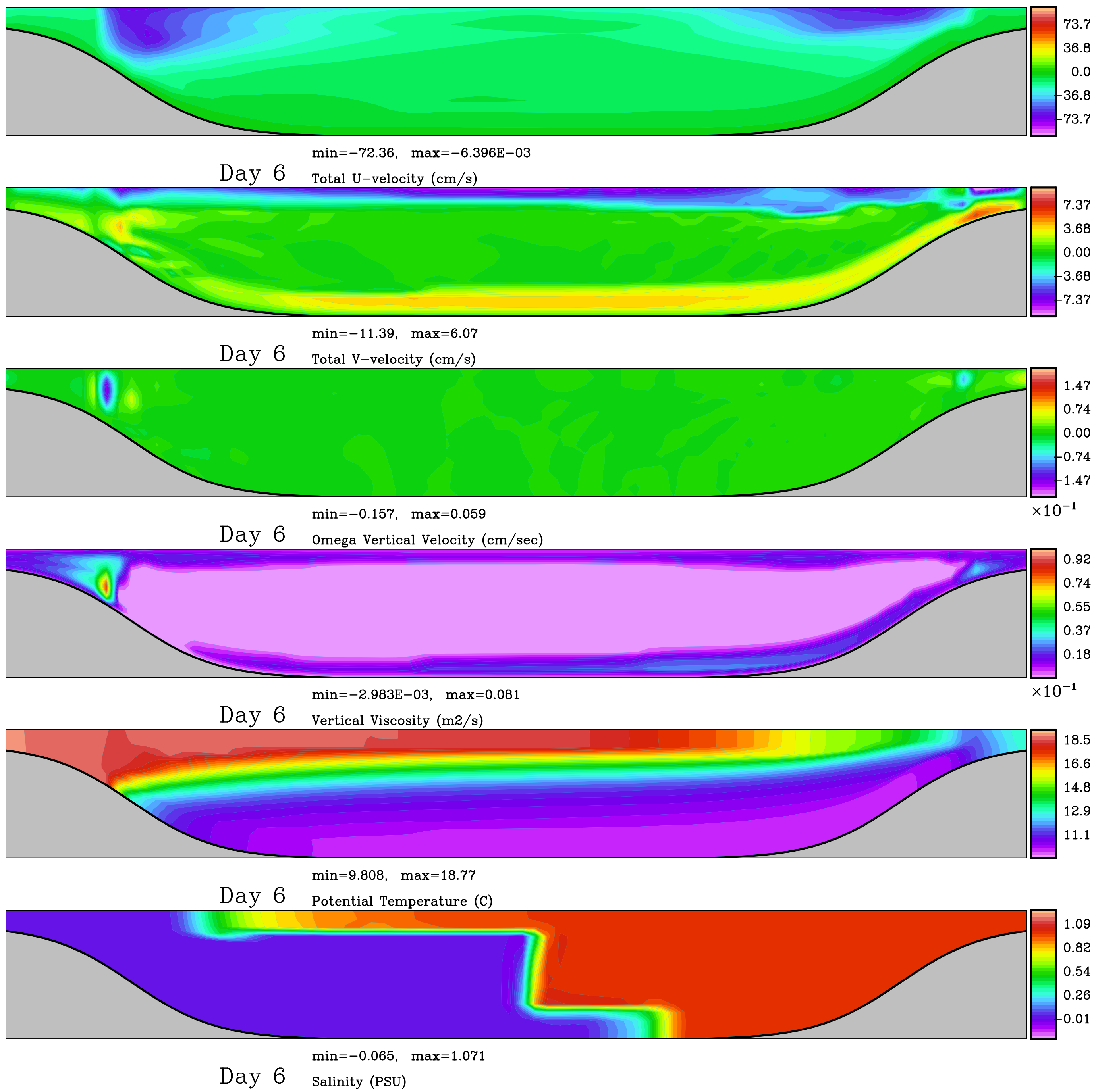


Wind-Driven Upwelling/Downwelling over a Periodic Channel
73.7
36.8
0.0
-36.8
-73.7
min=-26.87, max=1.207
Day 1
Total U-velocity (cm/s)
7.37
3.68
0.00
-3.68
-7.37
min=-8.474, max=2.966
Day 1
Total V-velocity (cm/s)
1.47
0.74
0.00
-0.74
-1.47
×10⁻¹
min=-0.019, max=0.034
Day 1
Omega Vertical Velocity (cm/sec)
0.92
0.74
0.55
0.37
0.18
×10⁻¹
min=6.306E-05, max=0.01
Day 1
Vertical Viscosity (m2/s)
18.5
16.6
14.8
12.9
11.1
min=9.622, max=18.96
Day 1
Potential Temperature (C)
1.09
0.82
0.54
0.26
0.01
min=-0.012, max=1.02
Day 1
Salinity (PSU)


Wind–Driven Upwelling/Downwelling over a Periodic Channel
73.7
36.8
0.0
-36.8
-73.7
min=−60.27, max=−0.035
Day 3
Total U–velocity (cm/s)
7.37
3.68
0.00
-3.68
-7.37
min=−13.79, max=7.74
Day 3
Total V–velocity (cm/s)
1.47
0.74
0.00
-0.74
-1.47
$\times10^{-1}$
min=−0.065, max=0.079
Day 3
Omega Vertical Velocity (cm/sec)
KPP
0.92
0.74
0.55
0.37
0.18
$\times10^{-1}$
min=−1.128E−04, max=0.031
Day 3
Vertical Viscosity (m2/s)
18.5
16.6
14.8
12.9
11.1
min=9.744, max=18.8
Day 3
Potential Temperature (C)
1.09
0.82
0.54
0.26
-0.01
min=−0.073, max=1.076
Day 3
Salinity (PSU)

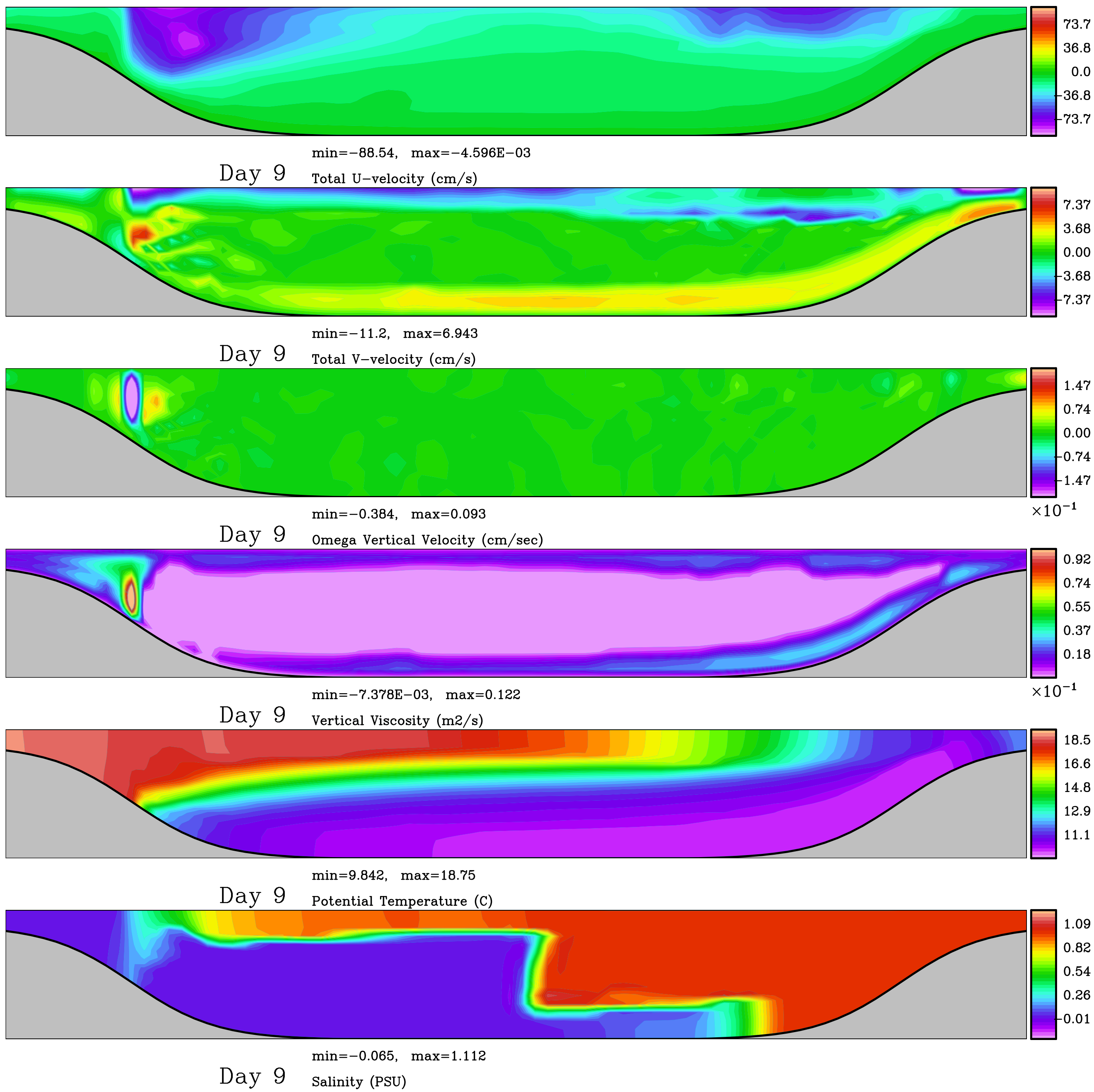
Wind−Driven Upwelling/Downwelling over a Periodic Channel

KPP



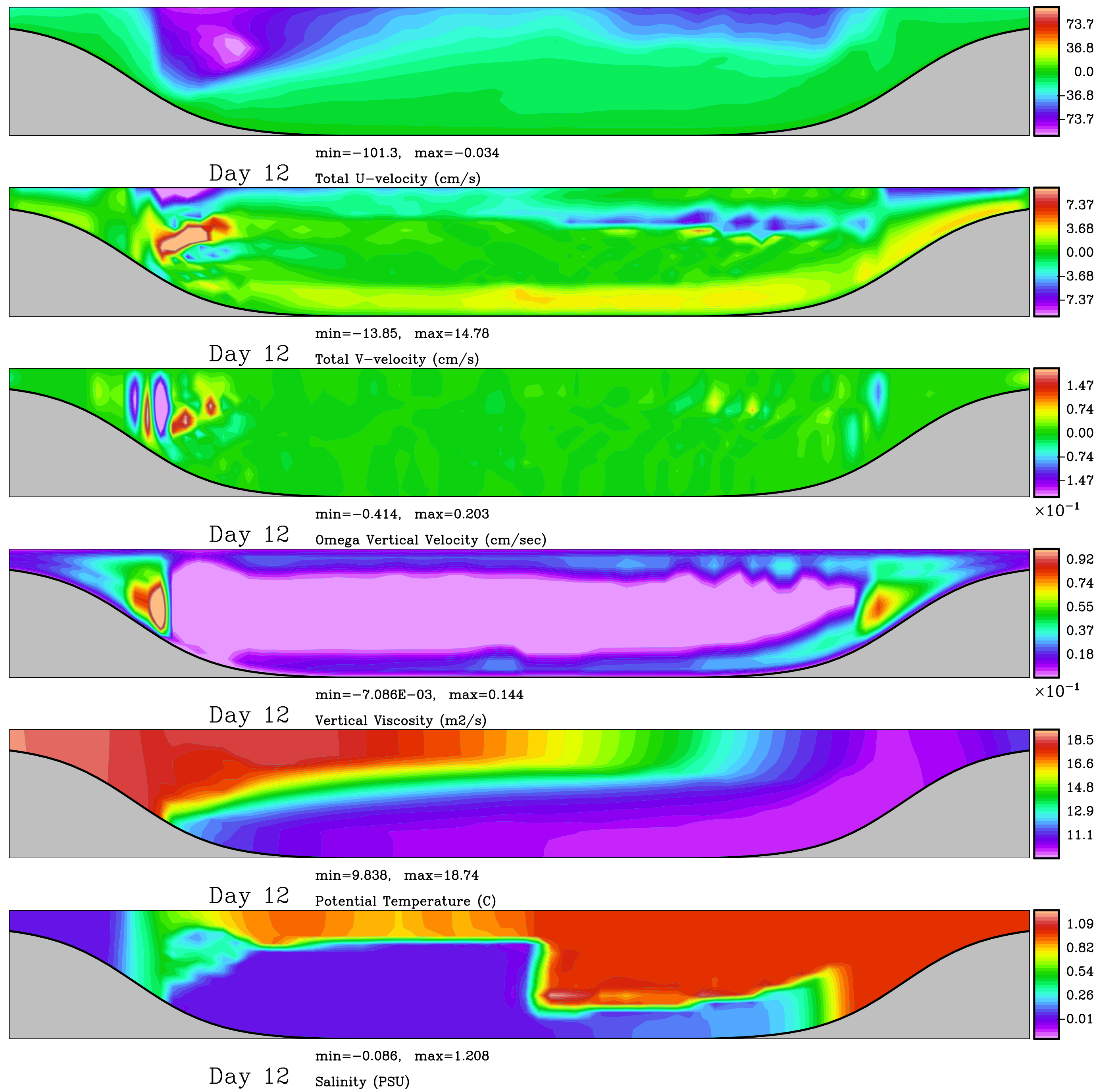
Wind−Driven Upwelling/Downwelling over a Periodic Channel

KPP

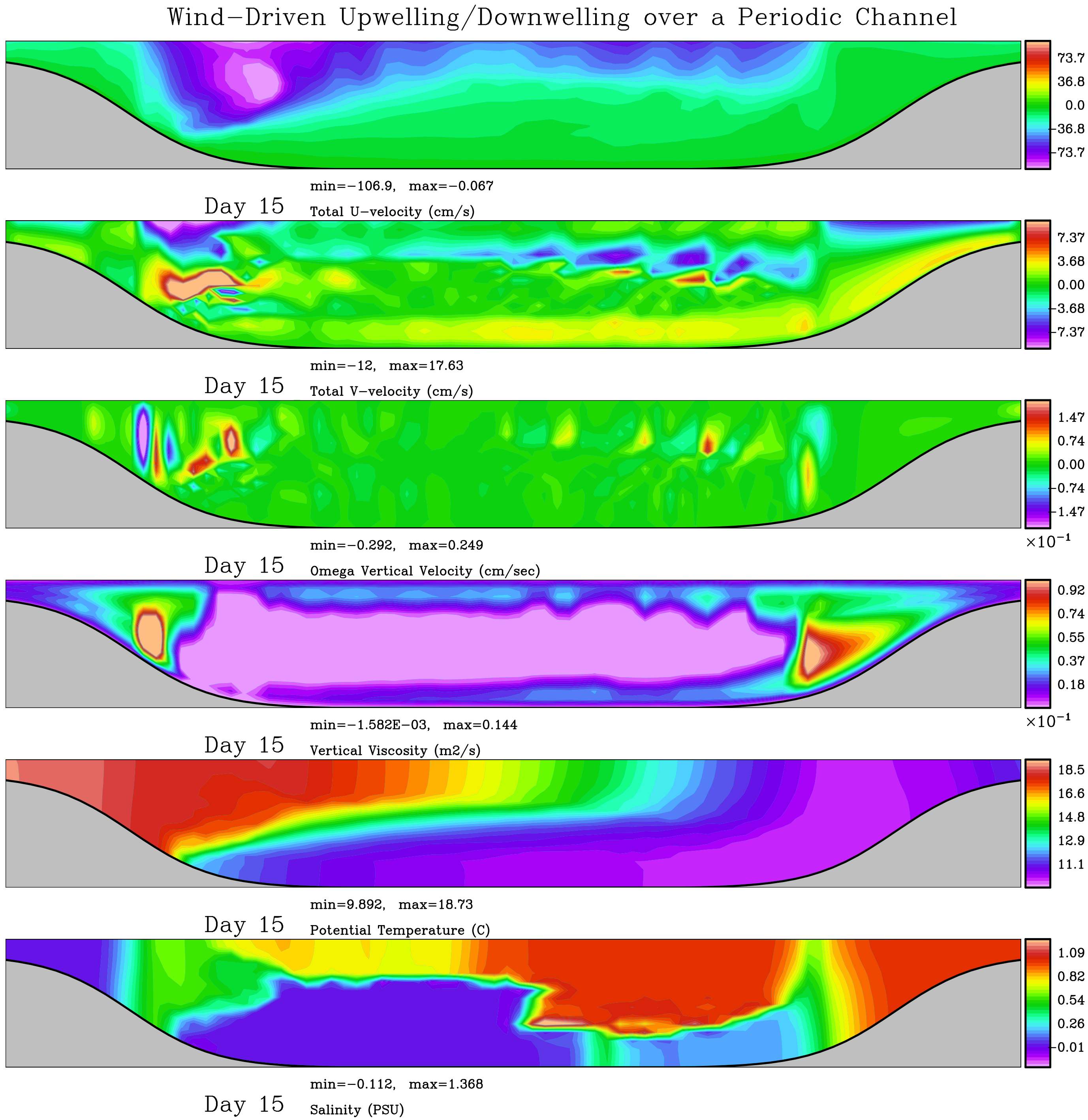


Wind−Driven Upwelling/Downwelling over a Periodic Channel

KPP



KPP



Wind−Driven Upwelling/Downwelling over a Periodic Channel
73.7
36.8
0.0
-36.8
-73.7
min=−106.9, max=−0.067
Day 15
Total U−velocity (cm/s)
7.37
3.68
0.00
-3.68
-7.37
min=−12, max=17.63
Day 15
Total V−velocity (cm/s)
1.47
0.74
0.00
-0.74
-1.47
×10⁻¹
min=−0.292, max=0.249
Day 15
Omega Vertical Velocity (cm/sec)
0.92
0.74
0.55
0.37
0.18
×10⁻¹
min=−1.582E−03, max=0.144
Day 15
Vertical Viscosity (m2/s)
18.5
16.6
14.8
12.9
11.1
min=9.892, max=18.73
Day 15
Potential Temperature (C)
1.09
0.82
0.54
0.26
0.01
min=−0.112, max=1.368
Day 15
Salinity (PSU)

Realistic example: USWC L4 Palos Verdes configuration

USWC L4 Palos Verdes grid configuration

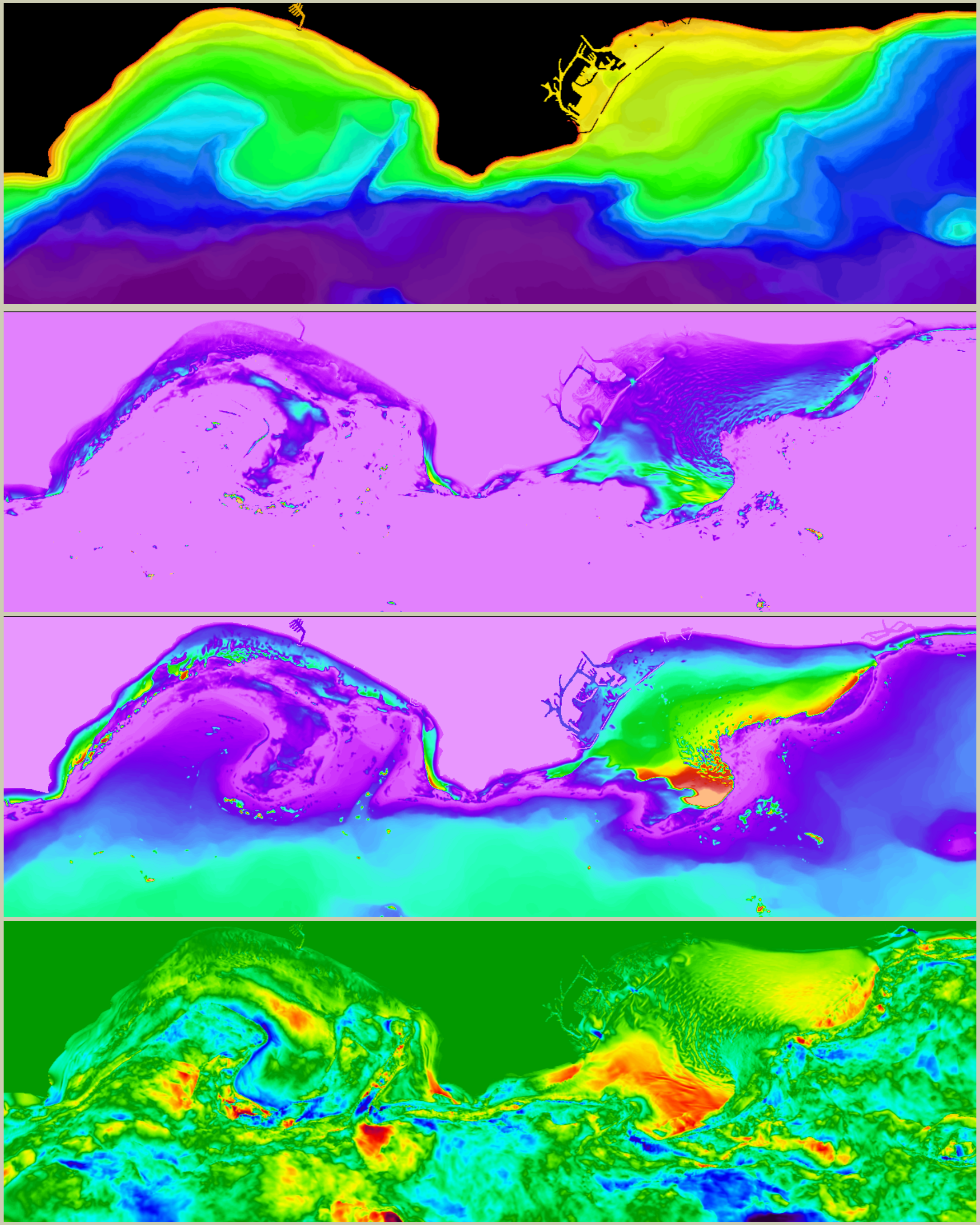
$\Delta x = 75m$

h, logscale, $\mathtt{min} = 1.5m$ $\mathtt{max} = 900m$

r_D, max=0.01

h_{bbl} by KPP max=$50m$

u, $\pm 0.2\, m/s$



Summary:

- Overall 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 ROMS 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 variant of ROMS kernel (currently used by AGRIF), because of extrapolation of 3D→BM forcing, and 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**