On feasibility of using ROMS as ocean dynamical core in next generation CESM

Alexander F. Shchepetkin, UCLA

21st Annual CESM Workshop

Breckenridge, Colorado, June 20-23, 2016

Algorithmic Features of ROMS: Mathematical Kernel

- **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 most-often used "z-sigma"

- orthogonal curvilinear grid in horizontal directions (keeping all metric terms)

- Boussinesq approximation, with EOS stiffening (not all)

- **time stepping engine:** split-explicit, free-surface, barotropic-baroclinic mode splitting; **exact** simultaneous finite-volume, finite-time-step, **conservation** and **constancy** for tracers via **2-way weighted fast-time-averaging** for barotropic velocities and free surface

- built around new time-stepping algorithms for hyperbolic system equations (waves); **always use forward-backward feedback**: momentum – tracers (3D) and free-surface – barotropic velocities (2D)

- **exact restart capability**

- selectively **higher-than-second-order** accuracy for critical terms: advection (*both* tracers and **momentum**, pressure-gradient, etc

- **adaptively-implicit** vertical advection for momentum and tracers – can exceed vertical CFL limit, but makes **no compromise** in accuracy when CFL is within the stability bounds of explicit scheme, nor makes any non-differentiable switching

- rotated (isopycnic or else) lateral diffusion (and/or hyper-) for tracers using switched triads stencil and **vertical implicitness** for mitigate stability constraints under stiff slope conditions

...Algorithmic Features: Physics

Primarily intended for modeling highly-turbulent flow regimes, sub-mesoscale fronts (sub $1km$-resolution), flow-topography interaction, etc... run in a limited-area

- ability to deal with open boundary conditions and grid nesting
- focus on higher-order advection
- **vortex stretching due to flow over non-uniform topography, topographic Rossby effect**
- bottom-boundary layer phenomena: ability to focus resolution near bottom
- more than one vertical mixing closure scheme
- **"If-less" KPP for top and bottom boundary layers**
- sub-models: biological, WEC, sediments, Sea-Ice, etc...

At the same time

- long-term **water-mass conservation properties** were historically de-emphasised – such errors naturally tend to ventilate through open boundaries in a typical US West Coast configuration. **This is being reconsidered.**
- ROMS never intended to become a climate model – **a sigma-model is unlikely to succeed at 1-degree resolution** – thought precursor sumulations were performed
- ROMS was never run in a global configuration

...Algorithmic Features: Computer science

- parallel via 2-level 2D-sub-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
- time stepping machinery for 3D part is build to balance computational effort between hydrodynamic core (advection, Coriolis, pressure gradient terms) and mixing parameterization schemes (horizontal and vertical) – the latter are kept outside the predictor-corrector cycle
- *intelligent* self-tracking and self-documentation
- massive use of CPP for management and configuration

ROMS Community

August 24, 1997 as starting date for the project: **almost 19 years old now**

October 1998: Acronym **"ROMS"** (a.k.a. Regional Oceanic Modeling System) was introduced by Dale Haidvogel during Davis Meeting

ROMS evolved (as opposite to *intelligent design*) with full range of consequences of this process:

- from Hernan's SCRUM 3.0 code
- mess, claims, controversy, scrutiny...
- *absence of claims*, ...sometimes
- inheritance, legacies, sometimes irrational choices, branches
- **there is more than one code called ROMS**

Linux-like-spirited community with multiple development cent*re*s

- exchange of ideas, mutual help network
- **voices to unify and make one "official" code → go elsewhere and unify something else**

ROMS have survived ... despite (or due?) all of the above

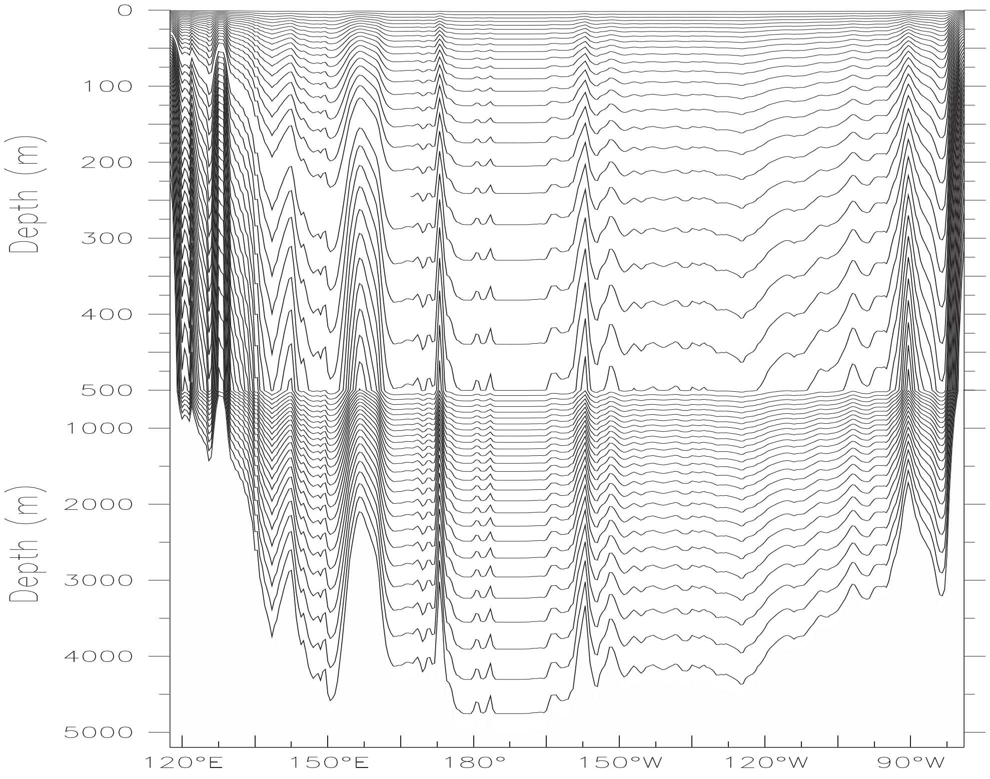
- **z-sigma** $z = z^{(0)} + \zeta\left(1 + \frac{z^0}{h}\right)$ $\qquad z^{(0)} = h \cdot \frac{h_c \cdot s + h \cdot C(s)}{h + h_c}$ $\qquad h = h(x,y)$

  $h_c = \text{const}$

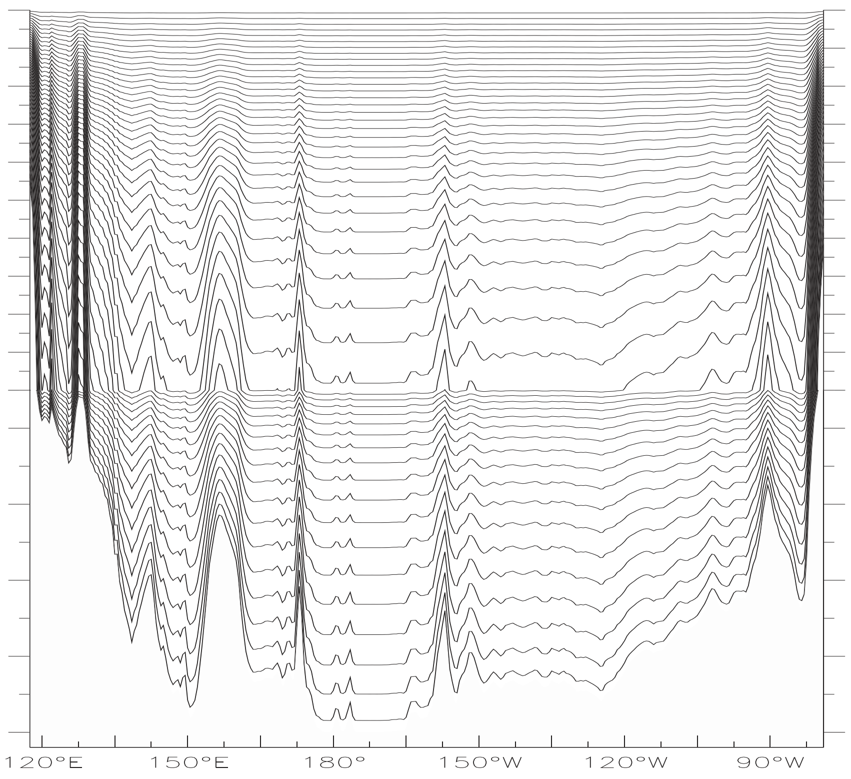
  $C(s) \equiv C[S(s)]$ where

$$S(s) = \frac{1 - \cosh(\theta_s s)}{\cosh(\theta_s) - 1} \begin{array}{l} \nearrow 0 \\ \searrow -1 \end{array} \qquad C(S) = \frac{\exp\{\theta_b S\} - 1}{1 - \exp\{-\theta_b\}} \begin{array}{llll} \nearrow & 0, & s = 0, & \text{surface} \\ \searrow & -1, & s = -1, & \text{bottom} \end{array}$$

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



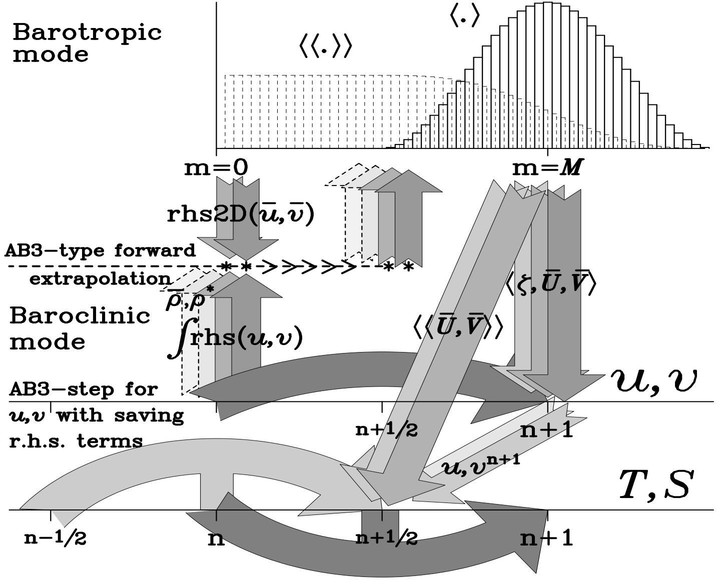
Song & Haidvogel (1994), θ_s=6, θ_s=0.15, h_c=50



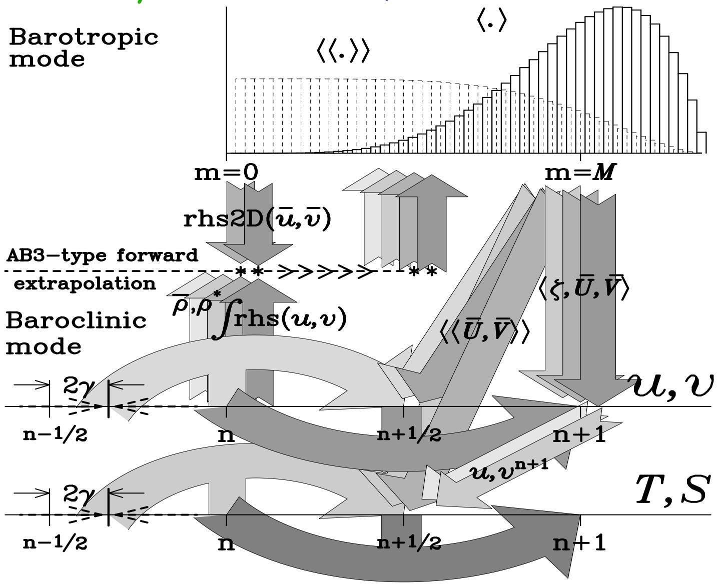
Lemarié et. al. (2011), θ_s=10, θ_s=2, h_c=400

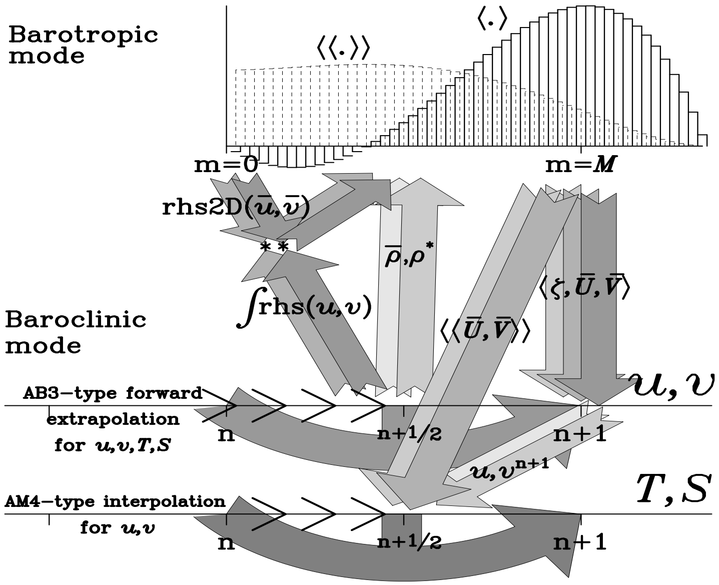
Four variants of ROMS kernel

Rutgers

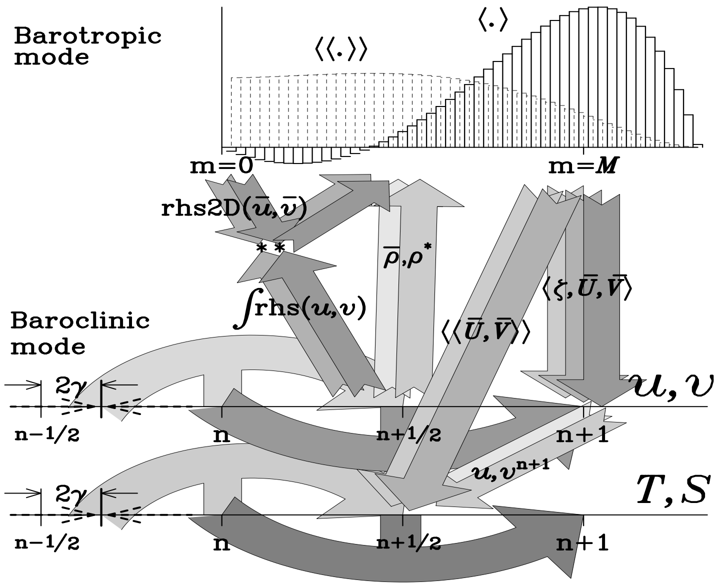


AGRIF/ old UCLA; CROCO





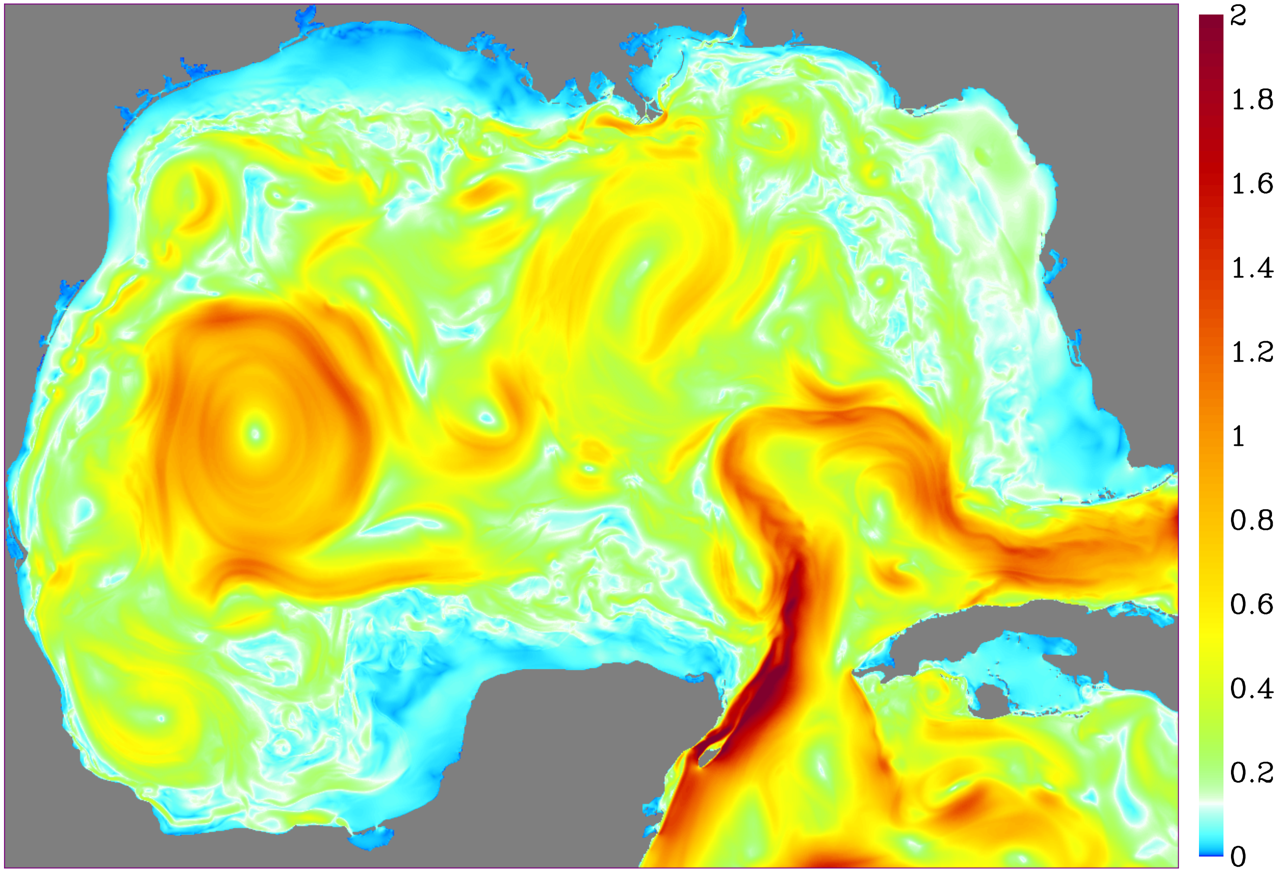
Non-hydrostatic code prototype



UCLA (current)

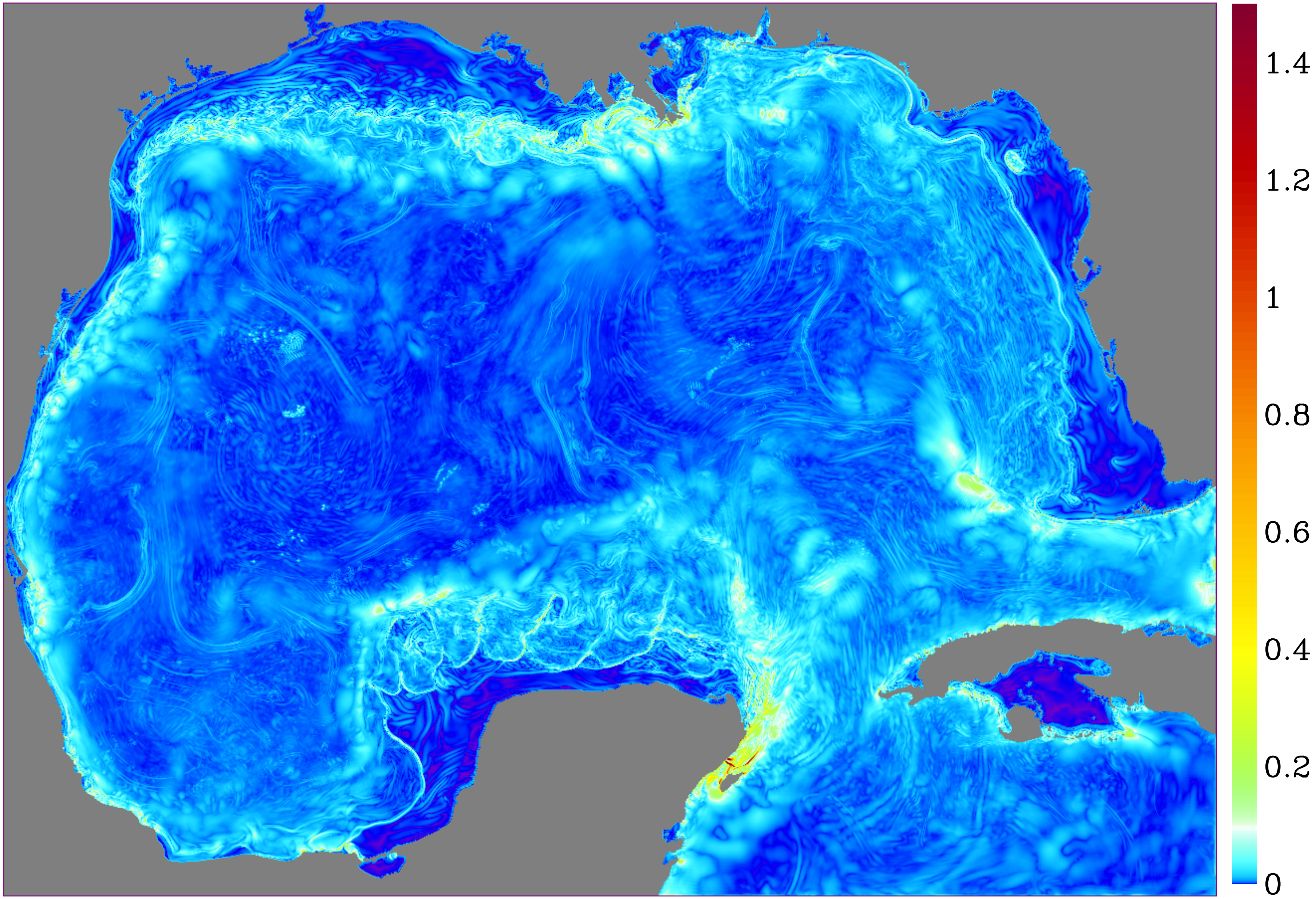
Adaptively implicit vertical advection

The motivation is to keep max horizontal Courant number at healthy 0.6

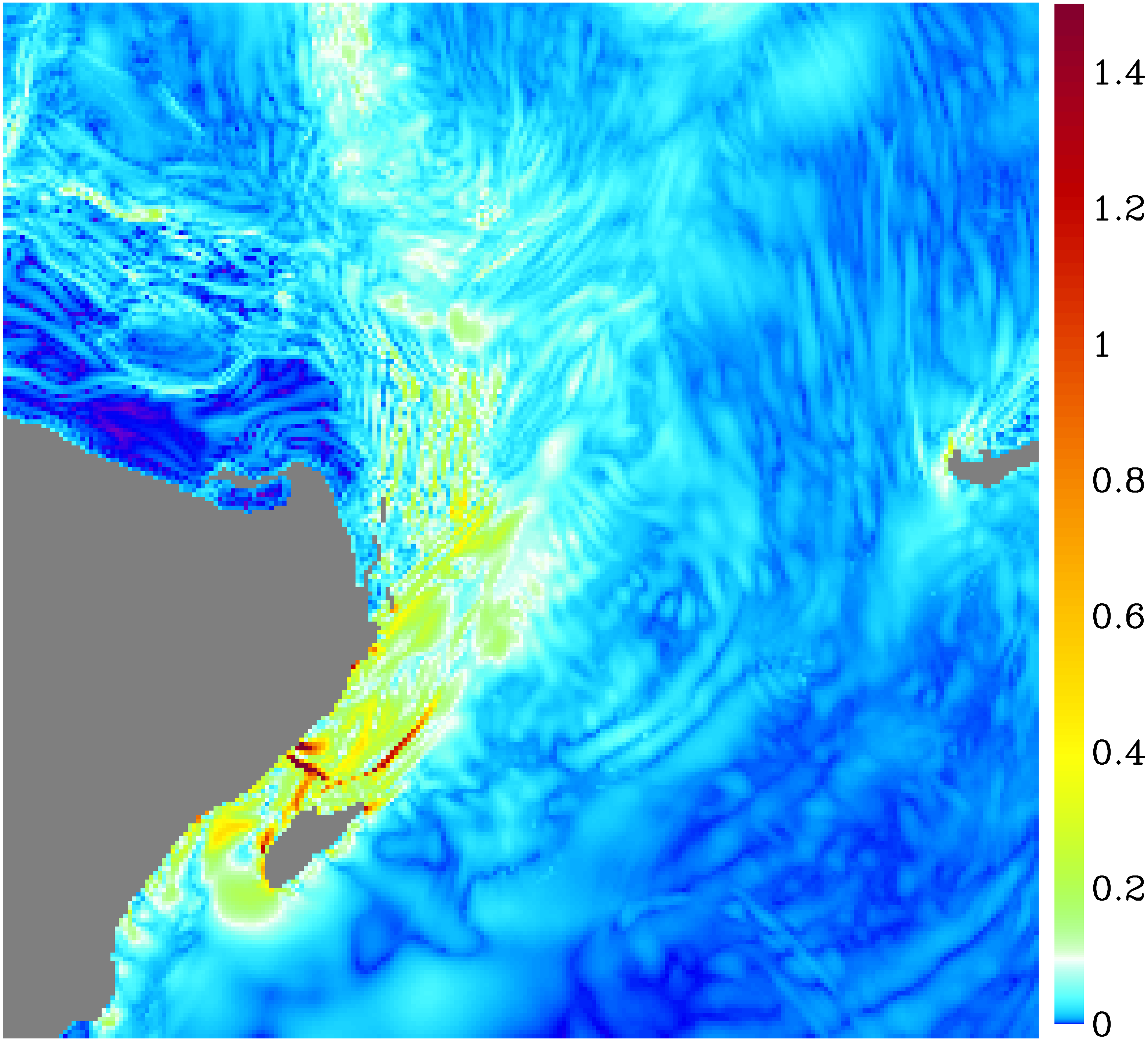


horizontal speed, max in each vertical column

...but it is not always possible.

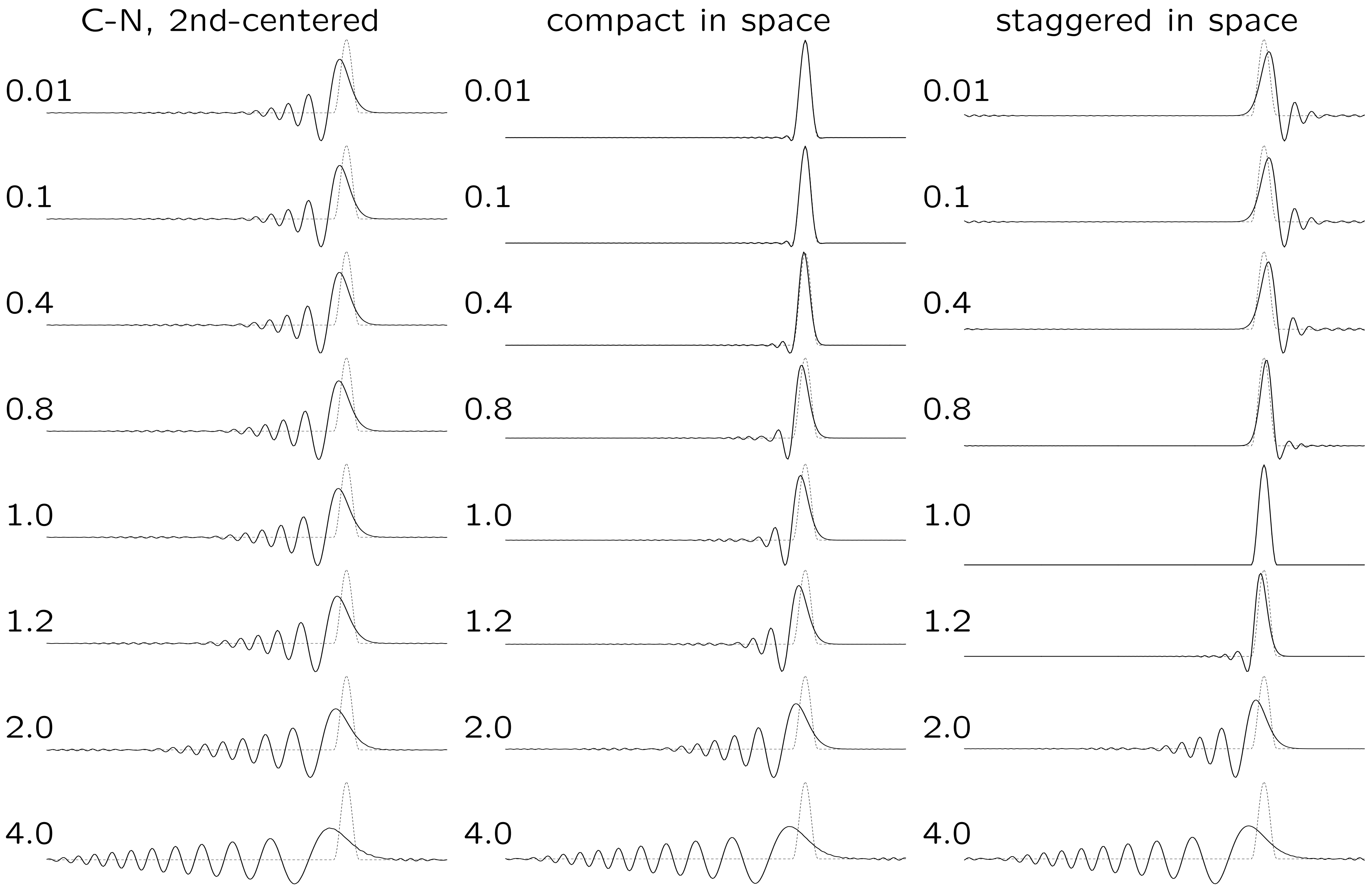


vertical Courant number, max in each vertical column


1.4
1.2
1
0.8
0.6
0.4
0.2
0

Why adaptive?

Why not to use an implicit method from a textbook?



Advection and dispersive spreading of a narrow pulse by non-dissipative, unconditionally stable implicit schemes using different Courant number regimes. Equal-weight $\theta = 1/2$ Crank-Nicolson stepping in all three cases. *Left column* centered second-order differencing in space; *middle* compact-centered (fourth-order); *right* staggered in space, centered around $(x_j - \Delta x/2, y_n + \Delta t/2)$.

Adaptively implicit vertical advection operator:

Vertical fluxes for the tracer and velocity fields are discretized involving the advected field at $n+1$ time step which are yet unknown,

$$FC_{k+{}^1\!/_2} = W_{k+{}^1\!/_2} \cdot \mathscr{Q}\left(q_k^{n+1}, q_{k\pm 1}^{n+1}, q_k^{n+{}^1\!/_2}, q_{k\pm 1}^{n+{}^1\!/_2}, \ldots\right)$$

rearrange by splitting vertical velocity into two parts,

$$W_{k+{}^1\!/_2} = W_{k+{}^1\!/_2}^{(e)} + W_{k+{}^1\!/_2}^{(i)}, \qquad \forall k = 0, 1, \ldots, N$$

where $W_{k+{}^1\!/_2}^{(i)}$ is for terms involving $q_k^{n+1}, q_{k\pm 1}^{n+1}$ only (*i.e.*, implicit part), while $W_{k+{}^1\!/_2}^{(e)}$ for the remaining $q_k^{n+{}^1\!/_2}, q_{k\pm 1}^{n+{}^1\!/_2}, \ldots$, then:

- **$W^{(e)}$-terms are computed within r.h.s via standard algorithm**
- **$W^{(i)}$-terms are integrated into the vertically implicit operator**

Combined advection-diffusion with upstream treatment of the implicit advection part

$$FC^{(i)}_{k+{}^1\!/_2} = W^{(i)}_{k+{}^1\!/_2} \cdot \begin{cases} q_k^{n+1}, & \text{if} \quad W^{(i)}_{k+{}^1\!/_2} > 0 \\ q_{k+1}^{n+1}, & \text{if} \quad W^{(i)}_{k+{}^1\!/_2} < 0 \end{cases}$$

$k = N$, uppermost grid box,

$$H_N^{n+1} q_N^{n+1} = H_N^n q_N^n + \Delta t \cdot \mathrm{rhs}'_N + \Delta t \cdot \mathit{SRFRC} - \Delta t A_{N-{}^1\!/_2} \frac{q_N^{n+1} - q_{N-1}^{n+1}}{\Delta z_{N-{}^1\!/_2}}$$
$$+\Delta t \left[\max\left(W^{(i)}_{N-{}^1\!/_2}, 0\right) q_{N-1}^{n+1} + \min\left(W^{(i)}_{N-{}^1\!/_2}, 0\right) q_N^{n+1} \right]$$

$k = 2, ..., N-1$

$$H_k^{n+1} q_k^{n+1} = H_k^n q_k^n + \Delta t \cdot \mathrm{rhs}'_k + \Delta t A_{k+{}^1\!/_2} \frac{q_{k+1}^{n+1} - q_k^{n+1}}{\Delta z_{k+{}^1\!/_2}}$$
$$-\Delta t \left[\max\left(W^{(i)}_{k+{}^1\!/_2}, 0\right) q_k^{n+1} + \min\left(W^{(i)}_{k+{}^1\!/_2}, 0\right) q_{k+1}^{n+1} \right]$$
$$-\Delta t A_{k-{}^1\!/_2} \frac{q_k^{n+1} - q_{k-1}^{n+1}}{\Delta z_{k-{}^1\!/_2}}$$
$$+\Delta t \left[\max\left(W^{(i)}_{k-{}^1\!/_2}, 0\right) q_{k-1}^{n+1} + \min\left(W^{(i)}_{k-{}^1\!/_2}, 0\right) q_k^{n+1} \right]$$

$k = 1$, bottom grid box,

$$H_1^{n+1} q_1^{n+1} = H_1^n q_1^n + \Delta t \cdot \mathrm{rhs}'_1 + \Delta t A_{{}^3\!/_2} \frac{q_2^{n+1} - q_1^{n+1}}{\Delta z_{{}^3\!/_2}}$$
$$-\Delta t \left[\max\left(W^{(i)}_{{}^3\!/_2}, 0\right) q_1^{n+1} + \min\left(W^{(i)}_{{}^3\!/_2}, 0\right) q_2^{n+1} \right]$$

The W-splitting works as follows: Compute $W_{k+1/2}$ the standard way; also compute finite-volume Courant number $\alpha_{i,j,k}$ at every grid box H_k as the sum of outgoing fluxes normalized by Δt and grid-box volume,

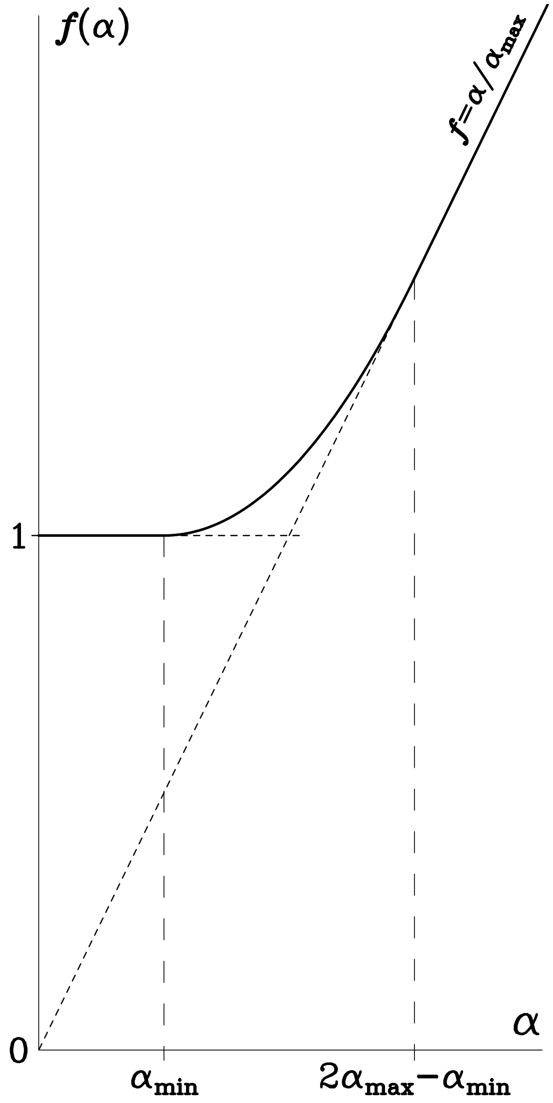
$$\alpha_{i,j,k} = \frac{\Delta t}{\Delta V_{i,j,k}} \cdot \Big[\max(FlxU_{i+1/2,j,k},0) - \min(FlxU_{i-1/2,j,k},0) \\ + \max(FlxV_{i,j+1/2,k},0) - \min(FlxV_{i,j-1/2,k},0) \\ + \max(W_{i,j,k+1/2},0) - \min(W_{i,j,k-1/2},0) \Big]$$

then the explicit part

$$W^{(e)}_{k+1/2} = \frac{W_{k+1/2}}{f(\alpha^*)}, \quad \text{where} \quad \begin{cases} \alpha^* = \alpha_k & \text{if} \quad W_{k+1/2} > 0 \\ \alpha^* = \alpha_{k+1} & \text{if} \quad W_{k+1/2} < 0 \end{cases}$$

and the limiting function

$$f(\alpha) = \begin{cases} 1\,, & \text{if} \quad \alpha \le \alpha_{\text{min}} \\ 1 + \dfrac{(\alpha - \alpha_{\text{min}})^2}{4\alpha_{\text{max}}\,(\alpha_{\text{max}} - \alpha_{\text{min}})}\,, & \text{if} \quad \alpha_{\text{min}} < \alpha < 2\alpha_{\text{max}} - \alpha_{\text{min}} \\ \alpha/\alpha_{\text{max}}\,, & \text{if} \quad \alpha \ge \alpha_{\text{max}} \end{cases}$$



made of three segments – constant, parabolic, and linear – smoothly matched to each other; α_{min} control the threshold below which the algorithm is fully explicit; α_{max}, and the maximum allowed Courant number **“never exceed speed”** for the explicit part

The implicit part is the “excess” velocity

$$W^{(i)}_{k+1/2} = W_{k+1/2} - W^{(e)}_{k+1/2}$$

Selectable α_{min} and α_{max} based on consideration of accuracy and numerical stability of the explicit part.

In the actual code all the above – computing $\alpha = \alpha_{i,j,k}$, then $f(\alpha)$ then splitting W is implanted into the computation of W itself, so none of the intermediates is stored as a 3D array.

Prime in rhs'_k means that the usual r.h.s. computed by ROMS code for the corresponding equations, except the replacement $W_{k+1/2} \to W^{(e)}_{k+1/2}$

$A_{k+1/2}$ is vertical viscosity/diffusion coefficient [including the stabilization terms (Lemarié et. al., 2012) in the case when isoneutral lateral diffusion is used]

The above takes into account kinematic b.c. at surface and bottom, $W_{N+1/2} = W_{1/2} = 0$, bottom no-flux b.c. for tracers. There is an extra term for momentum equation associated with bottom drag which also treated implicitly.

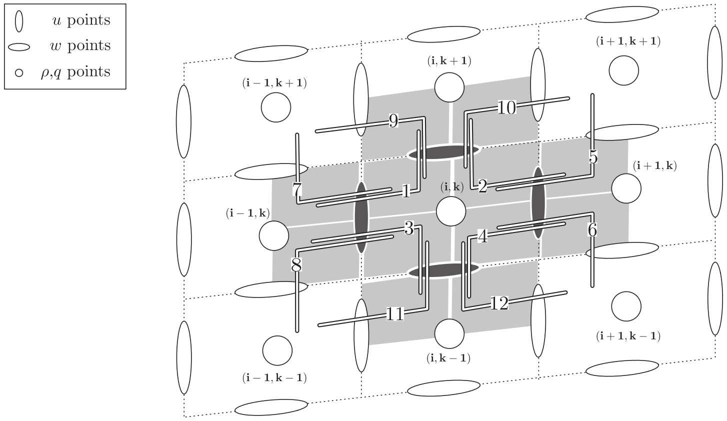
The modified algorithm **retains simultaneous conservation and constancy preservation properties for tracers**, despite the fact that grid box heights change due to changing free surface, $H_k^{n+1} \neq H_k^n$.

The motivation for using upstream discretization for the implicit part comes from the fact that it is monotonic, hence will not cause oscillation. Unavoidably it is diffusive, however this choice is justified by the observation that in practical model solutions large vertical velocities occur only in places with vanishing (or even unstable) stratification and, consequently, already large mixing set by the vertical parameterization scheme.

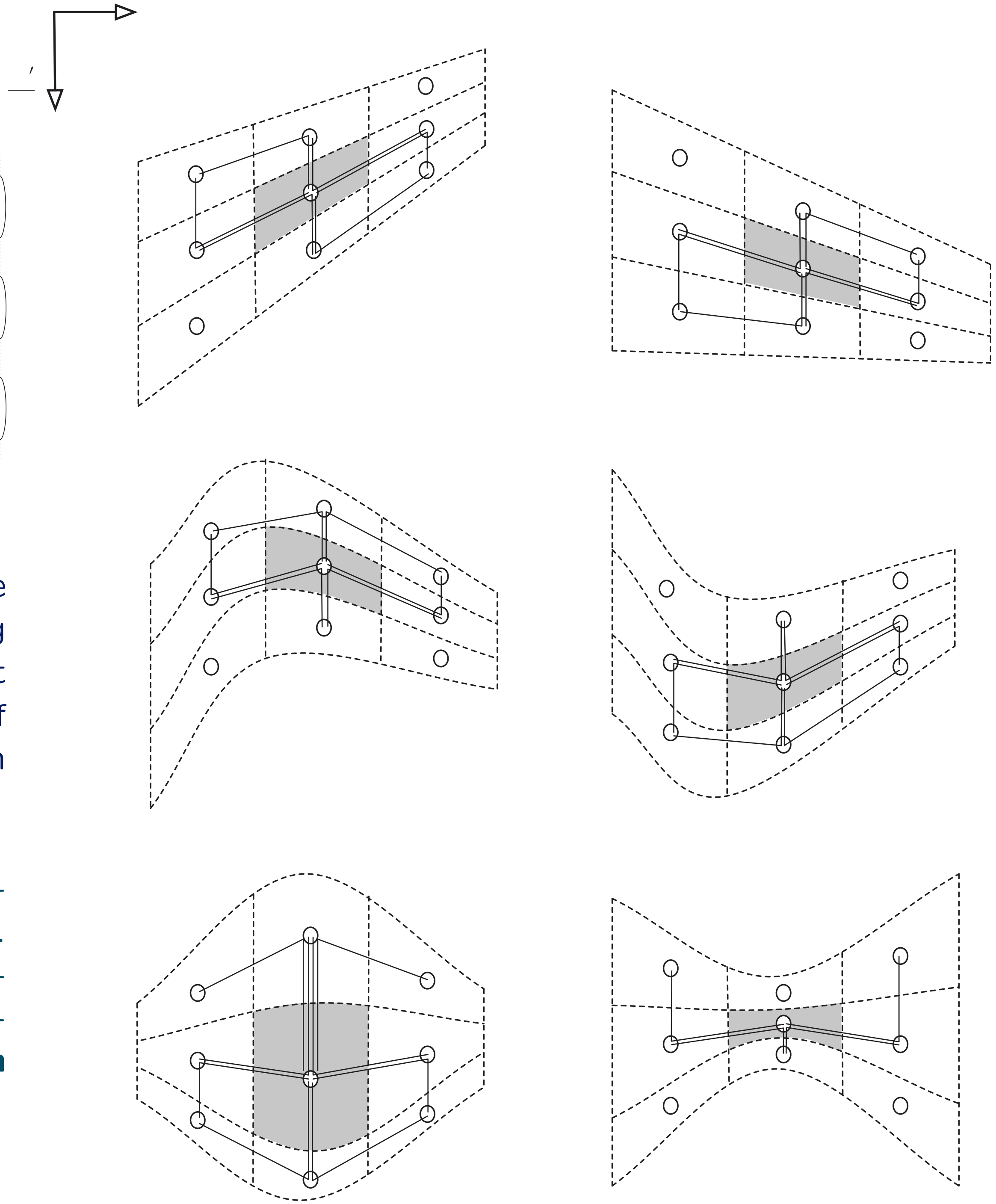
Well posed, diagonally dominant discrete system.

Rotated diffusion in sigma coordinates

Rotated isoneutral diffusion: explaining switching triads

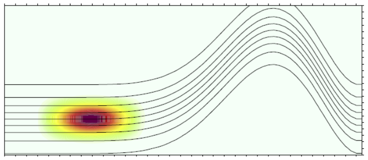


above: Griffies et. al. (1998) discrete isoneutral diffusion operator using all 12 triads around tracer point (they are arranged into two pairs of fluxes, horizontal and ”vertical”, each involving 4 triads)

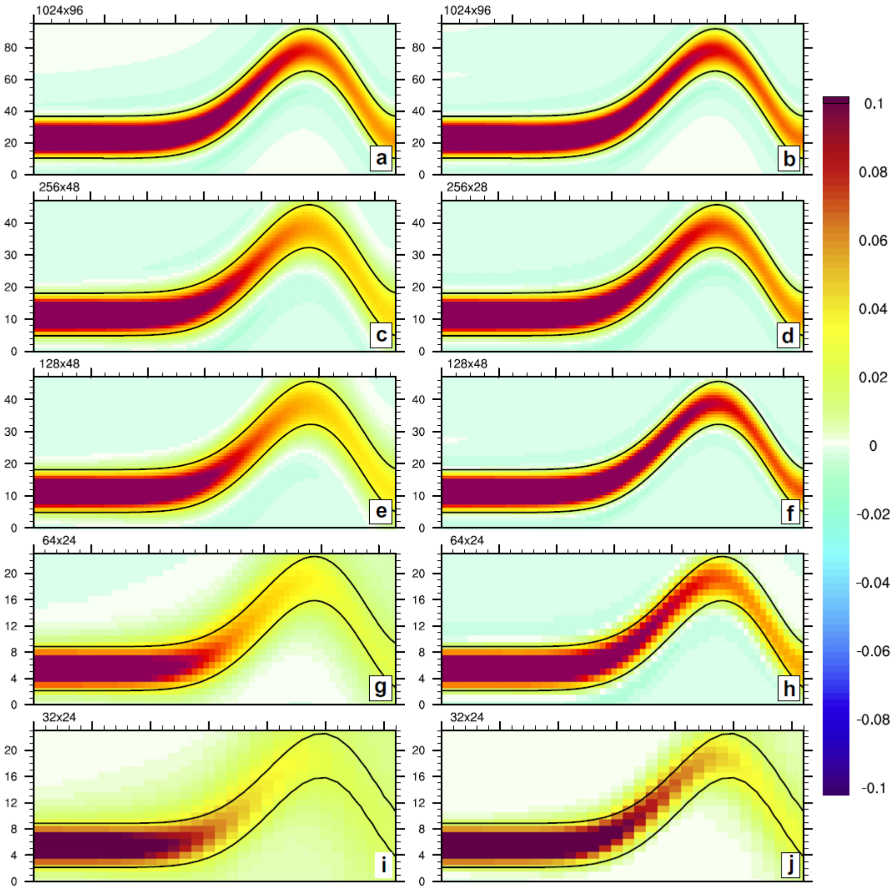


right: Lemarié et. al. (2012) switching triads in all possible situations. Note that vertical axis here is ”density”, so isoneutral direction is horizontal in all these charts; **only triads with sharp angle** ($< 90^O$) are selected.
”exact” if grid-point slope $s = 1$

Rotated isoneutral diffusion: convergence comparison

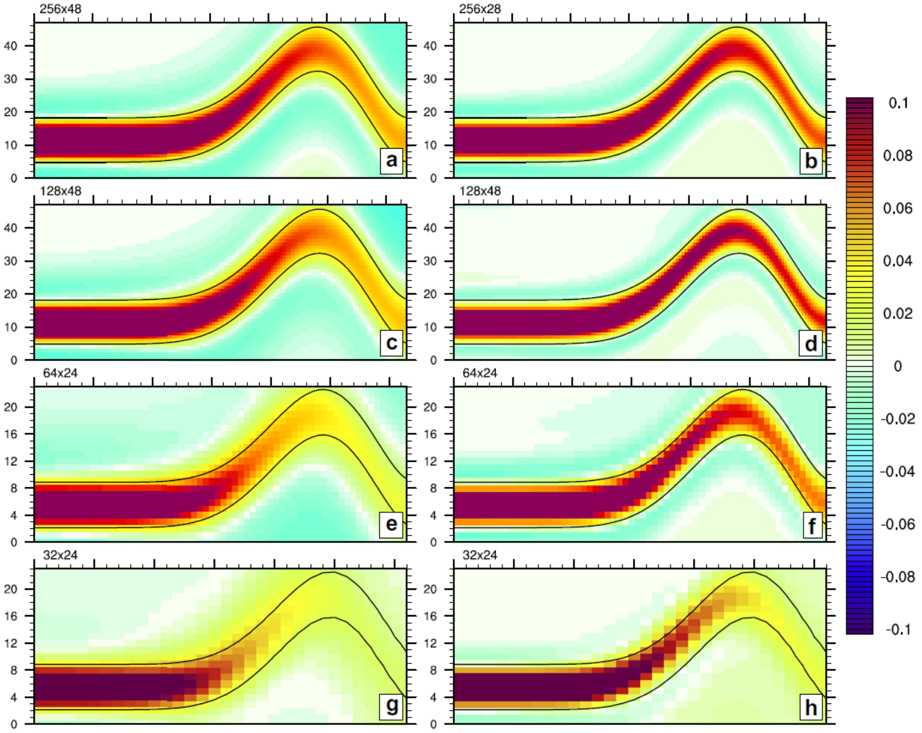


initial condition for smooth field test problem



non-switched switched

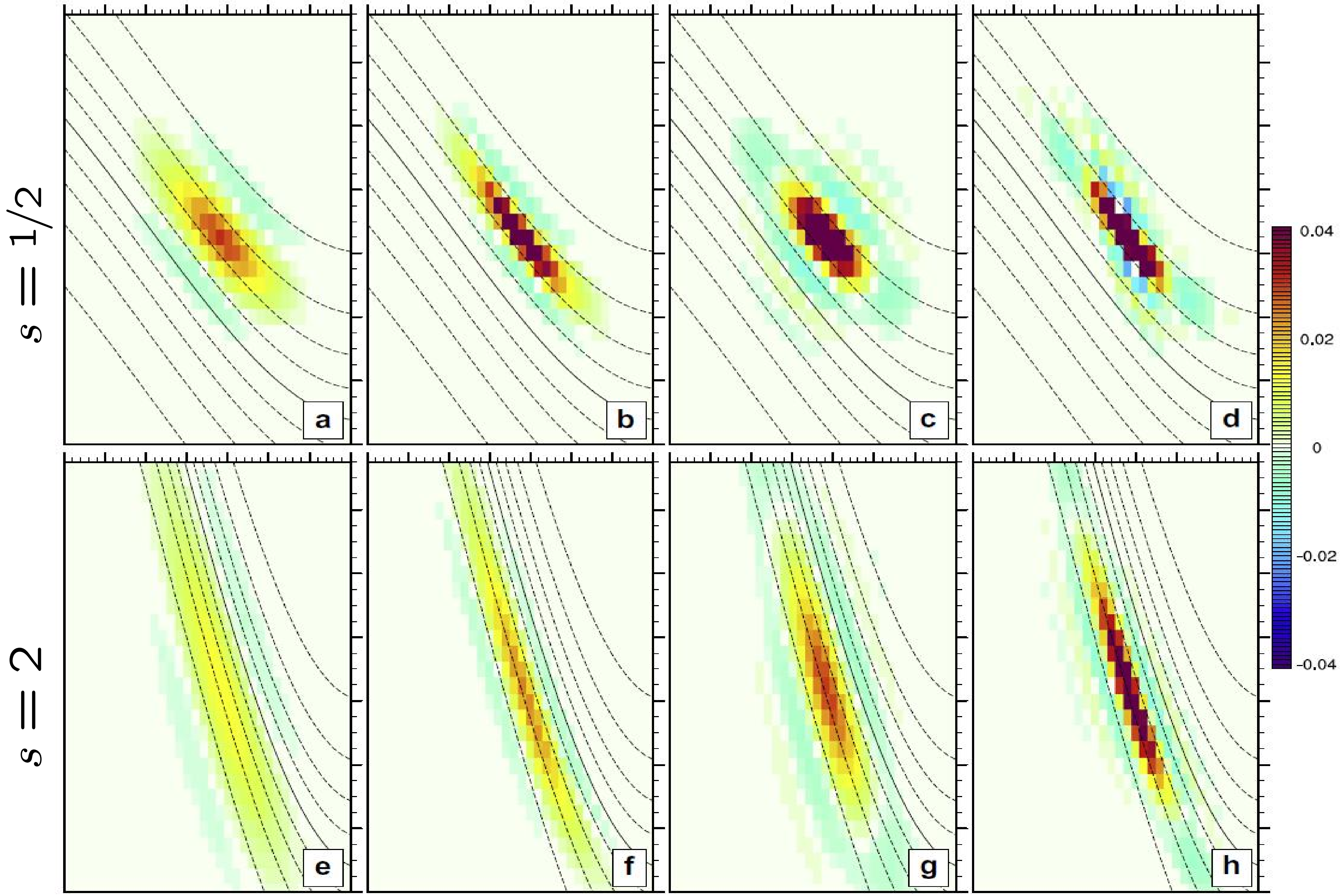
diffusion



non-switched switched

hyper-diffusion

Rotated isoneutral diffusion: spread of a single dot



non-switched switched non-switched switched

diffusion **hyper-diffusion**

“If-less” KPP via integral criterion alternative to Ri_b

Criterion for finding h_{bl}: Define surface PBL as an integral layer within which net production of turbulence due to shear-layer instability is balanced by dissipation due to stratification,

$$Cr(z) = \int_{z}^{\text{surface}} \mathscr{K}(z) \left\{ \left| \frac{\partial \mathbf{u}}{\partial z} \right|^2 - \frac{N^2}{\mathtt{Ri}_{\text{cr}}} - C_{\text{Ek}} \cdot f^2 \right\} dz' + \frac{V_t^2(z)}{z}$$

and search for crossing point $Cr(z) = 0$.

$$N^2 = -\frac{g}{\rho_0} \cdot \left. \frac{\partial \rho}{\partial z} \right|_{ad}$$ is B-V frequency ($ad \equiv$ adiabatic);

f is Coriolis parameter; C_{Ek} is a nondimensional constant;

$V_t^2(z)$ is unresolved turbulent velocity shear (same as in LMD94);

Integration Kernel $\mathscr{K}(z) = \dfrac{\zeta - z}{\epsilon h_{\text{bl}} + \zeta - z}$ is to ignore contribution from near-surface sublayer ϵh_{bl} where M-O similarity law is not valid (plays the same role as to distinguish between ρ_{ref} vs. ρ_{surf} in Ri_b of LMD94). ζ is free surface; $\epsilon = 0.1$.

“If-less” KPP...

- Same result as Ri_b the case of linear velocity profile, but otherwise

$$\int_{z'}^{z''} \left|\frac{\partial \mathbf{u}}{\partial z}\right|^2 dz \geq \frac{|\mathbf{u}'' - \mathbf{u}'|^2}{z'' - z'}$$

- $Cr(z)$ is **monotonic** for Ekman spiral $\Rightarrow$ no sudden jumps of h_{bl}
- Numerically more attractive, as $\mathbf{u}(z)$ and $\rho(z)$ can be reconstructed as continuous functions
- Avoids introduction of *reference* potential density: basically integration square of Brunt-Väisäla frequency. Allows formalism of *adiabatic* derivatives and differences to achieve monotonicity
- Correct account for thermobaric effect: Bill Large: to determine extent of BL one must bring water parcel **from reference depth to** $z = -h_{\mathrm{bl}}$ and compare its density with the ambient fluid **there**. We never did it this way in ROMS community (?)
- Avoids ambiguity for merging top and bottom BLs

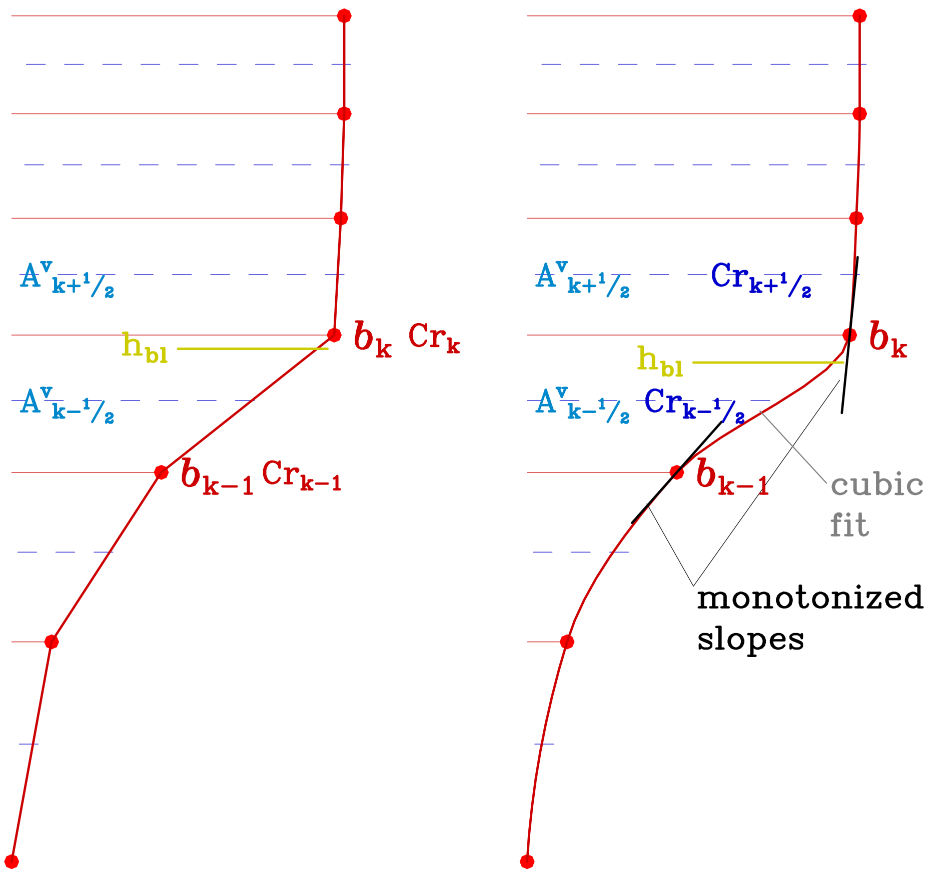
“If-less” KPP...

Pure physical limits: destabilizing vs. stabilizing effects:

- balance $\left|\frac{\partial \mathbf{u}}{\partial z}\right|^2$ vs. $\frac{N^2}{\mathtt{Ri}_{\mathsf{cr}}}$ $\Rightarrow$ shear layer instability

- $\left|\frac{\partial \mathbf{u}}{\partial z}\right|^2$ vs. $C_{\mathsf{Ek}} \cdot f^2$ $\Rightarrow$ turbulent Ekman layer

- *negatively* forced $\frac{N^2}{\mathtt{Ri}_{\mathsf{cr}}}$ vs. V_t^2 $\Rightarrow$ free convection



Monotonized reconstruction to compute $V_t^2{}_{k+1/2}$ and $\rho_{k+1/2}$, but **not** to interpolate Cr to find h_{bl}: because

$$Cr \sim w_s\sqrt{N^2} - N^2 d$$

$Cr(z)$ is **not monotonic** near

$$z = -h_{\mathsf{bl}}$$

even if $\rho(z)$ and $u, v(z)$ are

$\Rightarrow$ quadratic (cubic) interpolation for Cr is dangerous

Overall this is by far the largest cause of numerical sensitivities in KPP.

“If-less” KPP... Monin-Obukhov depth limit $h_{\mathsf{bl}} \le h_{\mathsf{MO}} = \dfrac{C^{\mathsf{MO}} \cdot u_*^3}{\kappa \cdot B_{\mathsf{f}}}$ if $B_{\mathsf{f}}(z) > 0$.

Because of solar radiation absorption, buoyancy forcing $B_{\mathsf{f}} = B_{\mathsf{f}}(z)$ increases with depth, possibly changing sign *from unstable to stable*
⇒ a case when $B_{\mathsf{f}}\left(-\text{overestimated } h_{\mathsf{bl}}\right) > 0$, but $B_{\mathsf{f}}\left(-h_{\mathsf{MO}}\right) < 0$
⇒ **hysteresis** and oscillations in h_{bl}

solution # 1: use $B_{\mathsf{f}} = B_{\mathsf{f}}\left(-h_{\mathsf{MO}}\right)$ in computation of h_{MO}, i.e. implicit search for k enclosing z^*, such that

$$z_k \le z^* \le z_{k+1} \qquad \text{and} \qquad h_{\mathsf{MO}}\left(z_k\right) \le |z^*| \le h_{\mathsf{MO}}\left(z_{k+1}\right)$$

then solve
$$\frac{h_{\mathsf{MO}k}\left(z_{k+1} - z^*\right) + h_{\mathsf{MO}k+1}\left(z^* - z_k\right)}{z_{k+1} - z_k} + z^* = 0$$

resulting in
$$h_{\mathsf{MO}} = -z^* = \frac{\frac{C^{\mathsf{MO}} u_*^3}{\kappa}\left({B_{\mathsf{f}}}'_{k+1} z_{k+1} - {B_{\mathsf{f}}}'_k z_k\right)}{{B_{\mathsf{f}}}'_{k+1} {B_{\mathsf{f}}}'_k \left(z_{k+1} - z_k\right) + \frac{C^{\mathsf{MO}} u_*^3}{\kappa}\left({B_{\mathsf{f}}}'_k - {B_{\mathsf{f}}}'_{k+1}\right)}$$

above ${B_{\mathsf{f}}}' = \max\left(B_{\mathsf{f}}, 0\right)$; if k not found ⇒ no limit; **no singularity** if either $B_{\mathsf{f}} \to 0$; limit applied **outside** $B_{\mathsf{f}} > 0$ **logic**: it is already taken into account in computing h_{MO}; **since** h_{bl} **is not involved ⇒ no possibility of hysteresis**

solution # 2: Eliminate M-O limit altogether.

“If-less” KPP... Ekman depth limitation: $h_{\mathsf{bl}} \leq h_{\mathsf{Ek}} = 0.7\frac{u_*}{f}$ applicable **only for stable** boundary layer; should be $h_{\mathsf{bl}} = h_{\mathsf{Ek}}$ for **neutral** forcing and stratification

Length $\mathscr{L} = u_*/f$ and velocity $\mathscr{U} = u_*$ are natural scaling parameters for neutrally stratified problem

$$i \cdot f\mathbf{u} = \frac{\partial}{\partial z}\left(w_m|z|\frac{\partial\mathbf{u}}{\partial z}\right)$$

where $\mathbf{u} = u + iv$, and $w_m \equiv \kappa u_*$, and κ is von Karman constant.

- Most vertical mixing schemes are "Coriolis-blind" any way.

- Coriolis effect plays no role in determining h_{bl} via bulk Ri criterion; h_{Ek}-limit is applied *a‘posteriori*, and only for stable buoyancy forcing.

- Because of light absorption, **stability increases downward resulting in hysteresis** if $B_{\mathsf{f}}\,(\text{unlimited } h_{\mathsf{bl}}) > 0$, but $B_{\mathsf{f}}\,(h_{\mathsf{Ek}}) < 0$ which is manifested by h_{bl} oscillations and jumps

??? integrate h_{Ek}-limit into KPP BL criterion, balance

$$\int\left|\frac{\partial\mathbf{u}}{\partial z}\right|^2 dz' \qquad \text{vs.} \qquad \int f^2 dz' \qquad ???$$

"If-less" KPP... Modified Ekman problem

$$i\cdot f\mathbf{u} = \frac{\partial}{\partial z}\left[w_m \mathscr{L} G\left(\frac{z}{\mathscr{L}}\right)\frac{\partial \mathbf{u}}{\partial z}\right]$$

G is KPP non-dimensional shape function

$$G(\sigma) = |\sigma|\,(1-\sigma)^2 + \begin{cases} \frac{(\sigma-\sigma_0)^2}{2\sigma_0}, & \sigma < \sigma_0 \\ 0 & \text{otherwise} \end{cases} \qquad \sigma_0 = 0.1$$

B.C.:

$$w_m \mathscr{L} G\left(\frac{z}{\mathscr{L}}\right)\frac{\partial \mathbf{u}}{\partial z}\bigg|_{z=0} = u_*^2 \mathbf{1}_\tau \qquad \Rightarrow \qquad \frac{\partial \mathbf{u}}{\partial z}\bigg|_{z=0} = \frac{u_* \mathbf{1}_\tau}{\kappa \mathscr{L} \sigma_0/2}$$

$$\mathbf{u} = 0, \text{ if } z < -\mathscr{L}$$

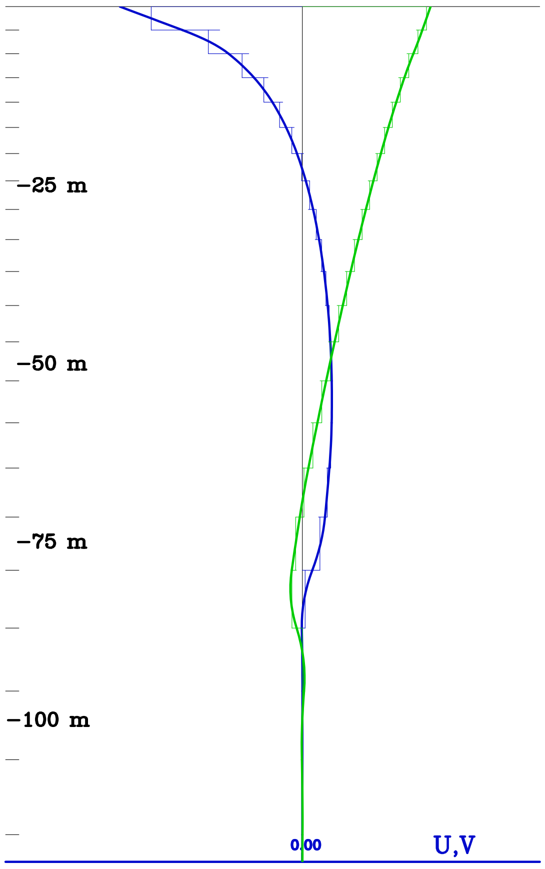
Nondimensionalization: Postulate that depth of generated this way boundary layer is equal to Ekman length and introduce scaling,

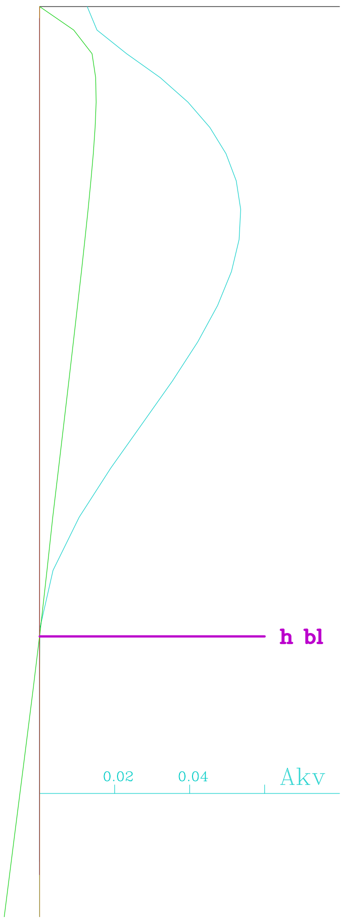
$$z = \mathscr{L}\sigma = \sigma \cdot 0.7 u_*/f \qquad\qquad \mathbf{u} = u_* \cdot \tilde{\mathbf{u}}\,,$$

hence

$$\frac{\partial}{\partial \sigma}\left(G(\sigma)\frac{\partial \tilde{\mathbf{u}}}{\partial \sigma}\right) = i\cdot\frac{\kappa}{0.7}\tilde{\mathbf{u}}\,, \qquad \frac{\partial \tilde{\mathbf{u}}}{\partial \sigma}\bigg|_{\sigma=0} = \frac{2}{\kappa\sigma_0}\,, \qquad \tilde{\mathbf{u}}\bigg|_{\sigma<-1} = 0$$

everything has been scaled out.





Recognize Coriolis force as a *stabilizing* factor (balancing vertical shear production), construct

$$Cr(z) = \int_{z}^{\text{surf}} \mathscr{K}(z') \left\{ \left| \frac{\partial \mathbf{u}}{\partial z} \right|^2 - C_{\text{Ek}} \cdot f^2 \right\} dz'$$

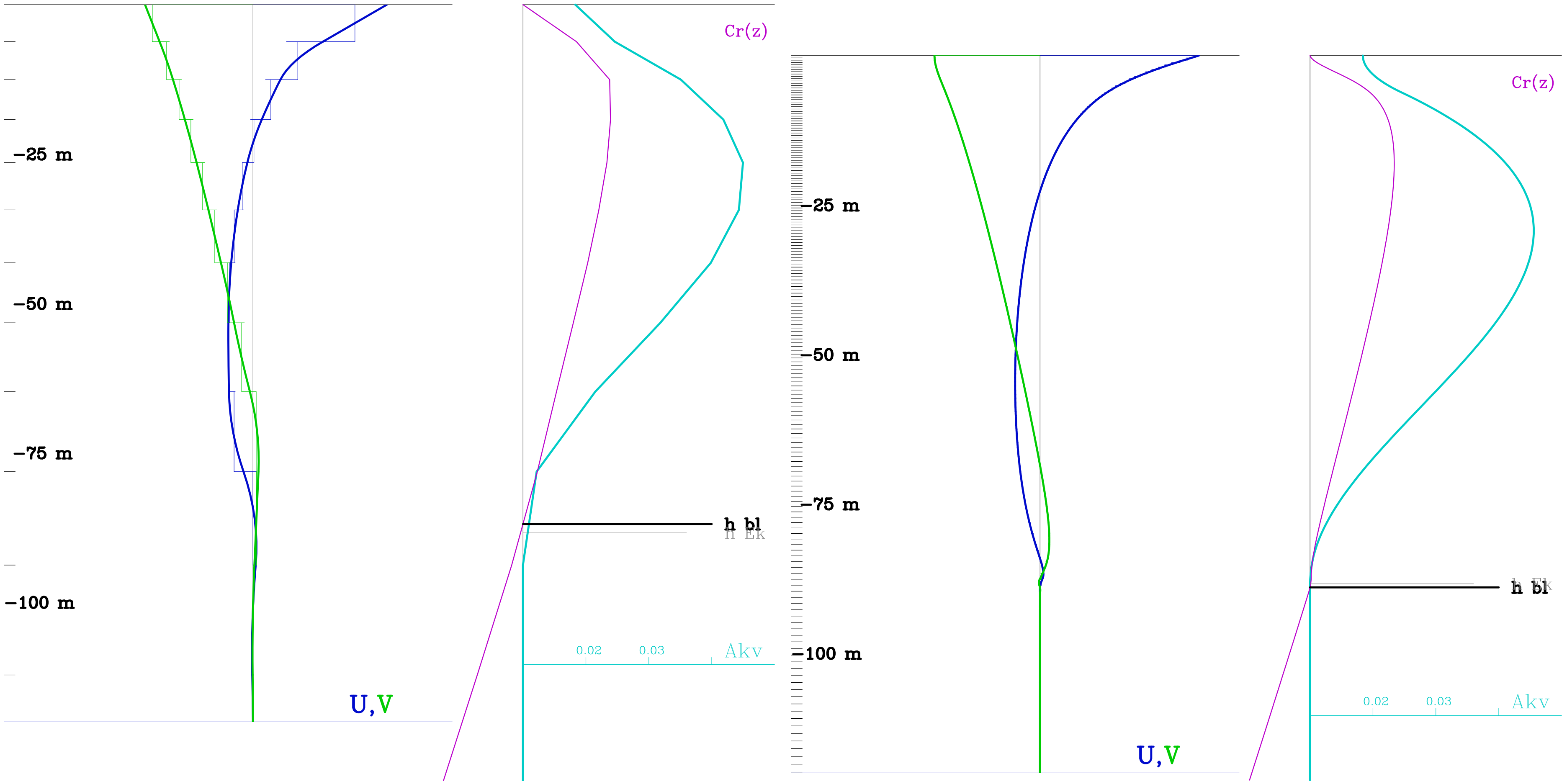
apply the same scaling

$$\widetilde{Cr}(\sigma) = \frac{1}{(0.7)^2} \int_{\sigma}^{0} \mathscr{K}(\sigma) \left\{ \left| \frac{\partial \tilde{\mathbf{u}}}{\partial \sigma} \right|^2 - C_{\text{Ek}} \right\} d\sigma'$$

and demand that $\widetilde{Cr}(-1) = 0$
which can be achieved by tuning

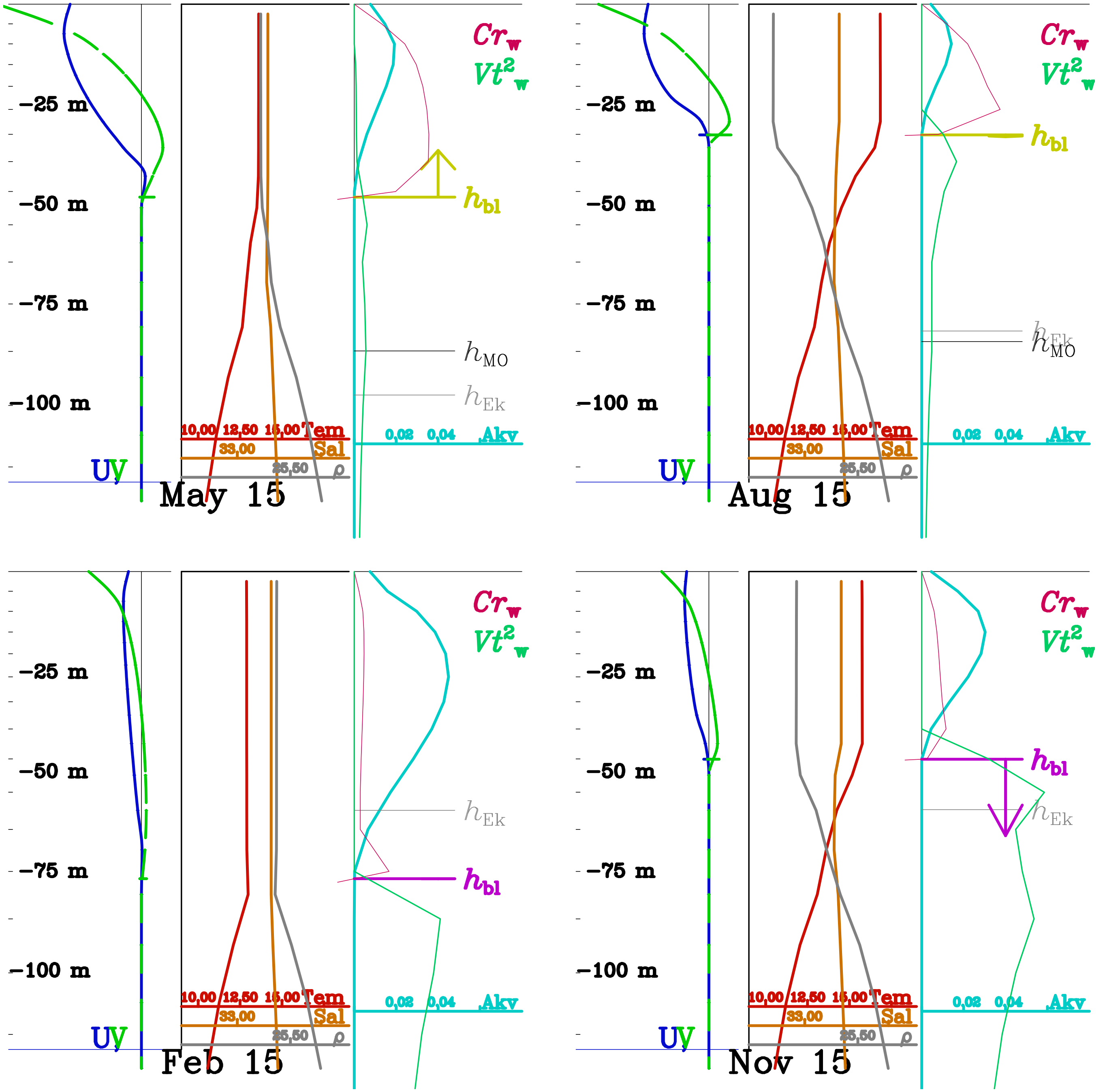
$C_{\text{Ek}} = 258$

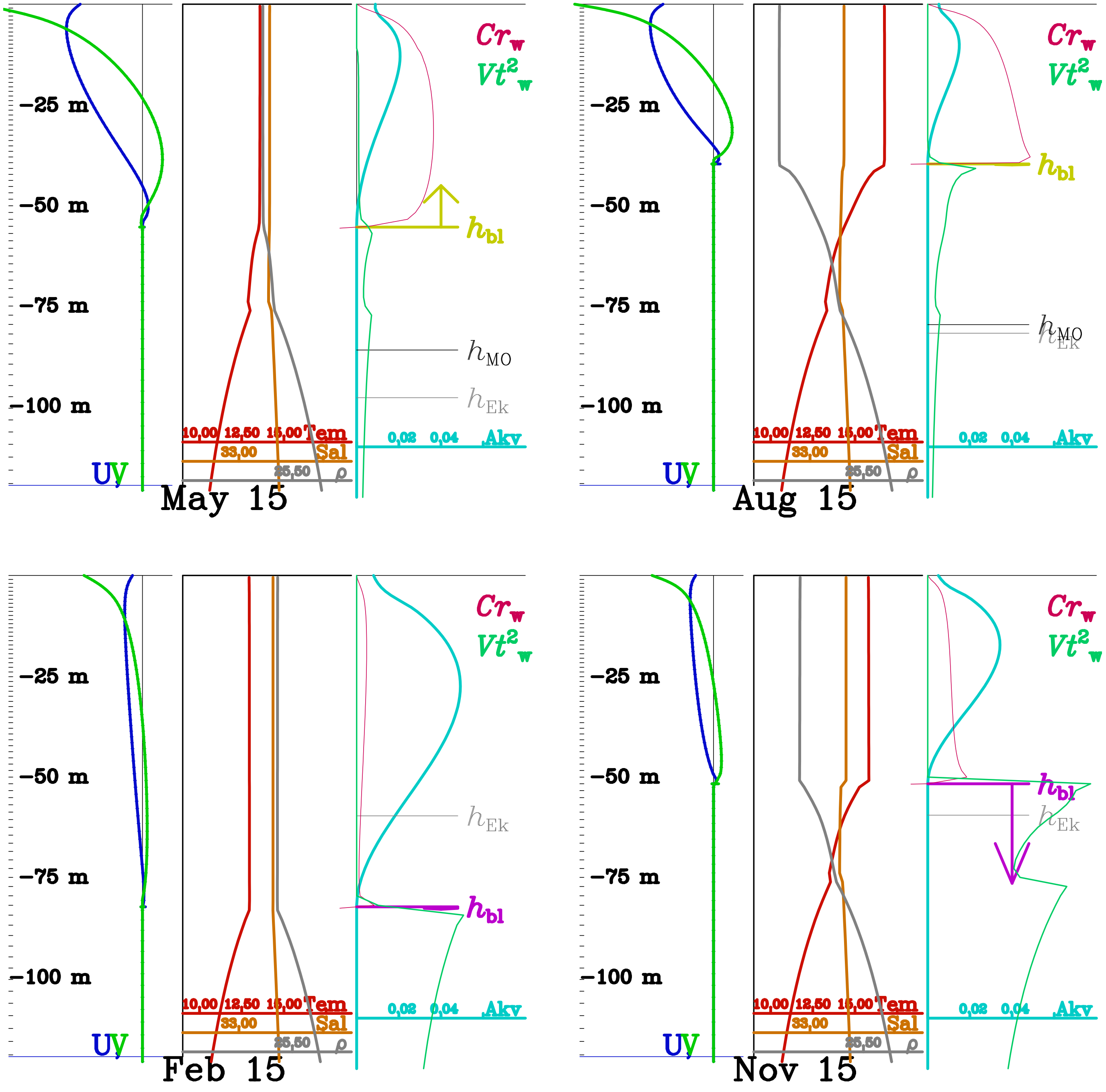
provided that $\mathscr{K}(\sigma) = |\sigma|/(|\sigma| + \epsilon)$
where $\epsilon = 0.1$



Coarse, $N = 32$ and fine, $N = 512$ resolution. h_{Ek} is shown for reference only and does not participate in determining h_{bl}.

- **presence of $\mathscr{K}(\sigma)$ is essential for convergence**
- **overall extremely robust**


-25 m
-50 m
-75 m
-100 m
UV
10,00 12,50 15,00 Tem
33,00 Sal
25,50 ρ
May 15
0,02 0,04 Akv
Cr_w
Vt^2_w
h_{bl}
h_{MO}
h_{Ek}
-25 m
-50 m
-75 m
-100 m
UV
10,00 12,50 15,00 Tem
33,00 Sal
25,50 ρ
Aug 15
0,02 0,04 Akv
Cr_w
Vt^2_w
h_{bl}
h_{Ek}
h_{MO}
-25 m
-50 m
-75 m
-100 m
UV
10,00 12,50 15,00 Tem
33,00 Sal
25,50 ρ
Feb 15
0,02 0,04 Akv
Cr_w
Vt^2_w
h_{Ek}
h_{bl}
-25 m
-50 m
-75 m
-100 m
UV
10,00 12,50 15,00 Tem
33,00 Sal
25,50 ρ
Nov 15
0,02 0,04 Akv
Cr_w
Vt^2_w
h_{bl}
h_{Ek}



May 15

Aug 15

Feb 15

Nov 15

Poor Man's Computing: Overlooked and underutilized resources

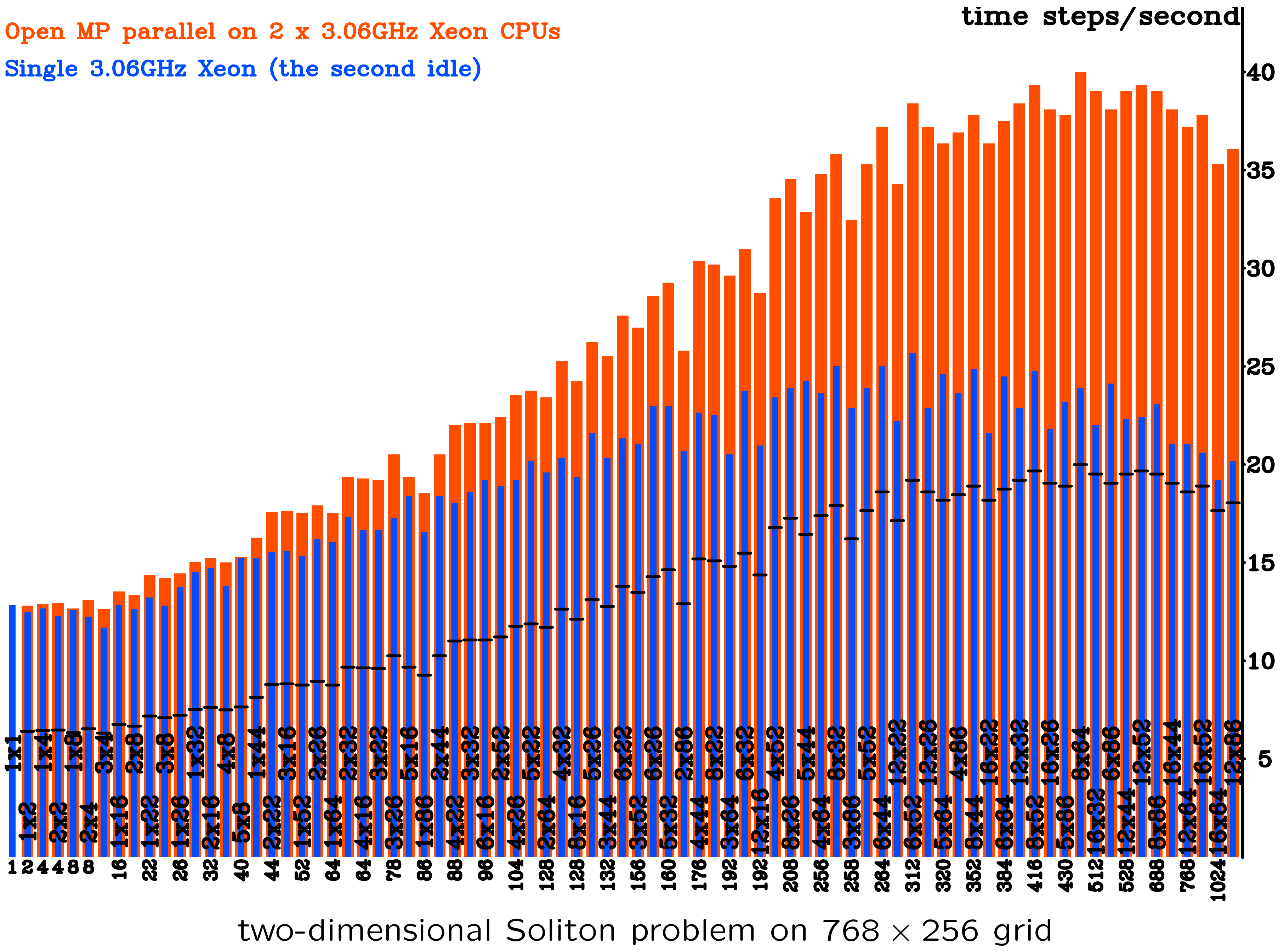
From its day 1 ROMS started as a coarse-grain multi-threaded (proprietary SGI directives, later OpenMP standard) parallel code with 2D subdomain decomposition (tiling) intended for SMP architecture.

- Loop-based parallelization approach **was rejected** from the start. Tiling with **ability to set number of tiles independently** from the number of CPUs.
- **False sharing avoidance** was the key for success/failure back then.

Departure SGI Origin 2000 *de-facto* downplayed the importance of SMP computing. MPI capability was added, but OpenMP always kept.

- SMPs are coming back in form of multi-core CPUs.
- ROMS in OpenMP mode optimally tiled for cache utilization **easily outperforms** the same ROMS code run as MPI by at least 2:1 ratio *within* a single SMP computer. Same applies to even a single-processor machine, tiling vs. single block.
- We knew this for over 10 years, others have no means to discover.
- Efficient Open MP played a key role (at least one of the) in successful proliferation.
- However, Open MP is limited to a single node only. Can we build a **dream code** which combines OpenMP-level single-processor performance with MPI scaling on multiple nodes?

Cache effects on SMP computer, back in 2004



Poor Man’s Computing: Why it is? Is it still relevant today?

This is why:

CPU family	Prescott Pentium IV HT	Kentsfield Core2 Quad	Nehalem Core i7 920	SandyBridge E Core i7 3820	IvyBridge E Core i7 4820K	Haswell E Core i7 5930K
year introduced	2004	2007	Nov. 2008	late 2011	2013	Aug. 2014
CPU clock speed	3.2GHz	2.4GHz	2.66GHz	3.2GHz	3.7GHz	3.5GHz
cores (threads)	1(2)	4	4(8)	4(8)	4(8)	6(12)
cache	1M L2	8M (4+4)	8M L3	10M	10M	15M
memory type	DDR 400	DDR2 667	DDR3 1333	DDR3 1600	DDR3 1600	DDR4 2400
CAS latency	2	5	8	9	9	15
latency, ns	10	15	12	11.2	11.2	12.5
number of channels	2	2	3	4	4	4
aggregate bandwidth, GB/sec	6.4	10.6	31.8	51.2	51.2	76.8
$\frac{\text{loads or stores}}{\text{per cycle, each core}}$	1	1	1	2	2	2
$\frac{\text{loads or stores}}{\text{per sec, all cores}}$, G/sec	3.2	9.6	10.6	25.6	29.6	42
load-store bandwidth for 8Byte operands, GB/sec	25.6	76.8	84.8	204.8	236.8	336
$\frac{\text{load-store}}{\text{vs. memory}}$ bandwidth ratio	**4**	**7.25**	**2.666**	**4**	**4.625**	**4.375**

Poor Man's continued ...

Most today's Linux clusters are made of 8 ... 64-way SMP nodes

- dual or quad CPU sockets; 6, 8, ..., 16-core CPUs
- shared outermost-level cache in each CPU chip
- shared memory bus by all cores within each CPU socket
- interconnect interface (Infiniband or whatever) is shared by all CPUs/cores on the board and become scarce resource
- hyper-threading is back for Intel chips

Most MPI codes are designed to

- separate communications from computations in time resulting in peak loads on interconnect followed periods of by inactivity
- treat multiple processors (cores) within each physical node as separate compute nodes, ignoring non-uniform topology of the machine, (if any)
- introduce competition for shared interconnect interface
- single subdomain — single processor policy motivated by "perimeter vs. area argument"
- ignore cache effects

MPI with Threads Inside and MPI without them

Now, at last, multiple threads are **officially included** into MPI-2 standard

Replaces `MPI_Init` with `MPI_Init_thread(requested, provided)`, where

$$\texttt{requested}, \texttt{provided} = \begin{cases} 0 = \texttt{MPI\_THREAD\_SINGLE} \text{ means no thread support} \\ 1 = \texttt{MPI\_THREAD\_FUNNELED} \text{ means threads are allowed,} \\ \qquad \text{but only master thread can execute MPI calls} \\ \\ 2 = \texttt{MPI\_THREAD\_SERIALIZED} \text{ multiple threads can do} \\ \qquad \text{MPI calls, but the calls are serialized} \\ \\ 3 = \texttt{MPI\_THREAD\_MULTIPLE} \text{ means multiple threads can} \\ \qquad \text{execute concurent MPI calls} \end{cases}$$

- The original motivation is to allow multiple subdomains for cache management to recover cache efficiency of optimally tiled code.

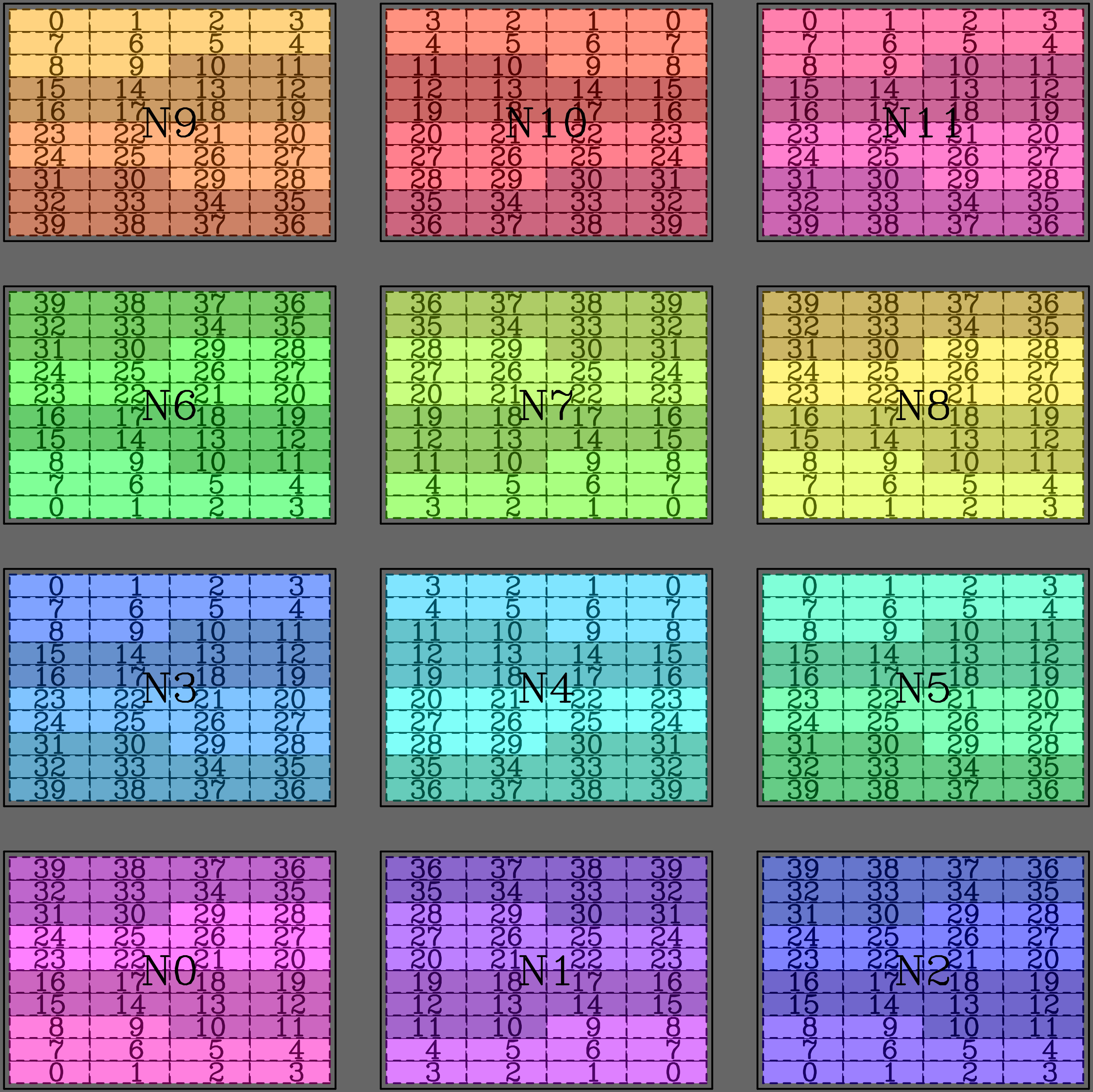
- But it turns out that there is more in it: tiling and threads can be used for tuning of scheduling of messages to alleviate competition for access to Network Interface within SMP nodes, simply put, when one thread sends messages, the other(s) compute and, and vice versa.

Approach:

- relies on the highest level of MPI thread support, `requested, provided` = 3,3
- 2-level 2-dimensional subdomain decomposition: tiles within MPI subdomains;
- once a thread completes working on a tile, it sends/receives relevant messages to its MPI-neighbor (if any). Because now there is (may be) more than one neighbor of each side, and because MPI *knows nothing* about threads (i.e., thread receiving an MPI message from MPI neighbor does not know from which thread on that node the message is coming) use **unique tags** to label messages sent when processing different tiles
- **mirror** thread trajectories for adjacent MPI subdomains: that is the key to avoid deadlocks
- inner tiling unavoidably fragments MPI messages into smaller ones. This is off-set by combining messages exchanging halos for different arrays into unified messages
- prepare to receive all → send all → check and unpack whatever arrives (no predetermined order)

And, finally, the code is just a tool; **the art is in its usage:** the number of possible permutations is now too large to be quickly explored.

Scheduling messages by inner tiling



N9

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N10

3	2	1	0
4	5	6	7
11	10	9	8
12	13	14	15
19	18	17	16
20	21	22	23
27	26	25	24
28	29	30	31
35	34	33	32
36	37	38	39

N11

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N6

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N7

36	37	38	39
35	34	33	32
28	29	30	31
27	26	25	24
20	21	22	23
19	18	17	16
12	13	14	15
11	10	9	8
4	5	6	7
3	2	1	0

N8

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N3

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N4

3	2	1	0
4	5	6	7
11	10	9	8
12	13	14	15
19	18	17	16
20	21	22	23
27	26	25	24
28	29	30	31
35	34	33	32
36	37	38	39

N5

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N0

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

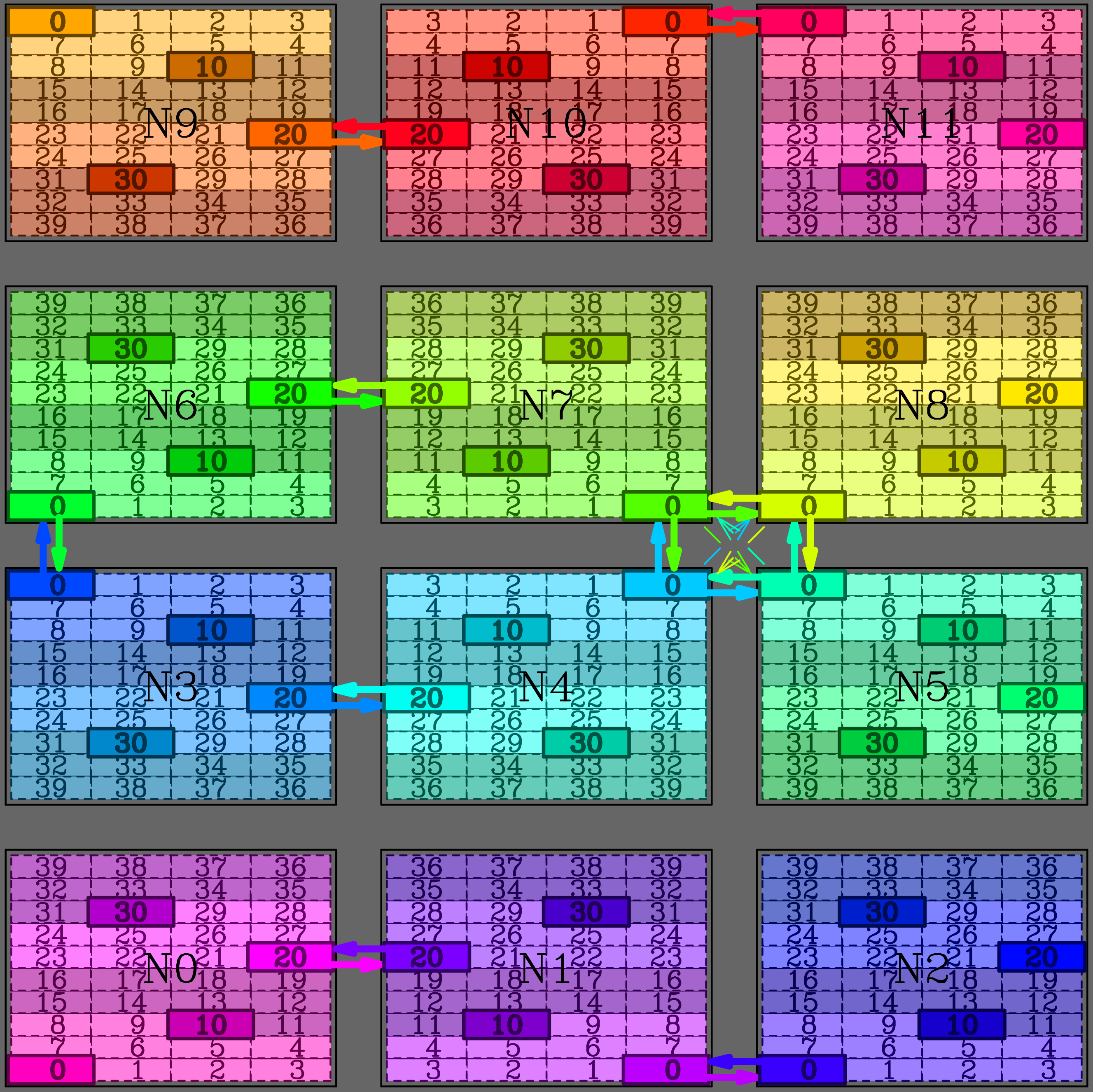
N1

36	37	38	39
35	34	33	32
28	29	30	31
27	26	25	24
20	21	22	23
19	18	17	16
12	13	14	15
11	10	9	8
4	5	6	7
3	2	1	0

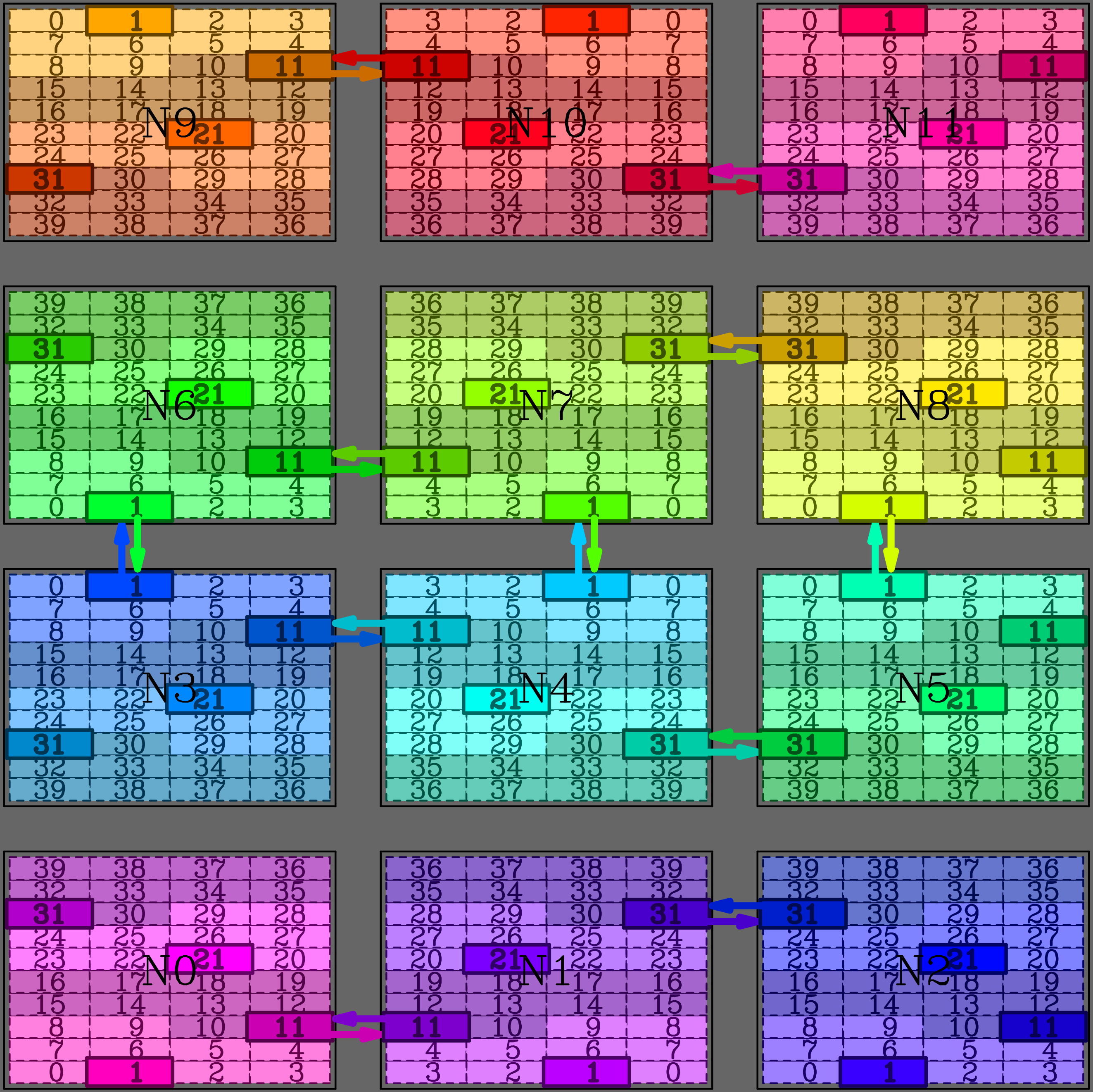
N2

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

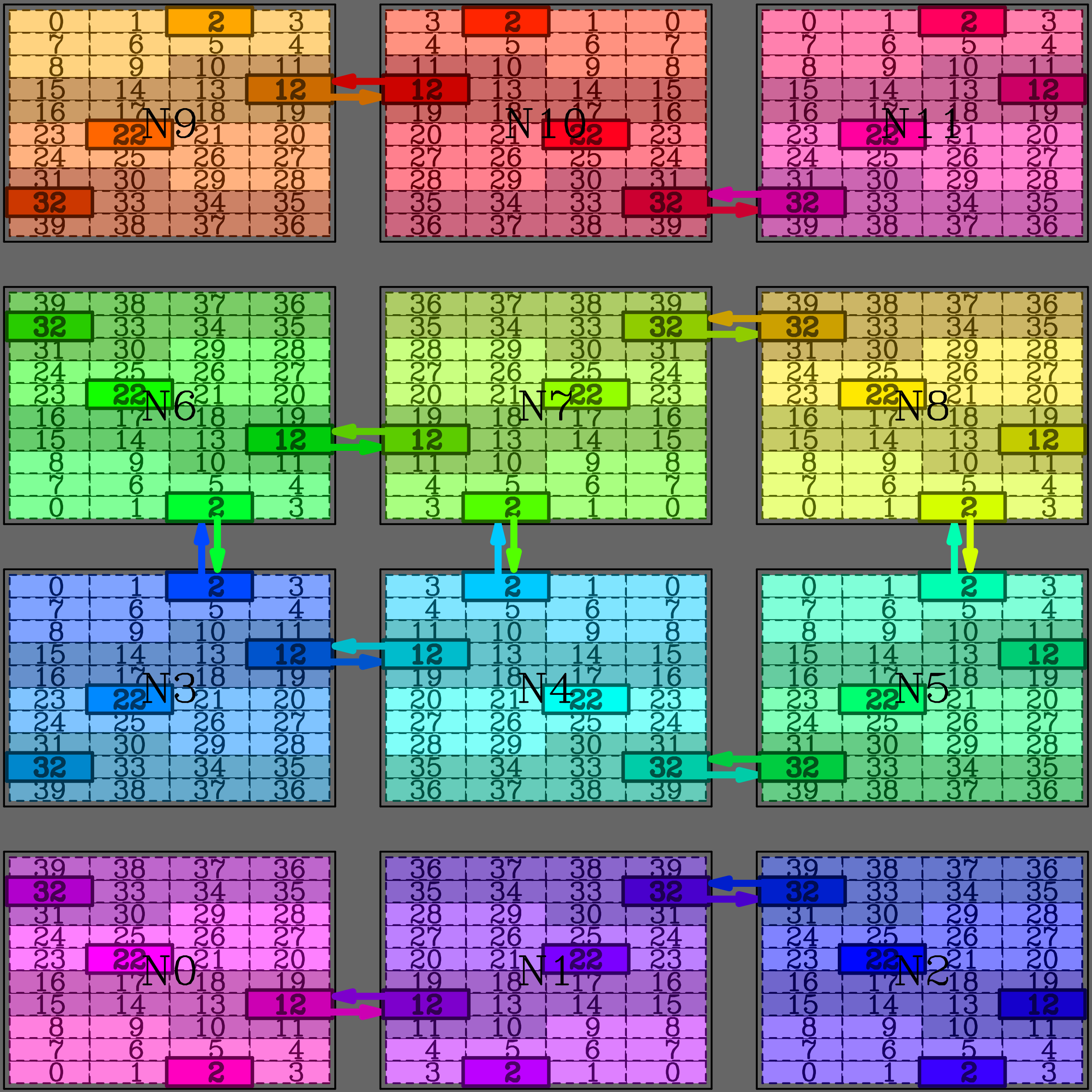
0/10

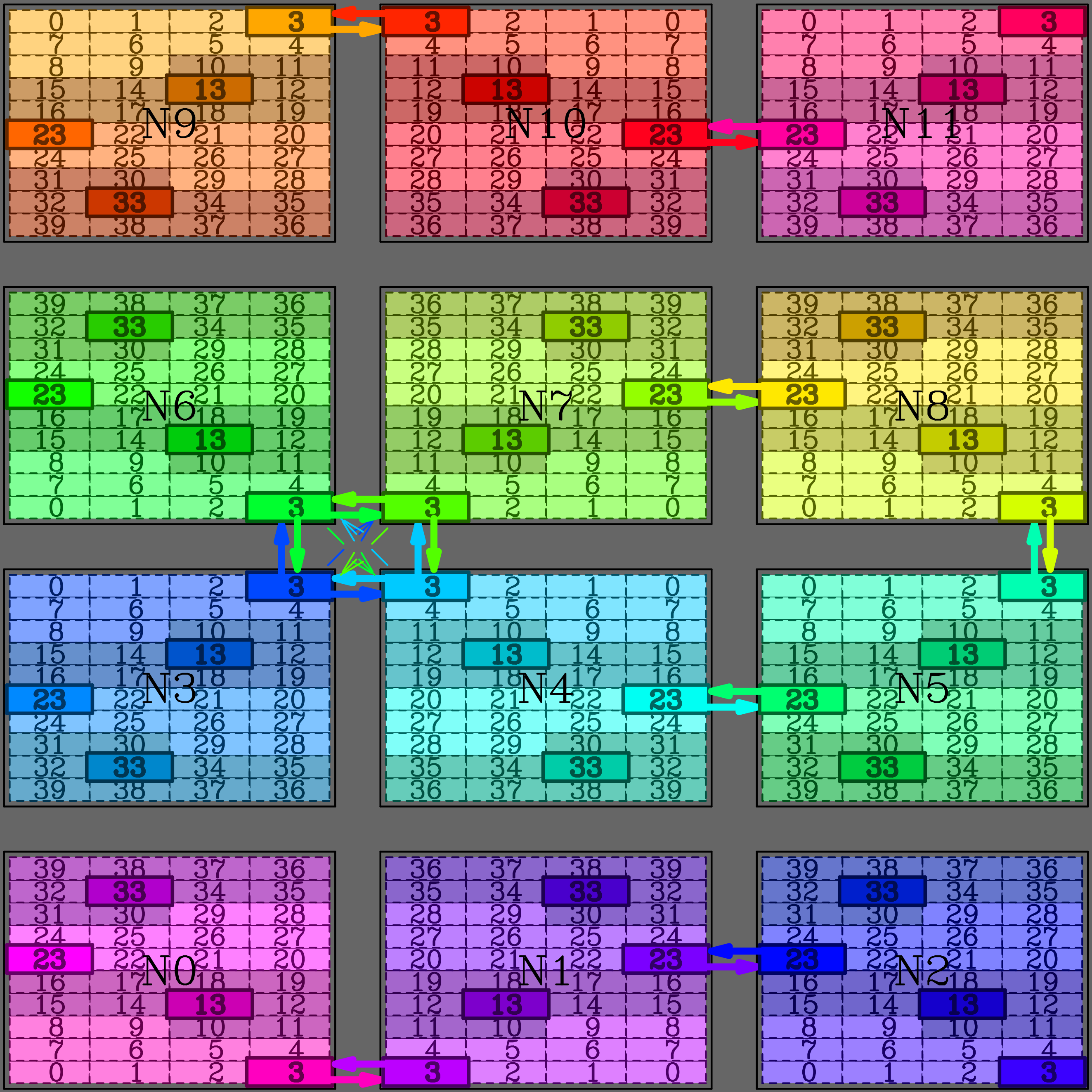


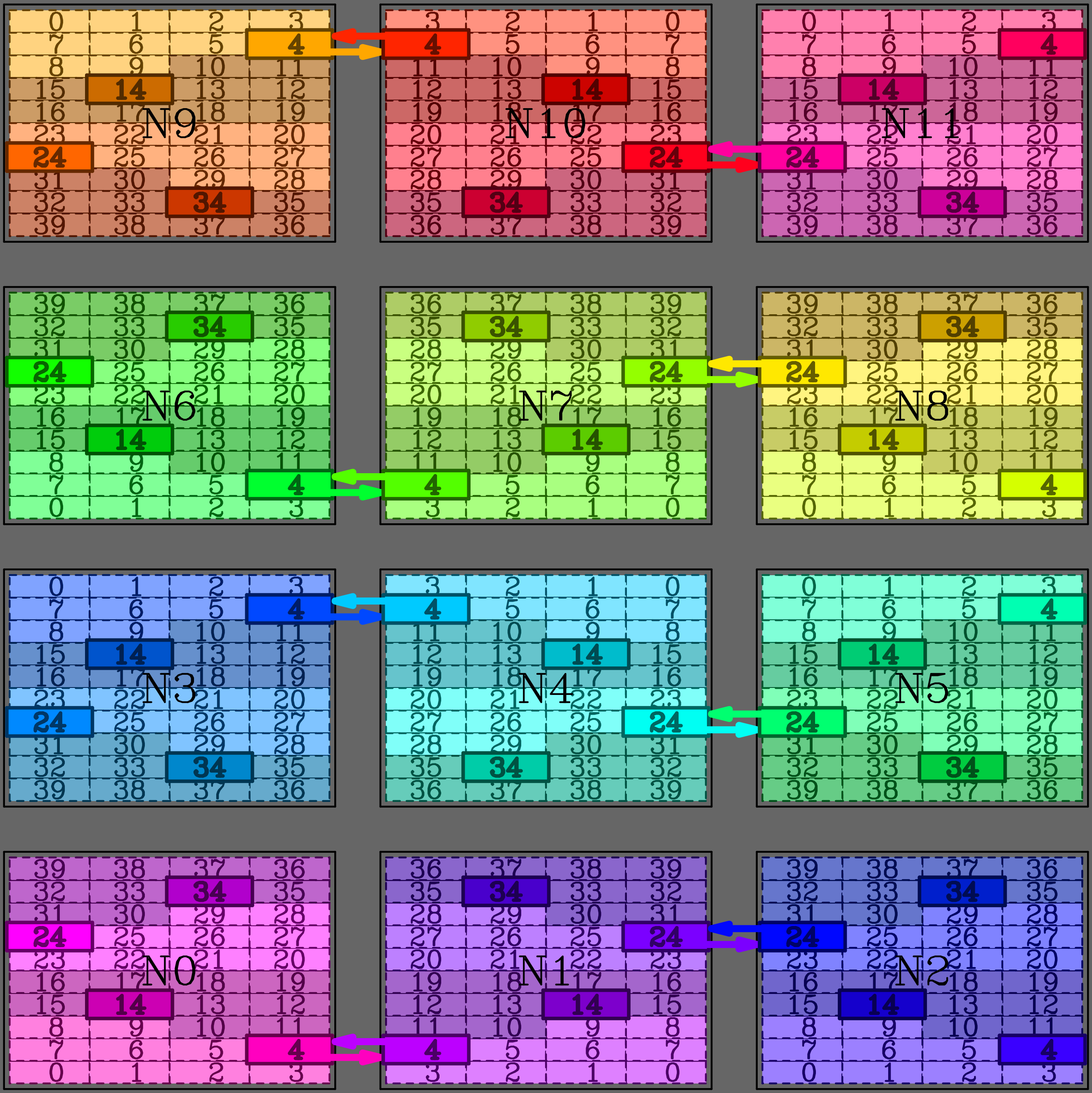
1/10

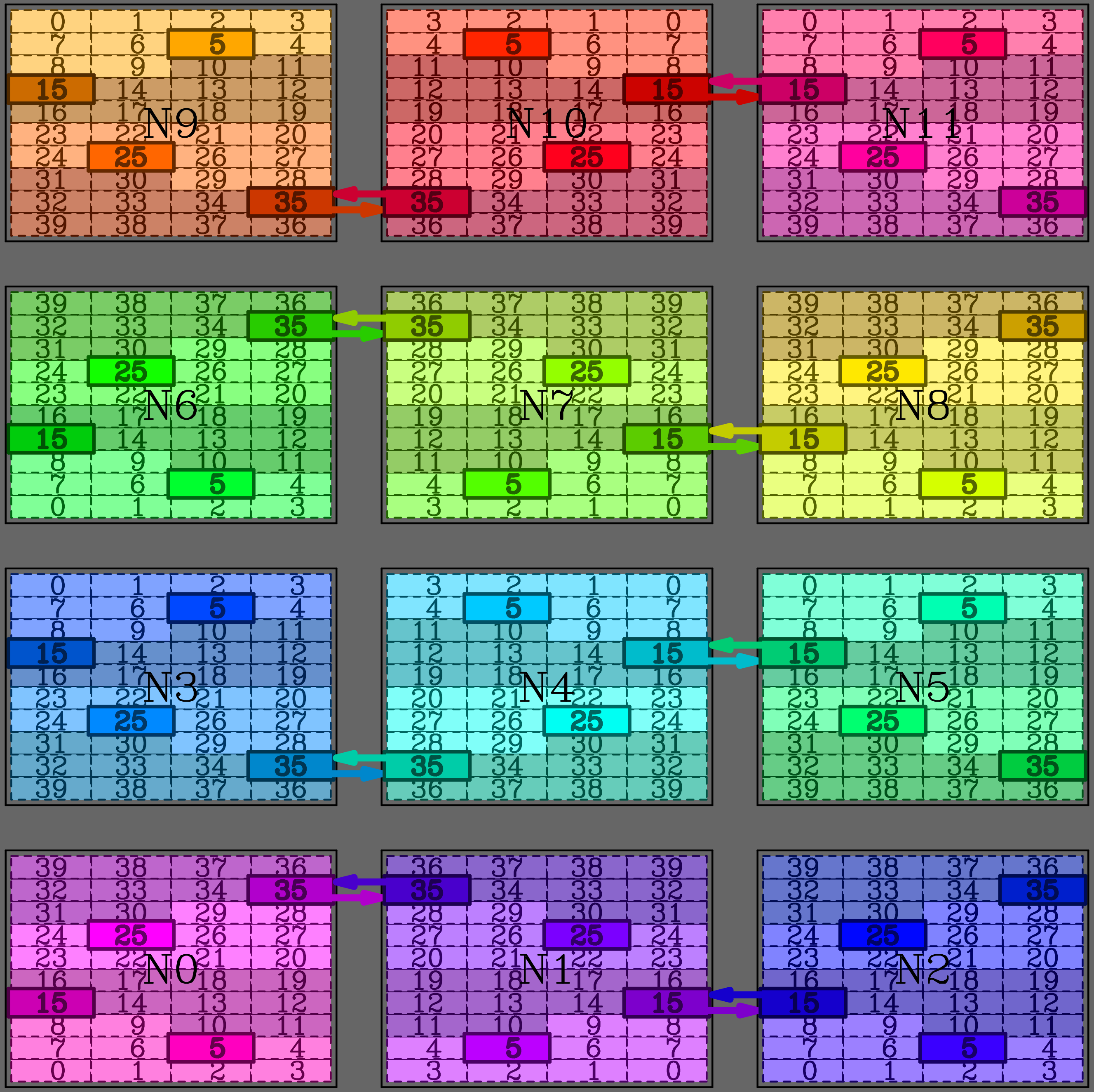


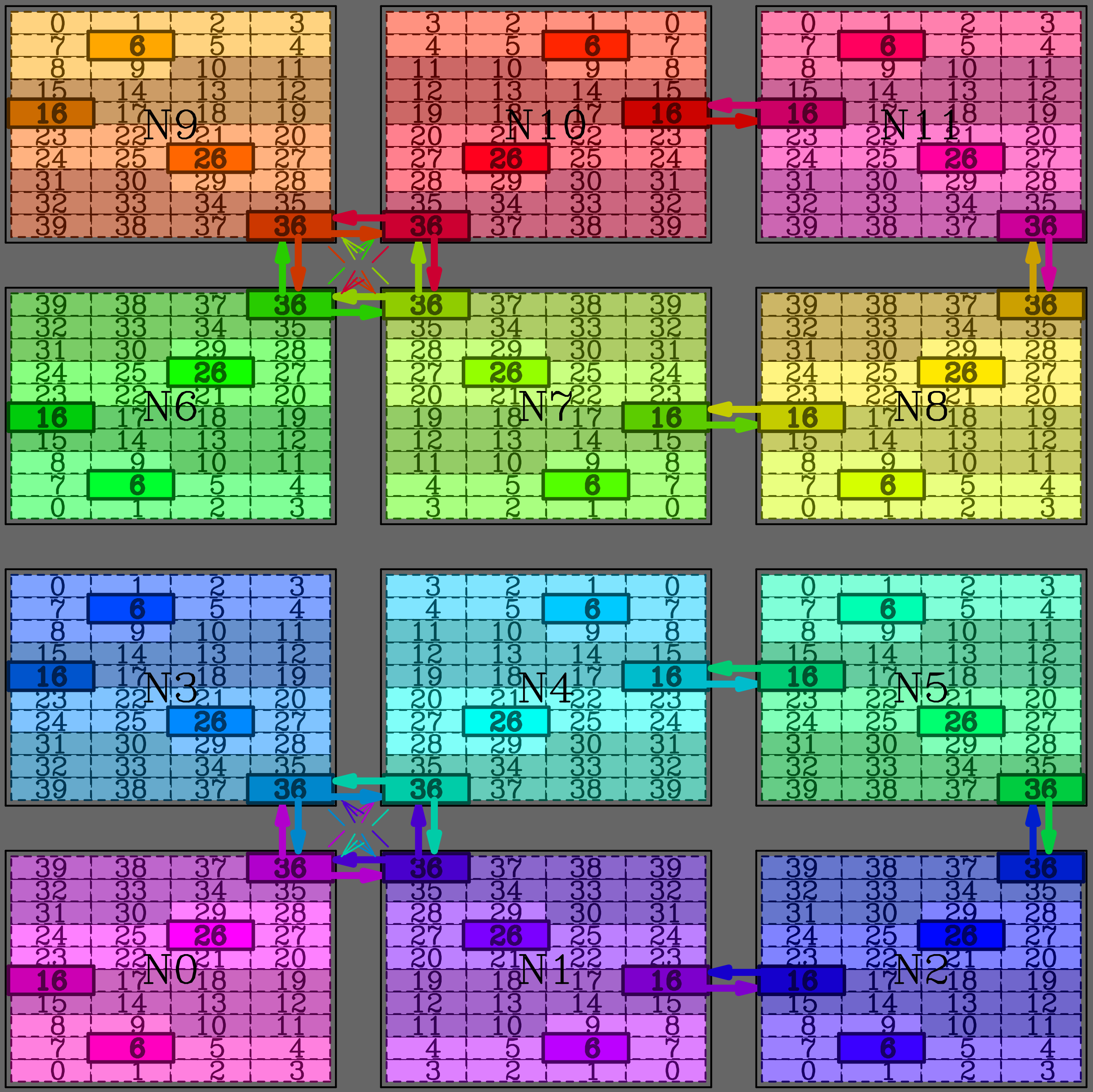
2/10

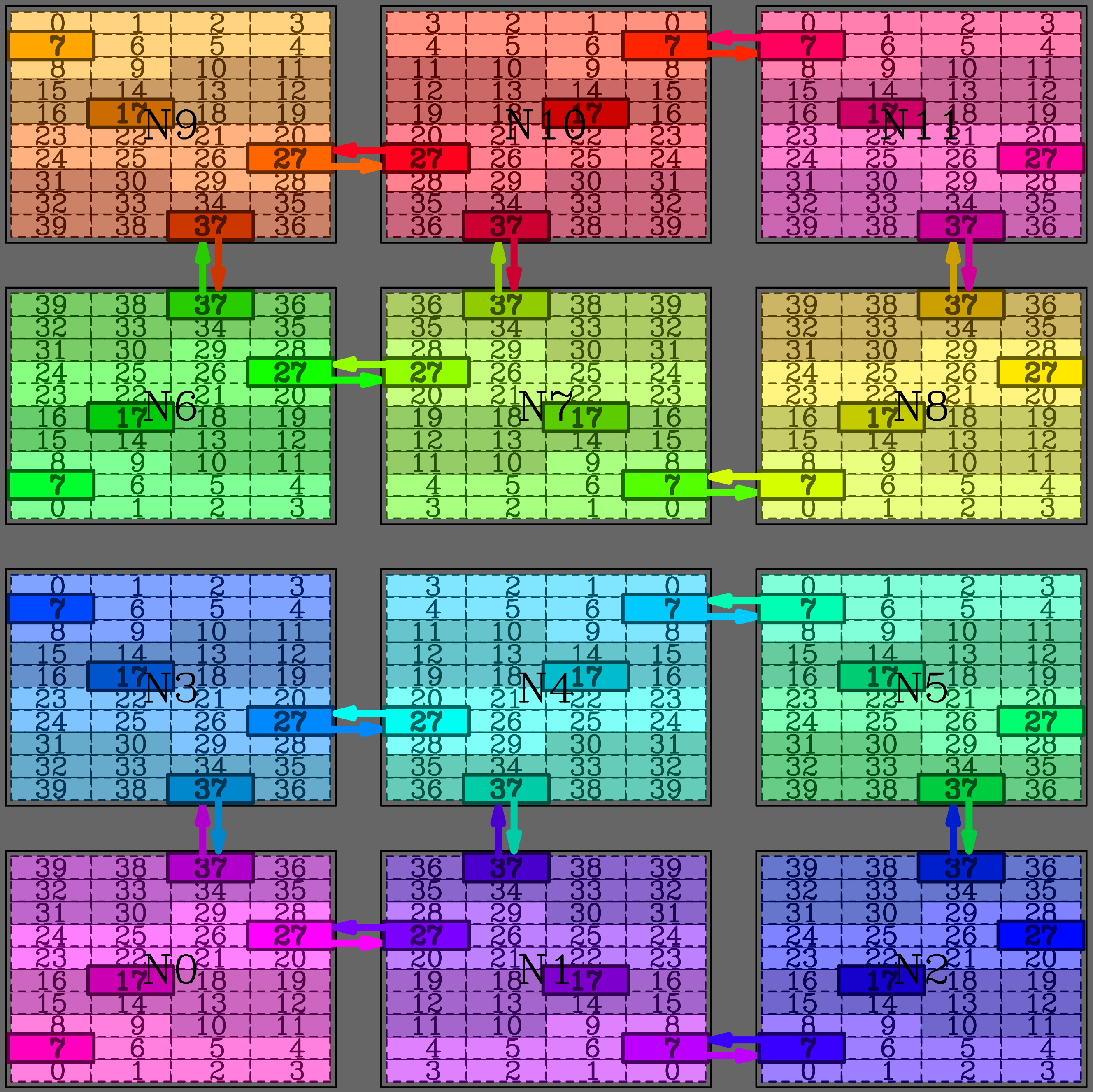

N9
N10
N11
N6
N7
N8
N3
N4
N5
N0
N1
N2

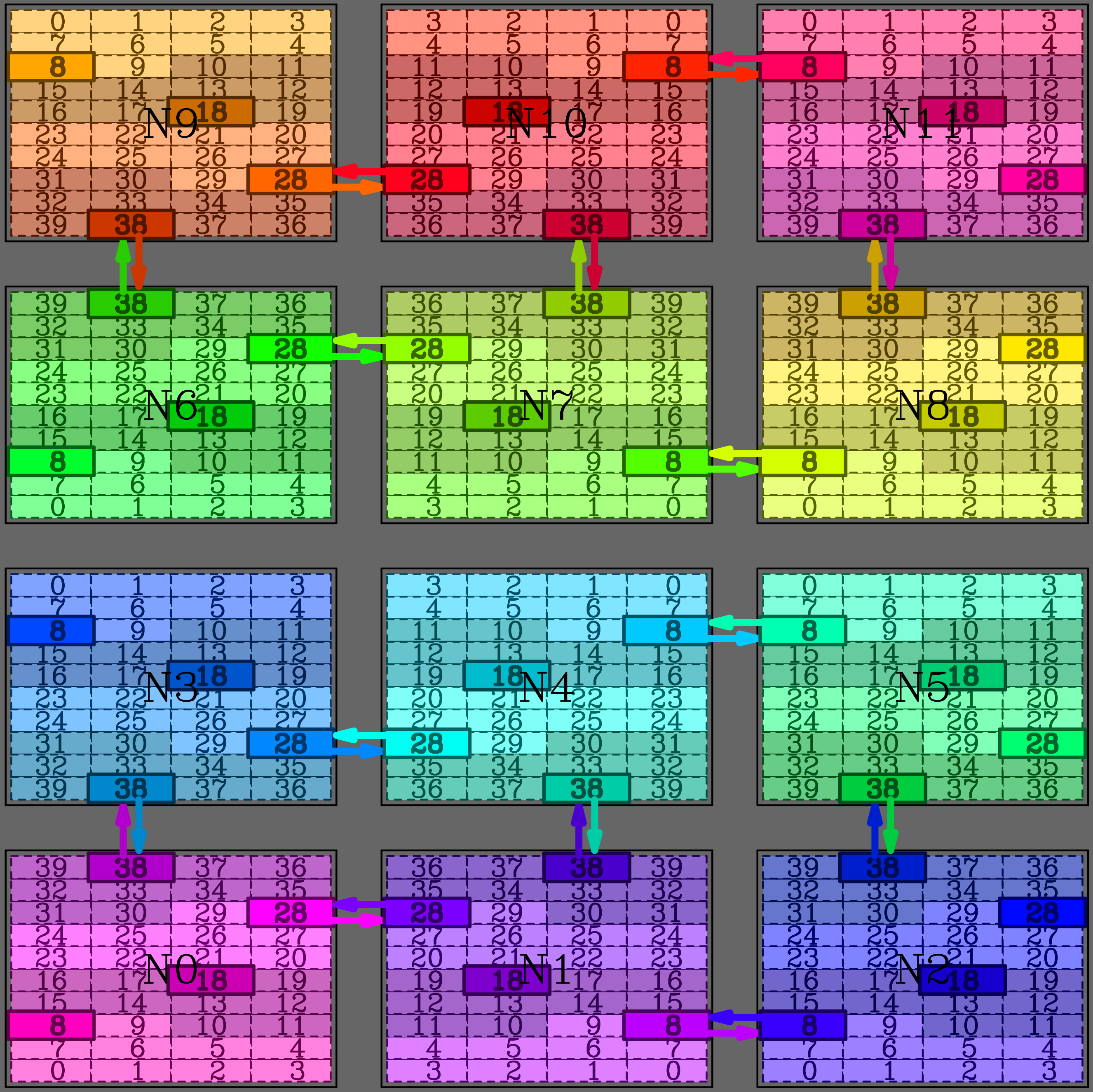

N9
N10
N11
N6
N7
N8
N3
N4
N5
N0
N1
N2


N9
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N10
3 2 1 0
4 5 6 7
11 10 9 8
12 13 14 15
19 18 17 16
20 21 22 23
27 26 25 24
28 29 30 31
35 34 33 32
36 37 38 39
N11
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N6
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3
N7
36 37 38 39
35 34 33 32
28 29 30 31
27 26 25 24
20 21 22 23
19 18 17 16
12 13 14 15
11 10 9 8
4 5 6 7
3 2 1 0
N8
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3
N3
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N4
3 2 1 0
4 5 6 7
11 10 9 8
12 13 14 15
19 18 17 16
20 21 22 23
27 26 25 24
28 29 30 31
35 34 33 32
36 37 38 39
N5
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N0
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3
N1
36 37 38 39
35 34 33 32
28 29 30 31
27 26 25 24
20 21 22 23
19 18 17 16
12 13 14 15
11 10 9 8
4 5 6 7
3 2 1 0
N2
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3


N9
N10
N11
N6
N7
N8
N3
N4
N5
N0
N1
N2


N9
N10
N11
N6
N7
N8
N3
N4
N5
N0
N1
N2


N9
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N10
3 2 1 0
4 5 6 7
11 10 9 8
12 13 14 15
19 18 17 16
20 21 22 23
27 26 25 24
28 29 30 31
35 34 33 32
36 37 38 39
N11
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N6
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3
N7
36 37 38 39
35 34 33 32
28 29 30 31
27 26 25 24
20 21 22 23
19 18 17 16
12 13 14 15
11 10 9 8
4 5 6 7
3 2 1 0
N8
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3
N3
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N4
3 2 1 0
4 5 6 7
11 10 9 8
12 13 14 15
19 18 17 16
20 21 22 23
27 26 25 24
28 29 30 31
35 34 33 32
36 37 38 39
N5
0 1 2 3
7 6 5 4
8 9 10 11
15 14 13 12
16 17 18 19
23 22 21 20
24 25 26 27
31 30 29 28
32 33 34 35
39 38 37 36
N0
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3
N1
36 37 38 39
35 34 33 32
28 29 30 31
27 26 25 24
20 21 22 23
19 18 17 16
12 13 14 15
11 10 9 8
4 5 6 7
3 2 1 0
N2
39 38 37 36
32 33 34 35
31 30 29 28
24 25 26 27
23 22 21 20
16 17 18 19
15 14 13 12
8 9 10 11
7 6 5 4
0 1 2 3



N9

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N10

3	2	1	0
4	5	6	7
11	10	9	8
12	13	14	15
19	18	17	16
20	21	22	23
27	26	25	24
28	29	30	31
35	34	33	32
36	37	38	39

N11

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N6

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N7

36	37	38	39
35	34	33	32
28	29	30	31
27	26	25	24
20	21	22	23
19	18	17	16
12	13	14	15
11	10	9	8
4	5	6	7
3	2	1	0

N8

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N3

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N4

3	2	1	0
4	5	6	7
11	10	9	8
12	13	14	15
19	18	17	16
20	21	22	23
27	26	25	24
28	29	30	31
35	34	33	32
36	37	38	39

N5

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N0

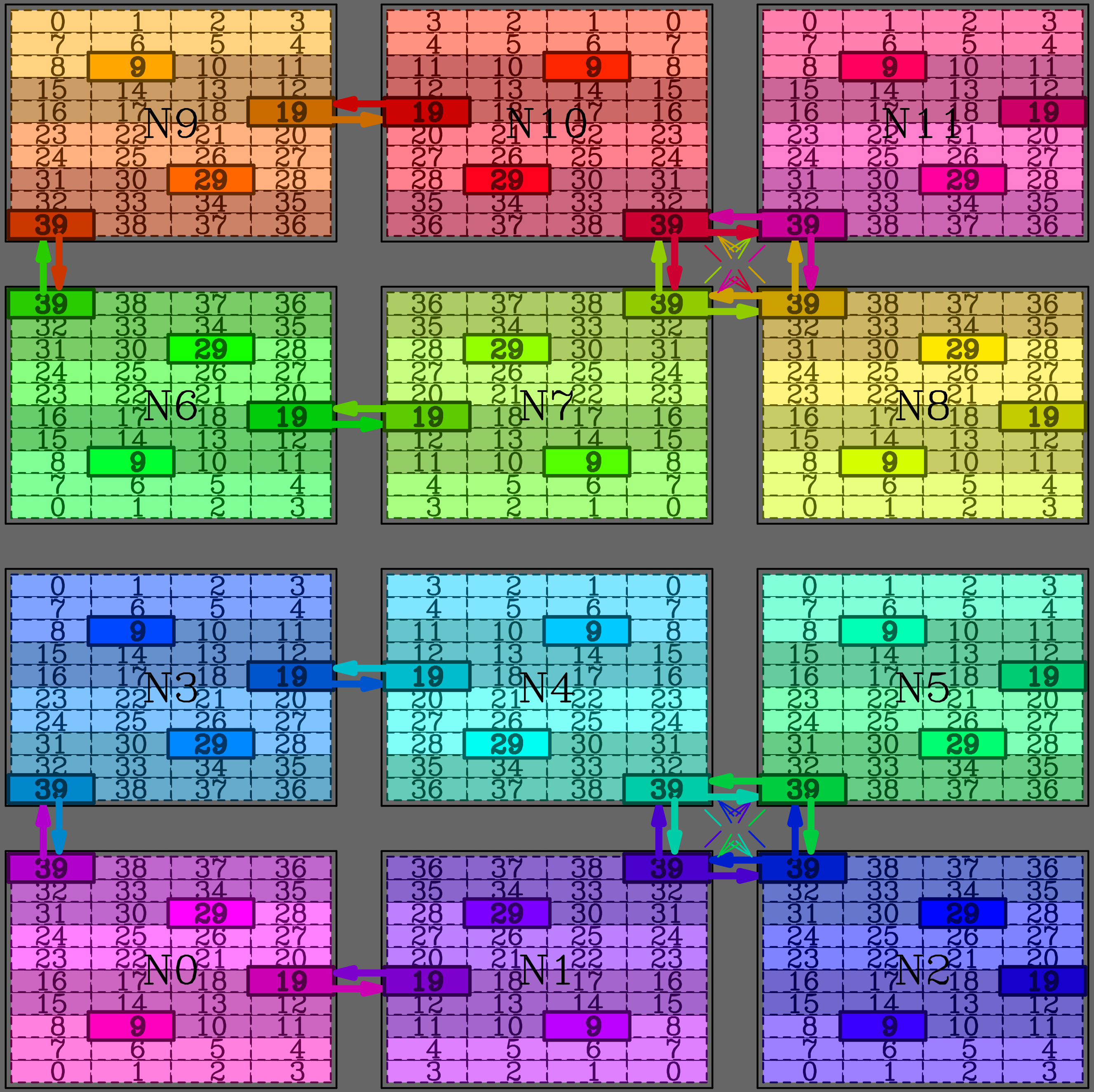
39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N1

36	37	38	39
35	34	33	32
28	29	30	31
27	26	25	24
20	21	22	23
19	18	17	16
12	13	14	15
11	10	9	8
4	5	6	7
3	2	1	0

N2

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3



N9

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N10

3	2	1	0
4	5	6	7
11	10	9	8
12	13	14	15
19	18	17	16
20	21	22	23
27	26	25	24
28	29	30	31
35	34	33	32
36	37	38	39

N11

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N6

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N7

36	37	38	39
35	34	33	32
28	29	30	31
27	26	25	24
20	21	22	23
19	18	17	16
12	13	14	15
11	10	9	8
4	5	6	7
3	2	1	0

N8

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N3

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N4

3	2	1	0
4	5	6	7
11	10	9	8
12	13	14	15
19	18	17	16
20	21	22	23
27	26	25	24
28	29	30	31
35	34	33	32
36	37	38	39

N5

0	1	2	3
7	6	5	4
8	9	10	11
15	14	13	12
16	17	18	19
23	22	21	20
24	25	26	27
31	30	29	28
32	33	34	35
39	38	37	36

N0

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

N1

36	37	38	39
35	34	33	32
28	29	30	31
27	26	25	24
20	21	22	23
19	18	17	16
12	13	14	15
11	10	9	8
4	5	6	7
3	2	1	0

N2

39	38	37	36
32	33	34	35
31	30	29	28
24	25	26	27
23	22	21	20
16	17	18	19
15	14	13	12
8	9	10	11
7	6	5	4
0	1	2	3

Poor Man's Conclusion

- besides cache optimization, inner tiling can be used to control scheduling of communications to alleviate competition between/among CPUs for network access
- works in "soft mode": threads are free-running, and there are no barriers around MPI calls, and no extra barriers relatively to the original OpenMP-only code.
- **thread trajectory does matter** by itself, and relatively to other threads, and it is no longer a simple zig-zag pattern
- partially overlaps computations and communications in time
- softens peak loads on network switch
- finer tiling also means smaller MPI-messages, which negatively affects performance. compromise is needed, and experience tells that message scheduling somewhat is more critical than cache optimization

Does it work?

- Yes, the code is functional and produces the correct result.
- Yes, it may exceed performance of pure MPI code sticking with one-subdomain-per processor(core) policy for problems of our interest.
- Yes, it is useful even if one ends up not using Open MP at all – it is still worth do tiling: **sparse mode**.
- Looking back: never trust you intuition about ideas for improving performance of MPI code: always try all the possibilities and select what works the best.
- Swappable MPI halo-exchange routines.