An Adaptive, Courant-number-dependent Implicit Scheme for Vertical Advection in Oceanic Modeling

Alexander F. Shchepetkin

Institute of Geophysics and Planetary Physics, University of California, Los Angeles,
405 Hilgard Avenue, Los Angeles, CA 90095-1567

e-mail: `old_galaxy@yahoo.com`

Abstract

An oceanic model with an Eulerian vertical coordinate and an explicit vertical advection scheme is subject to the Courant–Friedrichs–Lewy (CFL) limitation. Depending on the horizontal grid spacing, the horizontal-to-vertical grid resolution ratio and the flow pattern this limitation may easily become the most restrictive factor in choosing model time step, with the general tendency to become more severe as horizontal resolution becomes finer. Using terrain-following coordinate makes local vertical grid spacing depend on topography, ultimately resulting in very fine resolution in shallow areas in comparison with other models, z-coordinate, and isopycnic, which adds another factor in restricting time step. At the same time, terrain-following models are models of choice for the fine-resolution coastal modeling, often including tides interacting with topography resulting in large amplitude baroclinic vertical motions. In this article we examine the possibility of mitigating vertical CFL restriction, while at the same time avoiding numerical inaccuracies associated with standard implicit advection schemes. In doing so we design a combined algorithm which acts like a high-order explicit scheme when Courant numbers are small enough to allow explicit method (which is usually the case throughout the entire modeling domain except just few "hot spots"), while at the same time has the ability to adjust itself toward implicit scheme should it became necessary to avoid stability limitations. This is done in a seamless manner by continuously adjusting weighting between explicit and implicit components.

---

1   Introduction

Selecting the size of time step Δt in oceanic modeling requires to satisfy multiple criteria associated with different physical processes in order to guarantee numerical stability (*cf.,* Griffies and Adcroft,

2008, esp. Sec. 8.4 and Table 1 there). For large-scale, coarse-resolution $\Delta x \sim \mathcal{O}(50...100km)$ modeling inertial CFL, $f\Delta t$, is usually found to be the most restrictive limiting the time step not to exceed $7200...8600sec$ (*cf.,* Table 1 in Shchepetkin and McWilliams, 2005, hereafter cited as SM2005). A finer $\Delta x \sim \mathcal{O}(10...50km)$ resolution requires using smaller Δt due to CFL based on maximum propagation speed of internal waves, $c_1\Delta t/\Delta x$, where c_1 is the phase speed of the first baroclinic mode, and CFL based on horizontal advection speed, $|\mathbf{u}|\Delta t/\Delta x$. These two criteria act in a mutually additive manner (rather than independently), meaning that the phase speed should be added current velocity, intuitively [1] as $(|\mathbf{u}| + c_1)\Delta t/\Delta x$, and in they lead to a proportional decrease of time step with horizontal grid spacing, $\Delta t \sim \Delta x$, while in practice a slightly faster decrease is typically required due to the increase of maximum advection speed as the simulated flow becomes more energetic with grid refinement. Vertical mixing processes are treated using an implicit algorithm and impose no additional restriction on time step. Neither does the barotropic mode, which is solved separately by an implicit or split-explicit method. Because at such resolutions the depth-to-horizontal grid size ratio is small, $h/\Delta x < 1$, the finiteness of grid resolution removes the possibility of non-hydrostatic motions. In a hydrostatic model vertical velocity exists only as finite-volume flux integrated over grid-box area as seen from above, hence is effectively averaged tens of km^2, this naturally leading to very small numbers.

This situation changes with further refinement of horizontal grid. In fact, practical experience with simulations with horizontal resolution less than a few km reveals that limitation due to CFL associated with vertical advection $w\Delta t/\Delta z$ gradually becomes the most restrictive one, depending on the topographic features and/or whether or not the tides are part of the simulation. For sub-$1km$ grids it becomes so dominant that the imposed time step limitation is several times smaller than due to horizontal advection and/or internal wave phase speeds. This takes place even when the relevant vertical-to-horizontal aspect ratios and vertical accelerations (diagnosed from the actual solutions) are sufficiently small to cast doubt about validity of hydrostatic approximation. In part it is because $\Delta z/\Delta x \ll 1$, so while vertical velocity is small, in comparison with horizontal vertical CFL may be not be so.

Detailed investigation of the associated "hot spots" (characteristic locations on the model grid where numerical instability of the explicit code occurs first) reveals that large vertical CFL always occur near topographic features, where buoyancy stratification is weak or vanishing, but not necessarily in the shallowest areas where vertical grid spacing is the smallest due to topography-following coordinate. The causes for more restrictive vertical CFL at higher horizontal resolutions can be identified as follows

(*i*) the prevailing pattern in vertical velocity field are narrow upwelling or downwelling contours along the propagating fronts in temperature, salinity, or vorticity fields. Refinement of horizontal resolution causes sharpening of these fronts, however the integral uplifting or downwelling of water must balance the horizontal divergence, resulting in scaling of amplitude of vertical velocity as $w \sim 1/\Delta x$;

(*ii*) with finer resolution bottom topography is subject to lesser smoothing resulting in larger absolute slopes, which translate into larger vertical velocities, as well as capturing phenomena such as

---

[1] It should be noted that in anticipation that the baroclinic wave phase speed is typically larger than the maximum advection speed, modern ocean modeling codes may take the advantage of using a variation of forward-backward stepping for tracer and momentum equation resulting in a larger theoretical stability limit for waves than for advection, so the summation in $|\mathbf{u}|+c_1$ should be replaced with a weighted sum to take this into account.

74 topographic refraction and focusing on internal waves;
75 (*iii*) vertical mixing parameterization schemes have tendency to set off unbalanced states by rapidly
76 mixing negative stratification throughout some vertical columns;
77 (*iv*) bringing in new physical processes: high-resolution modeling tends to be accompanied with tidal
78 forcing, which, in combination with bottom topography and stratification results large-amplitude
79 baroclinic motions.

80 While implicit advection schemes offer a relief from CFL limitation, their drawbacks are well
81 known: unavoidable and potentially large dispersive errors increasing with CFL, and depending on
82 the detail of time and space discretization, large numerical viscosity as well (*cf.,* Shchepetkin and
83 McWilliams (2009)). In contrast, the explicit vertical advection schemes of ROMS are designed to be
84 high order in space (4th-centered or compact based parabolic spline fits), which makes it not feasible
85 to design an implicit version of comparable accuracy.

86 To overcome the dilemma in this article we pursue an adaptive approach where the advection
87 scheme remains explicit (as in the original code) everywhere except where/when local vertical ve-
88 locities exceed a threshold close to (but below) the explicit stability limit. Once this happens, a gradual
89 transition toward an implicit scheme begins via Courant-number-dependent weighting algorithm. As
90 we are not aware of any prototype of such approach published in the literature, we present it in full
91 detail.

2 Explicit and implicit advection at finite-Courant-numbers

93 Vertical advection in oceanic modeling poses specific requirements to avoid long-term accumula-
94 tion of numerical errors in tracer fields due to oscillatory vertical motions typical for ocean dynamics,
95 which in its turn makes the choice of advection schemes for vertical direction distinct in its priorities
96 from the algorithms used elsewhere. Thus, upstream biasing is generally avoided for the dynamically
97 active tracers, temperature and salinity: preserving monotonicity becomes a lesser priority over "re-
98 versibility" of dispersive errors – ability not to erode thermocline in repeated up-and-down motions.
99 This consideration may be reversed for the other tracers (turbulent kinetic energy, biological, etc..),
100 where maintaining positivity becomes a higher priority, and monotonicity limiters require that the
101 overall advection algorithm is two-time-level, logically forward-in-time, which unavoidably makes it
102 time-space-dependent (hence upstream-biased) for the numerical stability. Another constraint is that
103 the overall time stepping algorithm (including momentum equations and advection of active tracers)
104 must be stable with respect to baroclinic internal waves, Rueda et al. (2007), who showed that while
105 forward-in-time advection can be made stable by upstream-biasing if advection is considered *alone*,
106 upstream-bias by itself does not help to stabilize the mechanism of mutual feedback between momen-
107 tum and tracer equations which is associated with propagation of internal waves. In any case, leaving
108 dispersive errors unchecked is also not acceptable: in a realistic oceanic code dispersive overshoots are
109 detected as negative stratification by vertical mixing parameterization algorithm triggering enhanced
110 mixing, resulting to overall stable, but non-physical solution, *e.g.,* Hecht (2010). Similarly, overshoots
111 caused by an explicit scheme in the regime marginally beyond its limit of stability due to vertical
112 over-speeding may be viewed as negative stratification, hence triggering enhanced mixing by vertical
113 mixing parameterization scheme and for this reason may not lead to computational instability right a
114 way, however, once again, resulting in artifacts.

In this section we overview properties of known advection schemes focusing on their behavior over the entire range of Courant numbers within the limit of stability.

2.1 Explicit advection

Consider for simplicity one-dimensional advection

$$\partial_t q + c \cdot \partial_x q = 0 \tag{2.1}$$

with uniform velocity $c = const$ discretized on a uniformly-spaced grid $\left\{x_j \big| j = 1, .., N\right\}$, $\Delta x = x_{j+1} - x_j = const$. A flux-form algorithm updates q_j as

$$q_j^{n+1} = q_j^n - \Delta t \left[F_{j+1/2} - F_{j-1/2}\right] \Big/ \Delta x \,. \tag{2.2}$$

Computation of fluxes $F_{j+1/2}$ involves interpolation of field q in space as well as proper time placement in order ensure temporal stability and accuracy of the algorithm – note that thus far the time index of $F_{j+1/2}$ in (2.2) above is undefined – it should be "effectively" centered at $n + 1/2$ to ensure numerical stability and at least second-order accuracy.

For the spatial interpolation we assume that within each cell $x' \in [x_j - \Delta x/2, x_j + \Delta x/2]$ the distribution of $q = q(x')$ is approximated by a parabolic segment such that its averaged value within the cell j is equal to the given value q_j while the left- and right-side limits $\tilde{q}_j^L$ and $\tilde{q}_j^R$ are computed from the set of $\{q_j\}$ by an appropriate reconstruction algorithm,

$$q(x') = q_j + \frac{\tilde{q}_j^R - \tilde{q}_j^L}{\Delta x} x' + 3\frac{\tilde{q}_j^R - 2q_j + \tilde{q}_j^L}{\Delta x^2} \left(x'^2 - \frac{\Delta x^2}{12}\right) , \tag{2.3}$$

where it can be verified that $q\big|_{x' \to +\Delta x/2} \to \tilde{q}_j^R$, $q\big|_{x' \to -\Delta x/2} \to \tilde{q}_j^L$, and $\frac{1}{\Delta x} \int_{-\Delta x/2}^{+\Delta x/2} q(x') \, dx' = q_j$. The reconstructed profile may yield either continuous or discontinuous at the grid-box interfaces, either $\tilde{q}_j^R = \tilde{q}_{j+1}^L$ or $\tilde{q}_j^R \neq \tilde{q}_{j+1}^L$, depending on the specifics of the reconstruction algorithm, the degree of smoothness of field q on the grid scale, and whether or not enforcement of monotitonicity is desired. In the continuous case it is convenient to introduce the "shared" values $\tilde{q}_{j+1/2}$ at each interface,

$$\tilde{q}_j^R = \tilde{q}_{j+1/2} = \tilde{q}_{j+1}^L \,. \tag{2.4}$$

One of the options for vertical advection in ROMS is to compute vertical interface values by parabolic spline reconstruction. Initially motivated by its ability to work on highly stretched vertical grids without loss of accuracy, it is also known for a much smaller numerical dispersion relatively to other schemes of formally the same order of accuracy. Some other, more traditional algorithms are described Appendix A. On a uniform grid parabolic spline reconstruction leads to

$$\frac{1}{6}\tilde{q}_{j-1/2} + \frac{2}{3}\tilde{q}_{j+1/2} + \frac{1}{6}\tilde{q}_{j+3/2} = \frac{q_j + q_{j+1}}{2} \tag{2.5}$$

which needs to be solved for all half-integer-indexed $\tilde{q}_{j+1/2}$ simultaneously.

As for the time placement of $\tilde{q}_{j+1/2}$ in $F_{j+1/2}$, there are fundamentally two different approaches: either to use a suitable time stepping algorithm independent of spatial discretization (so called method

4

of lines, Hyman, 1979), or, conversely, use a variant of tracking the advected field back in time along the characteristics in time-space, leading to a semi-Lagrangian approach in conservation form (van Leer, 1979; Colella and Woodward, 1984; Leonard, 1991). An example of the former is LF-AM3 predictor corrector stepping,

$$q_j^{n+1/2} = \left(\frac{1}{2} - 2\gamma\right) q_j^{n-1} + \left(\frac{1}{2} + 2\gamma\right) q_j^n - \Delta t \cdot c \left(1 - 2\gamma\right) \left[\tilde{q}_{j+1/2}^n - \tilde{q}_{j-1/2}^n\right] \Big/ \Delta x \qquad (2.6)$$

followed by

$$q_j^{n+1} = q_j^n - \Delta t \cdot c \left[\tilde{q}_{j+1/2}^{n+1/2} - \tilde{q}_{j-1/2}^{n+1/2}\right] \Big/ \Delta x \,, \qquad (2.7)$$

where during both stages $F_{j+1/2} \equiv c\tilde{q}_{j+1/2}$ for the respective time indices; $\gamma = 1/12$ leads to the third-order temporal accuracy.

In the case of a semi-Lagrangian algorithm based on parabolic reconstruction,

$$F_{j+1/2} = c \left\{ \tilde{q}_j^R\Big|^n - \frac{\alpha}{2} \left[\tilde{q}_j^R\Big|^n - \tilde{q}_j^L\Big|^n + (3 - 2\alpha) \left(\tilde{q}_j^R\Big|^n + \tilde{q}_j^L\Big|^n - 2q_j^n \right) \right] \right\} \,, \qquad (2.8)$$

where $\alpha = |c|\Delta t/\Delta x$ is Courant number. The above is derived by integrating the parabolic segment (2.3) over the interval $[x_{j+1/2} - \alpha\Delta x, x_{j+1/2}]$ assuming that the flow direction is from left to right, hence c is positive [2]. In the case of negative c the integration interval becomes $[x_{j+1/2}, x_{j+1/2} + |\alpha| \cdot \Delta x]$, so the resultant $F_{j+1/2}$ involves q_{j+1}^n, $\tilde{q}_{j+1}^L\Big|^n$, $\tilde{q}_{j+1}^R\Big|^n$ instead of q_j^n, $\tilde{q}_j^R\Big|^n$ $\tilde{q}_j^L\Big|^n$ respectively. Note that left- and right-side limits switch their roles; α inside brackets in (2.8) depends on absolute value advecting velocity, so it is always positive. In the case of continuous reconstruction the side limits $\tilde{q}^R$ and $\tilde{q}^L$ are replaced with interface values $\tilde{q}_{j+1/2}$ with appropriate spatial indices according to (2.4).

To investigate accuracy of the above algorithms we consider a Fourier component,

$$q_j^n = \lambda^n \cdot \widehat{q}_k \cdot e^{ik\Delta x j} \qquad (2.9)$$

where $\widehat{q}_k$ is a Fourier amplitude of wavenumber k and λ is step multiplier, which ideally for the "exact" solution should be $\lambda = e^{-ick\Delta t} = e^{-i\alpha k\Delta x}$. It is worth to note that in Taylor expansion of

$$\lambda^{(\text{exact})} = 1 - i\alpha k\Delta x - \frac{\alpha^2 (k\Delta x)^2}{2} + \frac{i\alpha^3 (k\Delta x)^3}{6} + \frac{\alpha^4 (k\Delta x)^4}{24} - \dots \qquad (2.10)$$

all α and $k\Delta x$ appear together in equal powers.

By substituting (2.9) into (2.5) we find

$$\tilde{q}_{j+1/2} = \widehat{q}_k \cdot \frac{\cos\left(k\Delta x/2\right)}{(2/3) + (1/3)\cos(k\Delta x)} \cdot e^{ik\Delta x(j+1/2)} \qquad (2.11)$$

---

[2] This derivation appears in multiple sources, most notably Colella and Woodward (1984, see Sec. 1, esp. Eqs. (1.11)-(1.13)), but can be traced back to earlier work of van Leer (1979, Appendix B, esp. Eq. (B5)); Leonard (1979, Sec. 4, QUICKEST method). In the case of 1D advection in by uniform velocity c the overall scheme can be shown to be equivalent to interpolation of the field itself, Leonard (1988). In fact, under such condition substitution of (2.8) into (2.2) and taking into account (2.5) can be shown to be in equivalent to finding q_j^{n+1} as the result of cubic spline interpolation of field q^n to the location $x = x_j - \alpha\Delta x$.

therefore

$$\tilde{q}_{j+1/2} - \tilde{q}_{j-1/2} = \widehat{q}_k \cdot \frac{i\sin(k\Delta x)}{(2/3) + (1/3)\cos(k\Delta x)} \cdot e^{ik\Delta xj} = i\mathscr{K}\Delta x \cdot \widehat{q}_k \cdot e^{ik\Delta xj} \qquad (2.12)$$

where $\mathscr{K} = \mathscr{K}(k\Delta x)$ is the Fourier image of the finite-difference operator in (2.2) used in combination with (2.5). Its Taylor expansion

$$i\mathscr{K}\Delta x = ik\Delta x\left(1 - \frac{1}{180}(k\Delta x)^4 + ...\right) \qquad (2.13)$$

reveals that it is fourth-order accurate. Furthermore, its leading-order $(k\Delta x)^4$ truncation term is six times smaller than that of the conventional non-staggered fourth-order finite difference. Overall it has
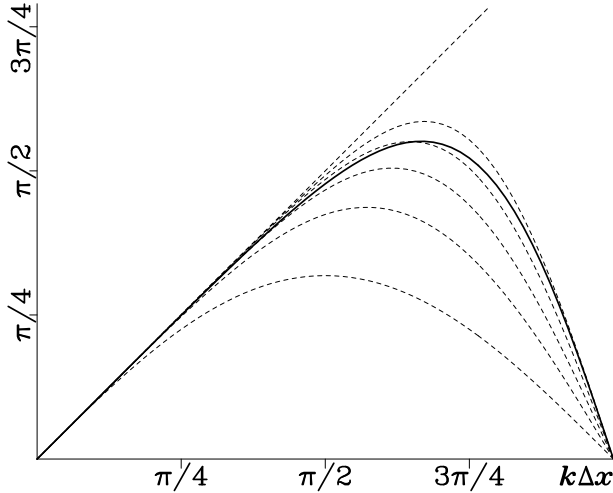

Fig. 1. Comparison of Fourier image of compact difference operator (2.2)-(2.5) with similar images of conventional finite differences. Thin dashed straight diagonal line corresponds to the "ideal" $\mathscr{K}(k\Delta x) = k$. The five thin dashed lines are for the conventional non-staggered finite-difference schemes, starting with the second-order (the lowest curve), then fourth-, sixth-, eighth-, and tenth-order (the highest). Superimposed bold solid curve is for $\mathscr{K} = \mathscr{K}(k\Delta x)$ via Eq. (2.12). While formally fourth-order accurate, it is most similar to eighth-order difference in effective resolution. It reaches the maximum value of $\mathscr{K}\Delta x = \sqrt{3}$ at $k\Delta x = 2\pi/3$.

a significantly wider range of wavenumbers for which the derivative is accurately computed[3], Fig. 1.

Inserting (2.12) into (2.6)-(2.7) and combining the two equations into one, we find the characteristic equation of LF-AM3 advection algorithm,

$$\lambda^2 - \left[1 - \alpha^2(\mathscr{K}\Delta x)^2(1 - 2\gamma) - i\alpha\mathscr{K}\Delta x\left(\frac{1}{2} + 2\gamma\right)\right]\lambda + i\alpha\mathscr{K}\Delta x\left(\frac{1}{2} - 2\gamma\right) = 0, \qquad (2.14)$$

---

[3] The fact that replacing polynomial interpolation to compute mid-point values by an implicit relation (2.5) results in a much more accurate approximation for the first derivative at co-located points (hence on a non-staggered grid) was noticed by Kreiss, who pioneered *compact fourth-order* differencing (private communication acknowledged in Orszag and Israeli, 1974 and also Hirsh, 1975). Eq. (2.5) combined with differencing (2.2) can be algebraically transformed into a tri-diagonal system for the first derivatives as co-located points, $d_j = \partial_x q\big|_{x=x_j}$, which is also known as classical *Padé scheme*. This terminology comes from the analogy with rational function approximation when symbolically writing the compact fourth-order finite-difference operator in form $\partial/\partial x \approx D_c/\left[1 + (1/6)\Delta x^2 D_+ D_-\right]$ where $D_c f = (f_{j+1} - f_{j-1})/(2\Delta x)$ is centered-, while $D_- f = (f_j - f_{j-1})/\Delta x$ and $D_+ f = (f_{j+1} - f_j)/\Delta x$ are left- and right-sided divided differences, *cf.,* Eq. (12) from Orszag and Israeli (1974). In the same notation the conventional fourth-order differencing is written as $\partial/\partial x \approx D_c\left[1 - (1/6)\Delta x^2 D_+ D_-\right]$ as it appears in Eq. (11) from the same source. "Division" by finite difference operator implies inversion of an implicit system. Naturally, its Fourier analysis leads to rational functions like (2.12), instead of than polynomials in powers of $\sin(k\Delta x)$ and $\cos(k\Delta x)$. Compact differencing is mathematically related to spline interpolation, De Boor (1978), and can be extended to arbitrarily high order of accuracy, Lele (1992).

which leads to $\lambda = \lambda(\alpha \mathscr{K} \Delta x)$ as a single variable function (the larger by amplitude root corresponds to the physical mode, the smaller to the computational), so Taylor expansion of λ keeps terms with equal powers of α and $\mathscr{K} \Delta x$ together, and because $i\mathscr{K}\Delta x$ matches $ik\Delta x$ up to $(k\Delta x)^4$-term, Taylor expansion of (2.14) in terms of powers of α and $k\Delta x$ also keeps their powers together, similarly to (2.10), for up to (including) the third power. This means that the overall time-space order of accuracy of this algorithm is the third if $\gamma = 1/12$ simply because the order of accuracy of LF-AM3 stepping is that. Fig. 2 investigates properties of (2.14) in further detail.

Besides showing real and imaginary parts of λ, we introduce two other metrics: dissipation per $1\Delta x$ travel

$$|\lambda|^{(1/\alpha)} \tag{2.15}$$

which is more informative than just absolute value of $|\lambda|$ because it accounts for the fact that using a smaller time step (hence smaller Courant number α) requires more steps to cover the same distance; and numerical-to-ideal phase speed ratio,

$$\text{since} \quad \lambda \equiv |\lambda| \cdot e^{\Delta\phi} \quad \text{while ideally} \quad \lambda^{(\text{exact})} = e^{-i\alpha k\Delta x} \quad \text{hence} \quad \frac{c^*}{c^{(\text{exact})}} = \frac{\Delta\phi}{\alpha\, k\Delta x}. \tag{2.16}$$

It is worth to note that despite having an apparent singularity when $\alpha \to 0$, dissipation measure $|\lambda|^{(1/\alpha)}$ is still finite and continuous. Indeed, any consistent discretization should yield $\lim\limits_{\alpha\to 0} |\lambda| = 1$, which means that asymptotically $|\lambda| \sim 1 - \beta\alpha^m$ as $\alpha \to 0$ where β is some coefficient and power m depends on the order of accuracy. Then one can expand

$$|\lambda|^{(1/\alpha)} = (1 - \beta\alpha^m)^{(1/\alpha)} = \exp\left\{\ln\left[(1 - \beta\alpha^m)^{(1/\alpha)}\right]\right\} = \exp\left\{\frac{1}{\alpha}\ln\left(1 - \beta\alpha^m\right)\right\}$$
$$= \exp\left\{-\beta\alpha^{m-1} + \mathcal{O}\left(\beta^2\alpha^{2m-1}\right)\right\} \begin{array}{l} \nearrow 1 - \beta\alpha^{m-1} + \mathcal{O}\left(\beta^2\alpha^{2m-2}\right), \quad \text{if } m \geq 2 \\ \searrow \exp\{-\beta\}, \qquad\qquad\qquad\quad \text{if } m = 1 \end{array} \tag{2.17}$$

which means that $\lim\limits_{\alpha\to 0} |\lambda|^{(1/\alpha)} = 1$ as long as $m \geq 2$. Such schemes are expected to be nondissipative in the limit of vanishingly small Courant number. Conversely $m = 1$ results in finite dissipation per $1\Delta x$ travel independently from α as long as α is sufficiently small.

The two bottom panels reveal that the algorithm has purely dispersive error at small Courant numbers: note that isolines of $c^*/c^{(\text{exact})}$ became vertical when approaching $\alpha = 0$ axis, which means that phase speed is independent from α for $\alpha \ll 1$. The dissipation is quadratically small in α when α is small. Increase of Courant number leads to the increase of dissipation and slight increase of phase speed over the entire range of wavenumbers – this is due to the LF-AM3 stepping irrespective from the particular spatial difference scheme as long as it is symmetric (not upstream-biased). Fourier components in the vicinity of $k\Delta x = \pi$ are neither dissipated, nor move. Consistently with Fig. 1, the largest phase increments are for wavenumbers in the vicinity of $k\Delta x = 2\pi/3$. The stability limit is $\alpha_{\max} = 1.5874/\sqrt{3} \approx 0.917$.

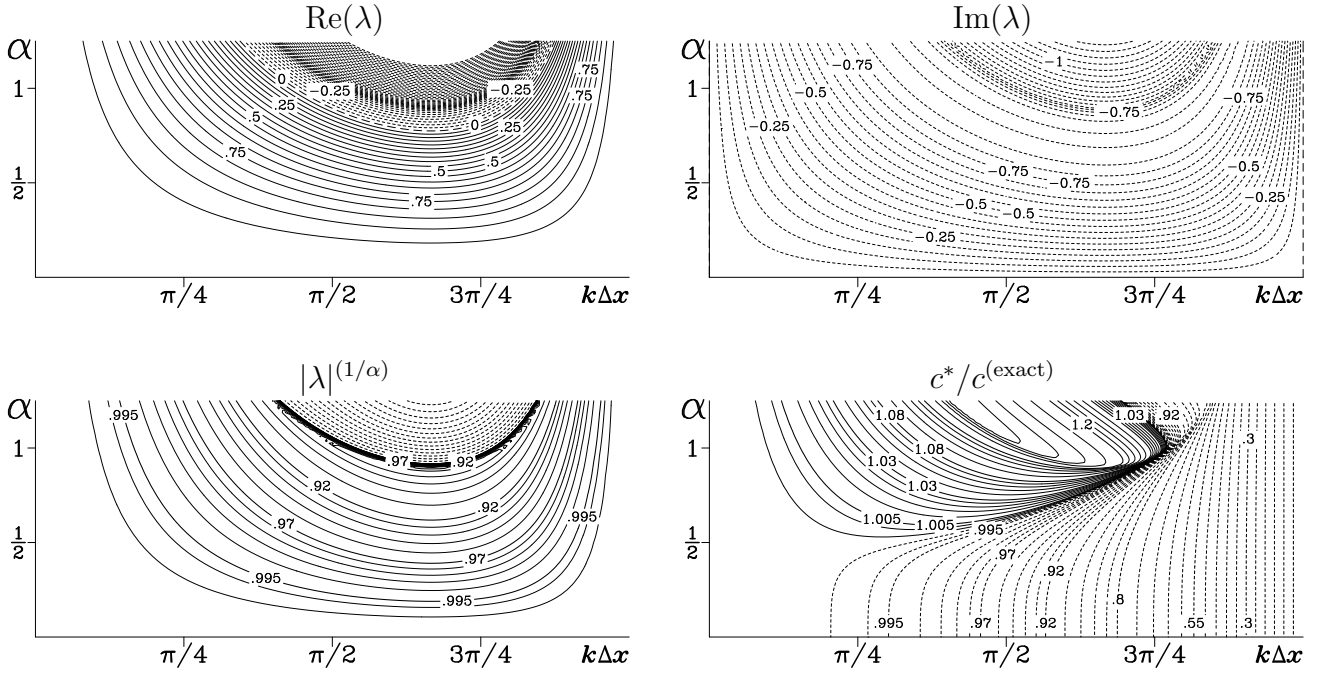Similarly, for the semi-Lagrangian algorithm (2.2), (2.5), (2.8) inserting (2.11) into (2.8) and sub-

7

Fig. 2. Real (*upper left panel*), imaginary (*upper right*) parts of step multiplier λ, amplitude multiplier normalized per $1\Delta x$ travel $|\lambda|^{(1/\alpha)}$ (*lower left*), and numerical to "ideal" phase speed ratio (*lower right*) for LF-AM3 advection algorithm (2.6)-(2.7) plotted as functions of normalized wavenumber $k\Delta x$ and Courant number α. Ideally both $|\lambda|^{(1/\alpha)}$ and $c^*/c^{(\text{exact})}$ should be uniformly equal to 1. Note that these two panels use non-uniform contour interval to allow very fine resolution in the vicinity of 1, while avoiding cluttering elsewhere (see Fig. 3). Thus, the contour interval is only 0.002 for values close to 1 (within the range of 0.995 to 1.005) however contour levels are selected to skip the exact value 1 half-way in between. Hence, as $|\lambda|^{(1/\alpha)} \equiv 1$ along both axes, $\alpha = 0$ and $k\Delta x = 0$, as well as along $k\Delta x = \pi$, the entire area below, left, and right from the lowest contour line (0.999 value) has values between 0.999 and 1 (that is within only 0.1% less than the ideal). Dashed contours in the amplitude plot corresponds $|\lambda| > 1$, which means that the algorithm is unstable within this area. Similarly, the contour-free area on the left and lower-left portion of $c^*/c^{(\text{exact})}$ has values within the range of 1 ± 0.001. Dashed contours in phase speed plot indicate slower phase speeds relative to its exact value, while solid indicate moving faster.
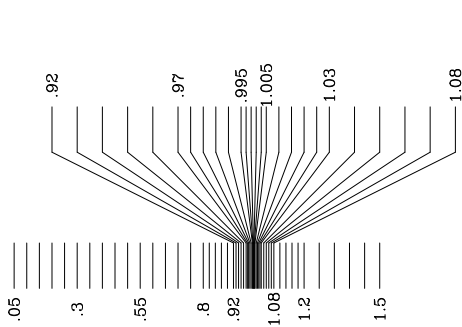

Fig. 3. Explanation of the nonuniform contour interval in $|\lambda|^{1/\alpha}$ and $c^*/c^{(\text{exact})}$ panels in Fig. 2. The lower portion shows placement of the selected contour levels in linear scale ranging from 0 to 1.5. Every fifth contour uses bold line and has a label. The interval is uniform between the labelled contours, but is allowed to change from one labelled line to the next by a factor of 2, 2.5, or 2.4 to allow refinement in the vicinity of $c^*/c^{(\text{exact})} \approx 1$. The upper portion is the vicinity of 1 magnified by a factor of 10. All $c^*/c^{(\text{exact})}$ panels in all figures throughout this article use the set of contour levels exactly as on the left. All $|\lambda|^{(1/\alpha)}$ panels use only the portion between 0 and 1 from this set, while above 1 (*i.e.,* in the unstable zone) the interval is 0.1 uniformly.

216   stituting the outcome into (2.2) yields its step multiplier,

217
$$\lambda = 1 - i\alpha \cdot \frac{\sin(k\Delta x)}{(2/3) + (1/3)\cos(k\Delta x)} \cdot \left[1 - \frac{\alpha^2}{3}\Big(1 - \cos(k\Delta x)\Big)\right]$$
$$- \alpha^2 \cdot \frac{1 - \cos(k\Delta x)}{(2/3) + (1/3)\cos(k\Delta x)} \cdot \left[1 - \frac{\alpha}{3}\Big(1 - \cos(k\Delta x)\Big)\right]. \tag{2.18}$$

8

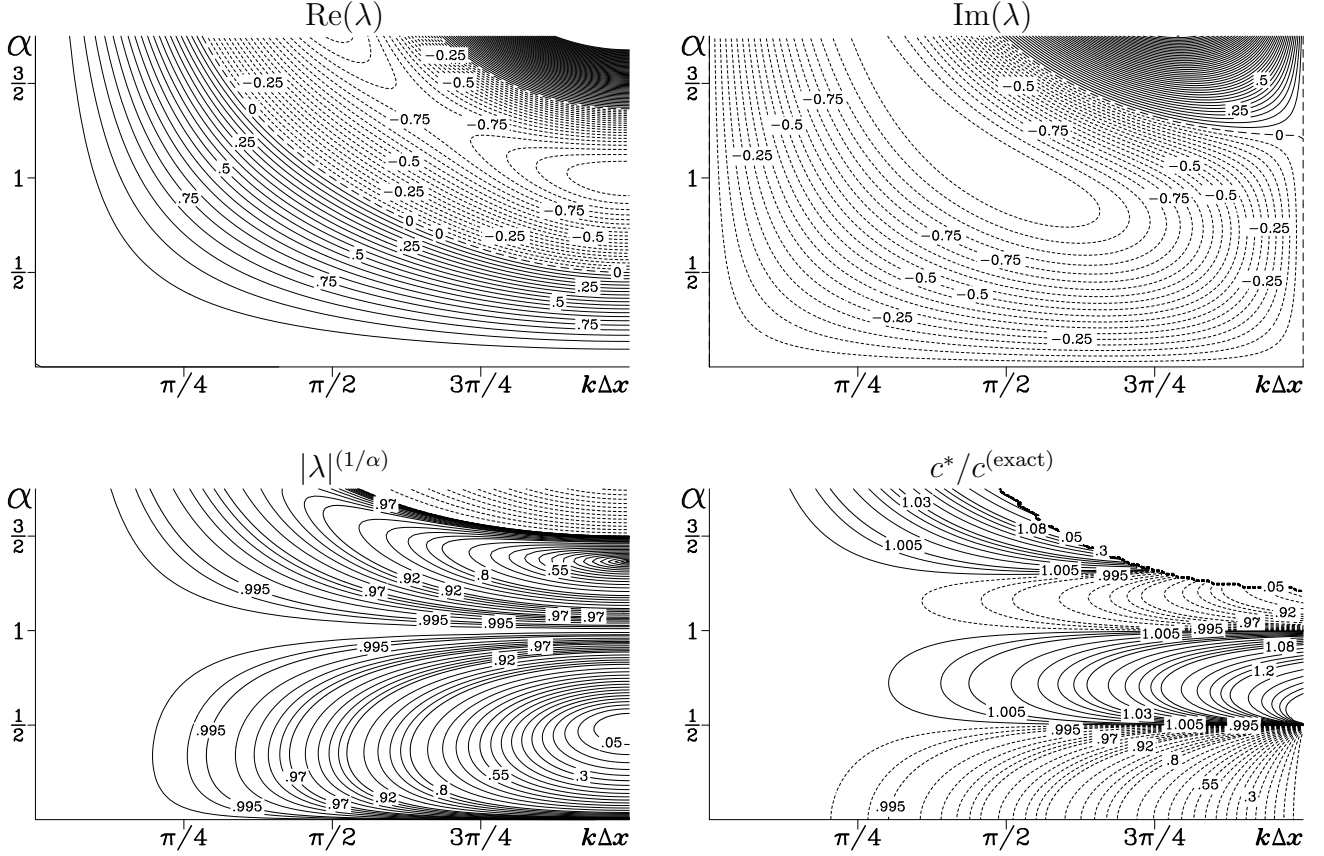The behavior of (2.18) is very complex and is studied in Fig. 4. At first, we note that substitution of

Fig. 4. Same as Fig. 2, but for semi-Lagrangian algorithm (2.2)-(2.8). The area free of contour lines on the left portion of $|\lambda|$ and $c^*/c^{\text{(exact)}}$ plots is due to having values very close to 1: in the case of amplitude the left-most contour is 0.999, while the free area on the phase speed plot has values within the range of 1 ± 0.001. Because when $\alpha = 1$, $\lambda = \lambda(k\Delta x, \alpha)$ given by Eq. (2.18) becomes exact, the free area is protruded all the way to the right in both plots along $\alpha = 1$ line (especially noticeable on $|\lambda|^{(1/\alpha)}$). The absence of contour lines in the upper-right corner of $c^*/c^{\text{(exact)}}$ plot is die to the fact that $\text{Im}(\lambda)$ changes sign from negative to positive, which means that the phase angle cannot be uniquely defined on the portion of $(k\Delta x, \alpha)$ plane zero-contour line of $\text{Im}(\lambda)$, lower-left, due to π and $-\pi$ ambiguity. Note that the stability limit for this algorithm $\alpha_{\max} = 3/2$. Also note that when $\alpha = 1/2$ the phase error vanishes identically for all $k\Delta x$; the amplitude is equal to zero at $(k\Delta x = \pi, \alpha = 1/2)$ (as it should) while the phase speed is discontinuous at this point resulting in contraction of contour lines there.

$\alpha = 1$ into the above leads to $\lambda = -i\sin(k\Delta x) + \cos(k\Delta x) = e^{ik\Delta x}$ which is the exact value. This is expected, because substituting $\alpha = 1$ into (2.8) turns it into $F_{j+1/2} = cq_j^n$, *i.e.,* making it equivalent to the upstream donor-cell scheme, which has the property of being exact as Courant number reaches 1 by merely resulting $q_j^{n+1} = q_{j-1}^n$ when this is substituted into (2.2). Then a Taylor expansion of (2.18) for $k\Delta x \ll 1$ yields,

$$\lambda = \underbrace{1 - i\alpha k\Delta x - \alpha^2 \frac{(k\Delta x)^2}{2} + i\alpha^3 \frac{(k\Delta x)^3}{6}}_{\text{match } \lambda^{\text{(exact)}} \text{ Eq. (2.10)}} - \underbrace{\alpha^2(1-2\alpha)\frac{(k\Delta x)^4}{24}}_{\text{vs. } +\alpha^4 \frac{(k\Delta x)^4}{24}} + \underbrace{i\alpha\left(1 - \frac{5\alpha^2}{2}\right)\frac{(k\Delta x)^5}{180}}_{\text{vs. } -i\alpha^5 \frac{(k\Delta x)^5}{120}} + ...,$$

(2.19)

9

which indicates that similarly to (2.10) α and $k\Delta x$ appear together in equal powers for up to $\alpha^3(k\Delta x)^3$ power, while the mismatch in $(k\Delta x)^4$ corresponds to the third-order accuracy. The leading-order truncation term (*i.e.,* deviation from the "ideal" $(k\Delta x)^4$-term) can be classified as dissipation of hyperdiffusive type, because in "bends" λ toward the interior of unit circle, as

$$-\alpha^2(1-2\alpha)\frac{(k\Delta x)^4}{24} - \alpha^4\frac{(k\Delta x)^4}{24} = \underbrace{-\alpha^2(1-\alpha)^2}_{\text{negative definite}}\frac{(k\Delta x)^4}{24} \tag{2.20}$$

with the maximum dissipation expected at $\alpha = {}^1\!/_2$, and vanishing again as Courant number reaches $\alpha = 1$. Here it is worth noting that the above Taylor expansion implies only smallness $k\Delta x \ll 1$, but makes no assumption about smallness of α, so setting $\alpha = 1$ above leads to the exact multiplier, including both $(k\Delta x)^4$ and $(k\Delta x)^5$ terms which mimics the property semi-Lagrangian schemes to be exact when Courant number reaches unity. Once Courant number departs from $\alpha = 0$, but still $\alpha \ll 1$, the hyperdiffusive effect is quadratically small with respect to α. In this respect, it is somewhat similar to Lax-Wendroff type scheme, rather than more typical semi-Lagrangian scheme QUICKEST (see Appendix A). Furthermore, comparing λ above with its "ideal" counterpart (2.10) reveals that

$$\frac{\lambda}{\lambda^{(\text{exact})}} = -\alpha^2(1-\alpha)^2\frac{(k\Delta x)^4}{24} + i\alpha\left(1 - 10\alpha^2 + 15\alpha^3 - 6\alpha^4\right)\frac{(k\Delta x)^5}{180} + ..., \tag{2.21}$$

where the entire r.h.s. can be interpreted as a numerically-induced "parasitic" step multiplier responsible for the appearance of numerical distortion. The $i\alpha(...)\,(k\Delta x)^5$-term indicates dispersive error and causes phase delay of high wavenumbers when $\alpha \ll 1$, as it acts against the "true" $i\alpha^5$-term in $\lambda^{(\text{exact})}$. Its appearance and magnitude can be traced back to (2.13) and is associated with the truncation error of the spatial differencing operator (see also Fig. 1). However, the quartic polynomial in powers of α inside the brackets in $i\alpha(...)\,(k\Delta x)^5$-term vanishes when $\alpha = {}^1\!/_2$, $\alpha = 1$, and $\alpha \approx 1.26376261582$, changing sign at each zero crossing. This corresponds to zero phase error and explains the appearance of alternating horizontal bands in the lower-right panel in Fig. 4. The stability limit is $\alpha_{\max} = 3/2$ instead of 1 more typical semi-Lagrangian schemes: assuming that the algorithm becomes unstable first for $2\Delta x$ mode (this assumption is confirmed by Fig. 4), we may simply substitute $k\Delta x = \pi$ into (2.18), which turns it into

$$\lambda = 1 - \alpha^2(6 - 4\alpha). \tag{2.22}$$

This function starts as $\lambda = 1$ together with $\partial\lambda/\partial\alpha = 0$ when $\alpha = 0$. Then it crosses $\lambda = 0$ at $\alpha = 1/2$, proceeds toward $\lambda = -1$ at $\alpha = 1$ which is an extremum as $\partial\lambda/\partial\alpha = 0$ there, then crosses $\lambda = 0$ again at $\alpha \approx 1.3660254$, and finally, crosses $\lambda = 1$ at $\alpha = \alpha_{\max} = 3/2$ thereafter exceeding $|\lambda| = 1$. The dissipation vanishes as Courant number becomes vanishingly small: in fact, its $|\lambda|^{(1/\alpha)} \to 1$ as $\alpha \to 0$, where it should be emphasized that for any numerically consistent method it is expected that $|\lambda| \to 1$ however $|\lambda|^{(1/\alpha)}$ involves ambiguity of $1^{+\infty}$ type making this property be more restrictive. In comparison with Fig. 2 the area where phase errors are within 1 ± 0.005 is much wider, both in terms of $k\Delta x$ range, and Courant numbers as well. The most distinctive feature of semi-Lagrangian algorithms is the identically vanishing phase error at both $\alpha = 1/2$ and $\alpha = 1$ as well.

To further illustrate the properties and evaluate performance of the two algorithms we setup a test
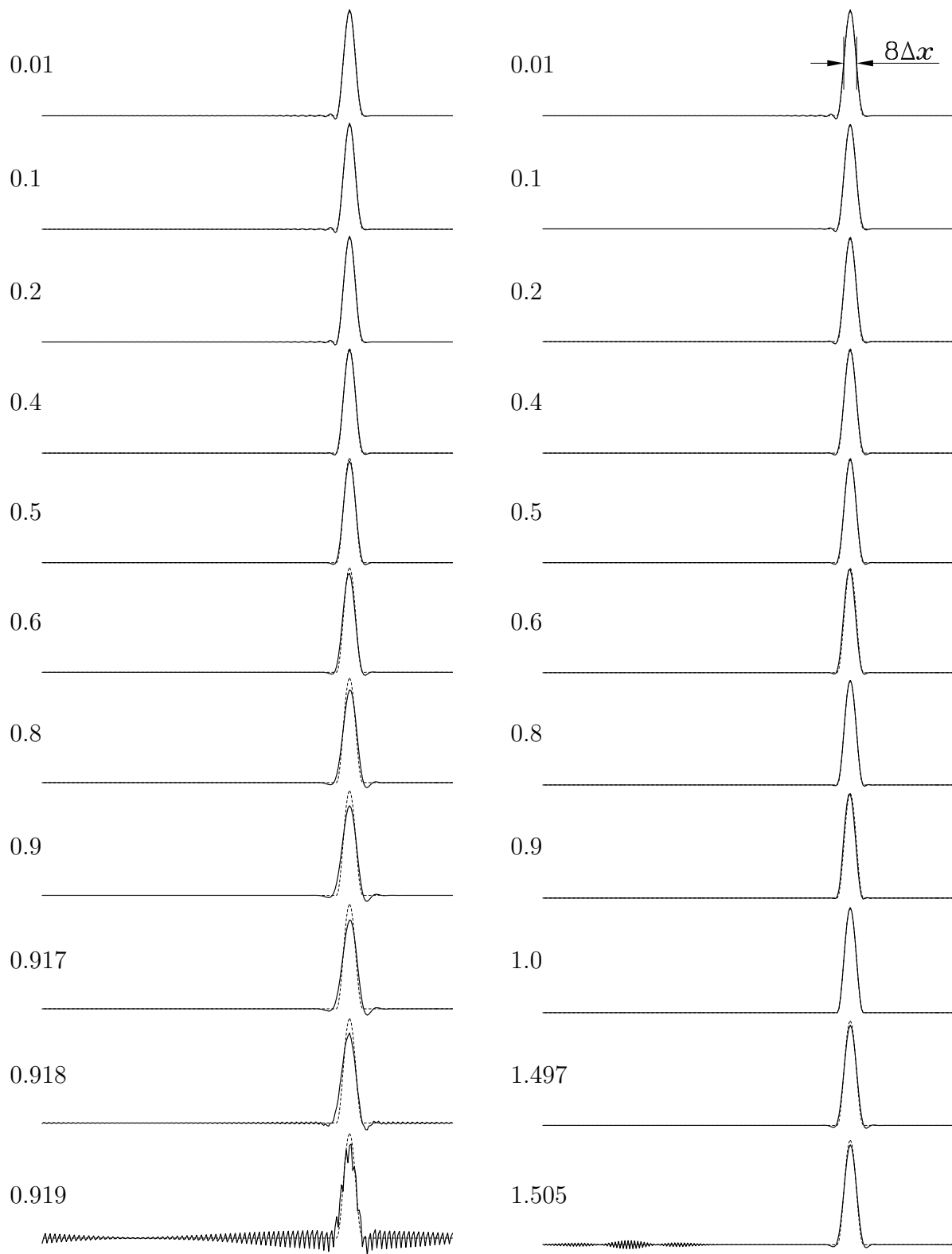
Fig. 5. Advection a narrow pulse by LF-AM3 (*left column*) and semi-Lagrangian (*right*) algorithms. Number on the left of each panel indicates Courant number, $\alpha = c\Delta t/\Delta x$. Bold solid line indicates numerical solution, dashed line exact solution. 256 points in all cases, left-right periodic boundary conditions, results are shown after one period (256-point travel, moving from left to right).

problem by initializing $q = q(x)$ as a narrow pulse,

$$q(x)\Big|_{t=0} = \begin{cases} \left[\cos\left(\frac{\pi}{2} \cdot \frac{x - x_0}{\sigma}\right)\right]^2, & |x - x_0| < \sigma \\ 0, & \text{otherwise} \end{cases} \quad \text{with} \quad \begin{cases} x \in [0, 1], & \Delta x = 1/256 \\ x_0 = 3/4, & \sigma = 1/32 \end{cases} \quad (2.23)$$

which is only $8\Delta x$-wide as measured at half of its height. The advecting velocity $c = 1$ is positive, hence the flow direction is from left to right. Periodic boundary conditions are assumed at the ends, so the pulse moves to the right, exits at the end and re-enters from the left. The duration of the test is exactly one period, so the exact solution should be the same as the initial condition. The results are shown on Fig. 5 covering the entire range of stability for each algorithm. Note that:

(*i*) slight trailing-edge oscillations in both cases when Courant numbers are small, these are explained by the predominantly dispersive nature of truncation error if $\alpha \to 0$ – delay due to $\mathscr{K} = \mathscr{K}(k\Delta x)$ stays below the ideal line;

(*ii*) perfectly symmetric overshoots for $\alpha = 0.5$ in semi-Lagrangian case (zero numerical dispersion);

(*iii*) LF-AM3 becoming more and more dissipative toward its limit of stability; semi-Lagrangian is exact at $\alpha = 1$;

(*iv*) leading-edge overshoots becoming more and more noticeable in the case of LF-AM3 for Courant numbers above 0.6. This maps onto the over-speeding region in $c^*/c^{(\text{exact})}$ panel Fig. 2; semi-Lagrangian profiles are free of this effect;

(*v*) numerical instability of LF-AM3 scheme due to exceeding the allowed Courant number occurs first at $3\Delta x$ mode corresponding to the highest point on the dispersion curve in Fig. 1, rather than $2\Delta x$ of the semi-Lagrangian scheme.

Overall both algorithms maintain the shape of the pulse in all computational regimes within their ranges of stability.

2.2 *Implicit advection*

Our next goal is to survey the properties of implicit advection schemes. While, in principle, implicit time stepping can be made of an arbitrary high order of accuracy in time, making it unconditionally stable imposes a fundamental restriction of to be no more than the second-order, if the algorithm has pre-determined constant (non-adaptive) coefficients associated with the time stepping itself, *i.e.,* belongs to the class of linear multistep methods (Dahlquist, 1963, see Theorem 2.2 there), essentially reducing the possibilities to trapezoidal integration rule (including stepping over just one or more time steps Δt), backward Euler step, and positively-weighted linear combinations of them. Crank and Nicolson (1947) stepping combined with second-order centered spatial differencing,

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} + c \cdot \left\{\theta \cdot \frac{q_{j+1}^{n+1} - q_{j-1}^{n+1}}{2\Delta x} + (1 - \theta) \cdot \frac{q_{j+1}^n - q_{j-1}^n}{2\Delta x}\right\} = 0 \quad (2.24)$$

is, perhaps, the most widely known method of this kind. Each of the unknown q_j^{n+1} depends on its left and right neighbors, so the above set of equations constitutes a tri-diagonal linear system for the entire set $\left\{q_j^{n+1} | \forall j\right\}$ which, unlike (2.5), is no longer diagonally dominant if $\theta \cdot c\Delta t/\Delta x > 1$. Therefore the numerical stability of a standard Gaussian elimination procedure Richtmyer and Morton (1967, Sec.

8.5) is no longer guaranteed. In all the computations presented here we use cyclic reduction algorithm, *cf.,* Buzbee et al. (1970).

For a Fourier component $e^{ik\Delta x j}$ the step multiplier of algorithm (2.24) is

$$\lambda = \frac{1 - \alpha(1-\theta) \cdot i \sin(k\Delta x)}{1 + \alpha\theta \cdot i \sin(k\Delta x)} \tag{2.25}$$

It can be instantly verified that $\theta = 1/2$ makes the scheme non-dissipative $|\lambda| \equiv 1$ independently of k, $|\lambda| < 1$ if $\theta > 1/2$ meaning unconditional stability, conversely $|\lambda| > 1$ if $\theta < 1/2$. The asymptotic limit $\lambda \to -(1-\theta)/\theta$ if $\alpha \to \infty$ indicates that the phase of any Fourier component receives an increment not exceeding π per time step, and a finite decrease in amplitude if $1/2 \le \theta < 1$.

Furthermore, for $\theta = 1/2$ and $k\Delta x \ll 1$ expanding λ in Taylor series and comparing it with the ideal step multiplier yields,

$$\lambda - e^{-ick\Delta t} \equiv \lambda - e^{-i\alpha k\Delta x} = +i\alpha \frac{(k\Delta x)^3}{6} + i\alpha^3 \frac{(k\Delta x)^3}{12} + \dots \tag{2.26}$$

which indicates dispersive nature of the leading-order truncation term. It should be noted that in the case of ideal step multiplier $e^{-i\alpha k\Delta x}$ all powers of α and $k\Delta x$ always stay together, resulting in no mixed-power terms like $i\alpha(k\Delta x)^3/6$, which is the dominant factor responsible for the behavior of this scheme in term of the character of numerical distortion it generates in small Courant number regime. Both $(k\Delta x)^3$ terms are imaginary of the same sign (positive) leading to delay of higher wavenumbers. The outcome immediately shows up in the results of the test problem (2.23), Fig. 7, left column: trailing-edge dispersion ripples within the entire range of Courant numbers.

A compact version of (2.24) can be obtained by "spreading" the time derivative horizontally along x-direction,

$$\frac{1}{6} \cdot \frac{q_{j-1}^{n+1} - q_{j-1}^{n}}{\Delta t} + \frac{2}{3} \cdot \frac{q_{j}^{n+1} - q_{j}^{n}}{\Delta t} + \frac{1}{6} \cdot \frac{q_{j+1}^{n+1} - q_{j+1}^{n}}{\Delta t} + c \cdot \left\{ \theta \frac{q_{j+1}^{n+1} - q_{j-1}^{n+1}}{2\Delta x} + (1-\theta) \frac{q_{j+1}^{n} - q_{j-1}^{n}}{2\Delta x} \right\} = 0 \,, \tag{2.27}$$

which at the first glance may seem to be *ad hoc*, however is should be noted that the primary source of numerical error in (2.24) is non-staggered second-order differencing over $2\Delta x$. Replacing it with fourth-order compact derivative $d_j = \partial_x q\big|_{x=x_j} + \mathcal{O}\left(\Delta x^4\right)$ leads to the system

$$\begin{aligned} \frac{q_{j}^{n+1} - q_{j}^{n}}{\Delta t} + c \cdot d_{j}^{n+1/2} &= 0 \\ \frac{1}{6} d_{j-1}^{n+1/2} + \frac{2}{3} d_{j}^{n+1/2} + \frac{1}{6} d_{j+1}^{n+1/2} &= \frac{q_{j+1}^{n+1/2} - q_{j-1}^{n+1/2}}{2\Delta x} \end{aligned} \quad \text{where} \quad \begin{cases} q_{j}^{n+1/2} = \theta q_{j}^{n+1} + (1-\theta) q_{j}^{n} \\ d_{j}^{n+1/2} = \theta d_{j}^{n+1} + (1-\theta) d_{j}^{n} \end{cases} \tag{2.28}$$

where both q_j^{n+1} and d_j^{n+1} are the unknowns. One can use the first equation to express $d_j^{n+1/2}$ via q_j^{n+1} and q_j^n, substitute this into the second, and thus algebraically exclude all $d^{n+1/2}$ terms to obtain an implicit system for q^{n+1} alone

$$\left(\frac{1}{6} - \frac{c\Delta t}{2\Delta x}\theta \right) \cdot q_{j-1}^{n+1} + \frac{2}{3} \cdot q_{j}^{n+1} + \left(\frac{1}{6} + \frac{c\Delta t}{2\Delta x}\theta \right) \cdot q_{j+1}^{n+1} = \text{known terms involving } q^n, d^n \tag{2.29}$$
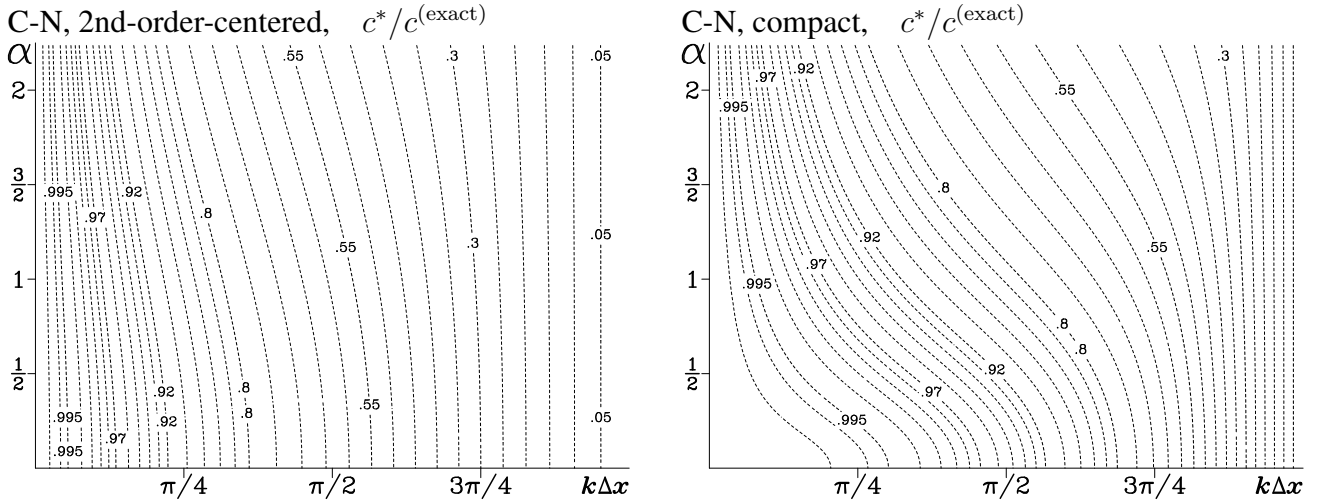
13

Fig. 6. Comparison of numerical-to-ideal phase speed ratios for Crank-Nicolson algorithms (2.24), *left* vs. (2.27), *right*. Note the format is exactly the same as the lower-right panels of Figs. 2 and 4, however the contour-free area on the lower-left portion of the plots is now much narrower, even for the case shown on the right.

which in fact is very similar to the system stemming from (2.27). Either way, the step multiplier is

$$\lambda = \frac{1 - \alpha(1-\theta) \cdot i\mathscr{K}\Delta x}{1 + \alpha\theta \cdot i\mathscr{K}\Delta x} \tag{2.30}$$

which differs from (2.25) by replacing $i\sin(k\Delta x)$ with $i\mathscr{K}\Delta x$ defined in (2.12),

$$i\sin(k\Delta x) \;\rightarrow\; i\mathscr{K}\Delta x = \frac{i\sin(k\Delta x)}{(2/3) + (1/3)\cos(k\Delta x)} \,. \tag{2.31}$$

This eliminates the $i\alpha(k\Delta x)^3$ truncation term, and, in fact, now powers of α and $(k\Delta x)$ are the same for all terms of powers less than $(k\Delta x)^5$. The numerical-to-ideal phase speed ratios for algorithms (2.24) and (2.27) are compared on Fig. 6. Overall, there is a major improvement for $\alpha \ll 1$ which is expected, however the gain in accuracy degrades quickly once α goes beyond $1/2$, making the scheme far less accurate than either of the explicit algorithms considered previously.

An alternative approach to reduce truncation error of (2.24) is to discretized (2.1) using on Taylor expansion centered about midpoint in both time and space, $(x,t) = \left(x_j - \frac{\Delta x}{2}, t_n + \frac{\Delta t}{2}\right)$,

$$\frac{q_j^{n+1} - q_j^n + q_{j-1}^{n+1} - q_{j-1}^n}{2\Delta t} + c \cdot \left\{ \theta \cdot \frac{q_j^{n+1} - q_{j-1}^{n+1}}{\Delta x} + (1-\theta) \cdot \frac{q_j^n - q_{j-1}^n}{\Delta x} \right\} = 0 \,, \tag{2.32}$$

which is to be interpreted as the implicit equation for index j if $c > 0$. Conversely, if $c < 0$, then it should involve q_j and q_{j+1}. Its step multiplier

$$\lambda = \frac{\frac{1}{2} - (1-\theta) + \alpha\left(\frac{1}{2} + (1-\theta)\right) e^{-ik\Delta x}}{\frac{1}{2} + \theta \;\; + \alpha\left(\frac{1}{2} - \theta\right) e^{-ik\Delta x}} \tag{2.33}$$

reveals that it is unconditionally stable as long as $\theta \geq 1/2$. If $\theta = 1/2$ it becomes second-order accurate in space-and-time and non-dissipative, $|\lambda| \equiv 1$. Furthermore, substitution of $\alpha = 1$ and $\theta = 1/2$ into (2.32) yields

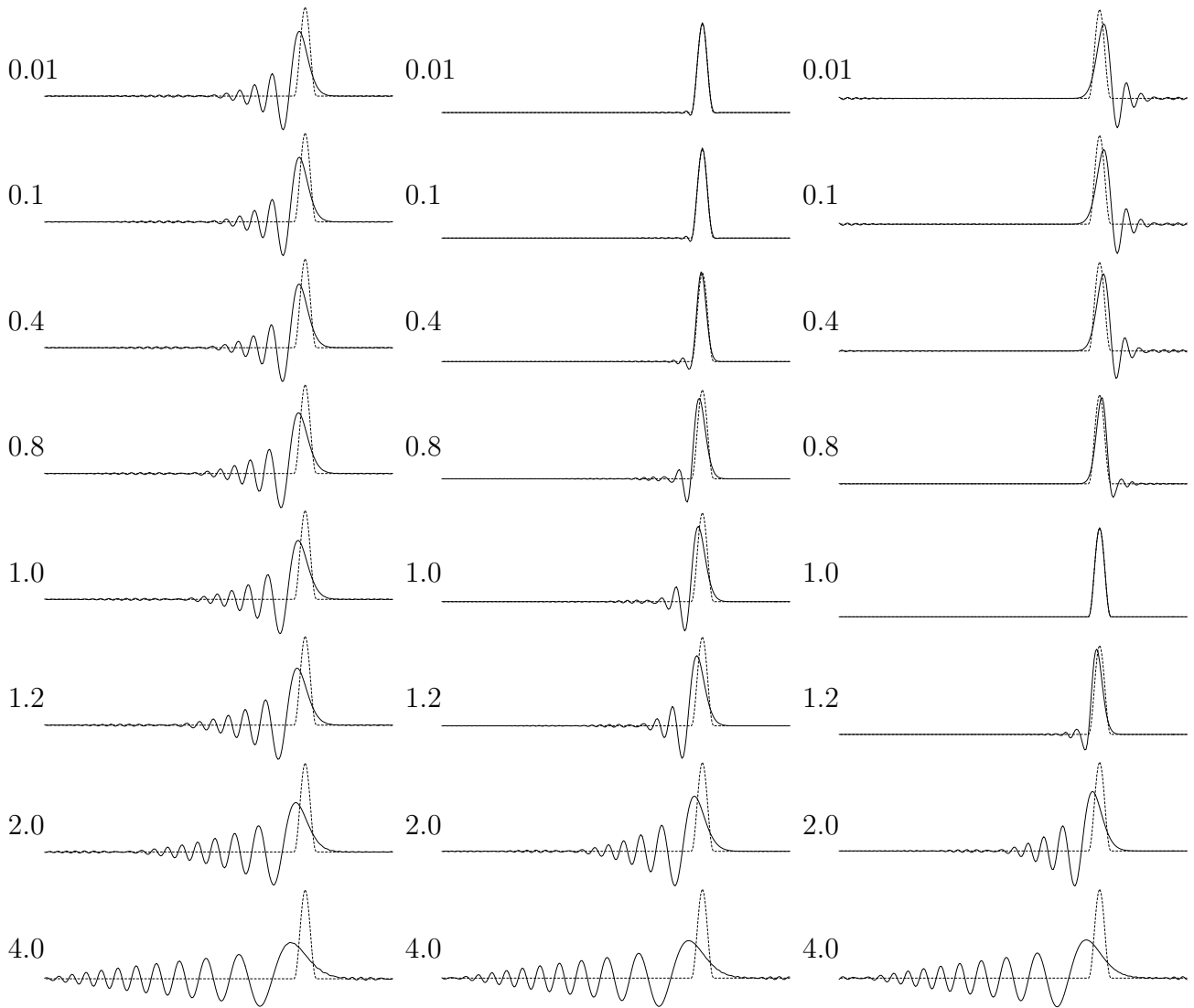$$q_j^{n+1} = q_{j-1}^n \,, \tag{2.34}$$

14

Fig. 7. Advection and dispersive spreading of a narrow pulse by non-dissipative, unconditionally stable implicit schemes using different Courant number regimes. The format is the same as in Fig. 5. Equal-weight ($\theta = 1/2$) Crank-Nicolson time stepping is used in all three cases. *Left column* second-order centered differencing in space (2.24); *middle* fourth-order compact scheme (2.27); *right* staggered in time-and-space, centered at $(x, t) = (j - 1/2, n + 1/2)$ scheme, (2.32).

344 which is exact. This means that the scheme has vanishing phase error in the vicinity of crossing $\alpha = 1$.

345   The results for the test problem (2.23) using all three implicit algorithms (2.24), (2.27), and (2.32)
346 are compared on Fig. 7. For small Courant numbers the middle-column is the most accurate; $\alpha = 1$
347 solution is exact in the right column, this scheme also produces the least dispersive spreading for $\alpha = 2$,
348 however, all three schemes produce virtually the same dispersion for $\alpha = 4$ and above. Quadratic
349 variance is maintained to machine accuracy by all three schemes in all three cases, which means that
350 the numerical errors are of purely dispersive nature. The fundamental limitation of any implicit scheme
351 is that for any given wavenumber k the step multiplier λ starts as $\lambda = 1$ when $\alpha = 0$ and follows
352 the lower half of unit circle asymptoting toward $\lambda \to -1$ when $\alpha \to \infty$. This applies to all three
353 algorithms, *cf.,* (2.25), (2.30), and (2.33). As the result no Fourier component is allowed to change its
354 phase beyond $-\pi$ in one time step (the phase increment is negative if $c > 0$), which means that there
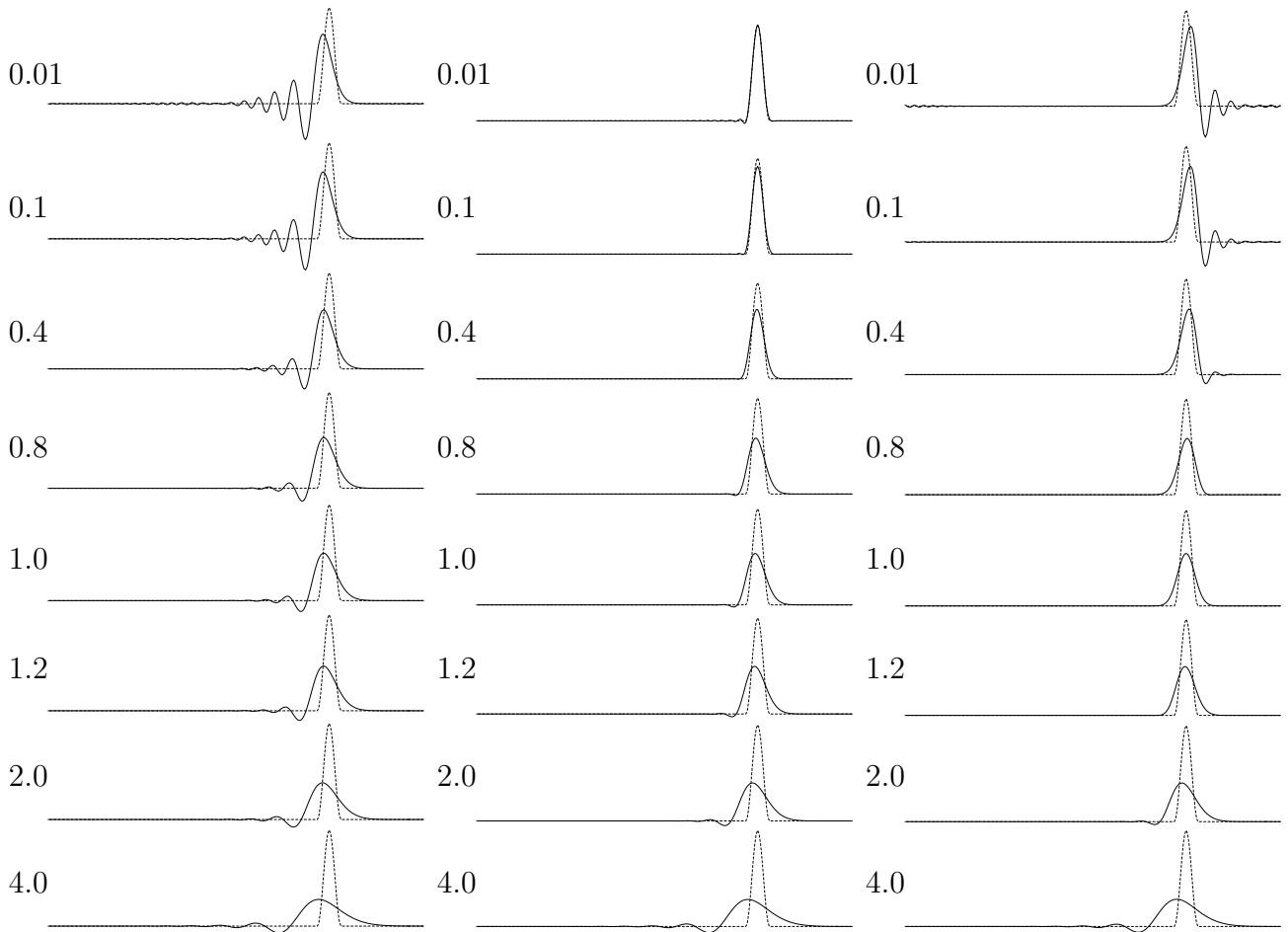
15

Fig. 8. Same as Fig. 7, but for $\theta = 0.55$ in all three cases.

is no other choice, but to slow down the propagation of highest wavenumber components – the ones for which $\alpha \cdot k\Delta x$ exceeds a threshold depending on the specifics of the particular algorithm. On the other hand, designing an algorithm which lets the phase increment to exceed π for some wavenumbers means admitting aliasing errors. Clearly, none of the behaviors shown on Fig. 7 are acceptable in practice, if Courant number exceeds $\alpha = 1$.

The only viable alternative is to selectively dissipate Fourier components propagation of which cannot be resolved in time. This can be achieved by choosing $\theta > 1/2$ which biases the coefficients of Crank-Nicolson time step toward backward Euler. Fig. 8 shows the results for $\theta = 0.55$. In comparison with Fig. 7 it makes little influence for small Courant numbers, which is expected, and makes virtually no change in dispersive properties, $c^*/c^{(\text{exact})}$, for the entire $(k\Delta x, \alpha)$-plane for all three schemes. The difference is most significant for the largest Courant numbers, where the oscillatory behavior is changed to dissipative. Some oscillations are still noticeable in the range $\alpha = 0.8...2$ even in the case of compact scheme in the middle. While further increase of θ would lead to a more dissipative behavior with less oscillations, it should be noted that $\theta = 0.55$ compact scheme is already more dissipative than either LF-AM3 or semi-Lagrangian explicit algorithms presented in Fig. 5 for the Courant number regimes starting at $\alpha \sim 0.4$. At the same time we note that dispersive errors of the nondissipative algorithm in Fig. 7, middle column, are also larger than on Fig. 5 for the same values of αs. Some dissipation is needed to control the oscillations, however using $\theta > 1/2$ is too non-selective: too much dissipation for $\alpha \sim 0.5$ while not enough beyond $\alpha = 1$.

16

In contrast explicit schemes can be designed to be time-space accurate for the entire range of their stability. This is explained by the tendency of canceling the phase errors due to time and space differencing once Courant number departs $\alpha \ll 1$, especially in the case of semi-Lagrangian algorithm (Fig. 4, lower-right panel and caption). Although to a lesser degree, there is also such tendency for LF-AM3 stepping because when $\alpha \ll 1$ the truncation error of time differencing is very small because it is third-order in time, but once α departs from being vanishingly small, the truncation error of LF-AM3 stepping introduces phase lead (*cf.,* Fig. 20 from SM2005) which partially compensates the delay caused by truncation error of spatial differencing[4]. In contrast, the second-order truncation error of Crank-Nicolson step (as well as its $\theta > 1/2$ variant) introduces additional phase delay to already existing delay due to spatial differencing. Consequently, the range of Courant numbers α which produce numerically accurate solutions is much narrower in the case of implicit schemes considered here.

3   Adaptively Implicit Advection

Because of the fundamental limitations set by Dahlquist (1963) it is impossible for a Linear Multi-step Method (LMM) to achieve unconditional stability while having higher than second-order temporal accuracy. On the other hand, as shown in the previous section, retaining third-order is highly desirable if a high-order spatial differencing is used to approximate advection operator. This conflict can be resolved by constructing an algorithm which is outside the class of LMM methods by allowing the coefficients of the implicit scheme to depend on Courant number, which means that the coefficients are adaptive. In this section we describe such an algorithm.

3.1   *Algorithm*

Implicit vertical advection fluxes for the tracer or velocity fields are discretized in such a way that their computation involve the advected field at the new time step, q_k^{n+1} which is yet unknown[5]

$$FC_{k+1/2} = W_{k+1/2} \cdot \mathscr{Q}\left(q_k^{n+1}, q_{k+1}^{n+1}, \tilde{q}_{\ k+1/2}^{\ n+1/2}\right), \tag{3.1}$$

---

[4] We should note that the analysis performed in this section may leave an impression that logically-forward-in-time semi-Lagrangian approach is far superior to LF-AM3 stepping (or, in fact, to any algorithm with spatial and temporal discretization done independently) in terms of accuracy per computational effort. However, our analysis is performed in one dimension. Generalization of LF-AM3 stepping to 3D is straightforward. In contrast, a semi-Lagrangian approach require introduction of cross-terms in order to avoid flux-splitting instability (Leonard et al., 1996) which dramatically increase its complexity and computational cost, therefore making the two classes of advection algorithm quite competitive.

[5] Similarly to the convention adopted in SM2005 (footnote 10 there), index k is used exclusively for the vertical direction while i and j for the horizontal. All three indices i, j, k may be either whole- or half-integer (*e.g.,* $k + 1/2$). Whole-integer index means that the variable is located in the middle of grid box centered around scalar point, while half-integer means that it is located at the interface between the two adjacent grid boxes. For compactness of notation we may omit i, j in equations if they are the same for all the indexed variables in the equation, and the variables are placed on the same location on the horizontally staggered grid, *i.e.,* all the operations take place within the same vertical column. Indices i, j are allowed to disappear and reappear as needed if horizontal differencing or interpolation takes place.

where $W_{k+1/2}$ is finite-volume vertical velocity flux in generalized sigma-coordinate sense; while half-integer-indexed in both vertical direction and time, $\tilde{q}_{k+1/2}^{\,n+1/2}$, are the flux values of an explicit advection algorithm usually involving a high-order spatial interpolation from grid-box-centered to interface location of the provisional field $q_k^{n+1/2}$, $\quad k = 1, ..., N$ which in its turn is computed by predictor sub-step, or interpolated from upstream values in the case of semi-Lagrangian advection (SM2005, Sec. 4 there). In the case of predictor-corrector algorithm both stages need to be implicit in vertical direction, so (3.1) written for corrector stage also applies from predictor, except that all time indices are shifted back by 1/2, hence $FC_{k+1/2} = W_{k+1/2} \cdot \mathscr{D}\left(q_k^{n+1/2}, q_{k+1}^{n+1/2}, \tilde{q}_{k+1/2}^{\,n}\right)$, with $q^{n+1/2}$ treated as unknowns.

The above expression for advective flux (3.1) can be rearranged by splitting $W_{k+1/2}$ into two parts,

$$W_{k+1/2} = W_{k+1/2}^{(e)} + W_{k+1/2}^{(i)}, \qquad \forall k = 0, 1, ..., N \tag{3.2}$$

where $W_{k+1/2}^{(i)}$ participates only in computing terms involving q_k^{n+1}, $q_{k\pm1}^{n+1}$ only (*i.e.,* implicit part), while $W_{k+1/2}^{(e)}$ is for the remaining $\tilde{q}_{k+1/2}^{\,n+1/2}$. Then, the $W^{(e)}$-terms are computed within the already existing algorithm of r.h.s. terms for the momentum and tracer equations, while computation of $W^{(i)}$-terms is combined with the implicit operator for vertical viscosity and diffusion. Assuming upstream treatment of the implicit part,

$$FC_{k+1/2}^{(i)} = W_{k+1/2}^{(i)} \cdot \begin{cases} q_k^{n+1}, & \text{if } W_{k+1/2}^{(i)} > 0 \\ q_{k+1}^{n+1}, & \text{if } W_{k+1/2}^{(i)} < 0 \end{cases} \tag{3.3}$$

the combined implicit advection-diffusion system becomes:

$k = N$, uppermost grid box,

$$
\begin{aligned}
H_N^{n+1} q_N^{n+1} = {} & H_N^n q_N^n + \Delta t \cdot \text{rhs}_N' + \Delta t \cdot SRFRC - \Delta t \cdot A_{N-1/2} \frac{q_N^{n+1} - q_{N-1}^{n+1}}{\Delta z_{N-1/2}} \\
& + \frac{\Delta t}{\Delta \mathscr{A}} \left[\max\left(W_{N-1/2}^{(i)}, 0\right) q_{N-1}^{n+1} + \min\left(W_{N-1/2}^{(i)}, 0\right) q_N^{n+1} \right]
\end{aligned}
\tag{3.4}
$$

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

$$
\begin{aligned}
H_k^{n+1} q_k^{n+1} = {} & H_k^n q_k^n + \Delta t \cdot \text{rhs}_k' + \Delta t \cdot A_{k+1/2} \frac{q_{k+1}^{n+1} - q_k^{n+1}}{\Delta z_{k+1/2}} - \frac{\Delta t}{\Delta \mathscr{A}} \left[\max\left(W_{k+1/2}^{(i)}, 0\right) q_k^{n+1} \right. \\
& \left. + \min\left(W_{k+1/2}^{(i)}, 0\right) q_{k+1}^{n+1} \right] \\
& - \Delta t \cdot A_{k-1/2} \frac{q_k^{n+1} - q_{k-1}^{n+1}}{\Delta z_{k-1/2}} + \frac{\Delta t}{\Delta \mathscr{A}} \left[\max\left(W_{k-1/2}^{(i)}, 0\right) q_{k-1}^{n+1} \right. \\
& \left. + \min\left(W_{k-1/2}^{(i)}, 0\right) q_k^{n+1} \right]
\end{aligned}
\tag{3.5}
$$

$k = 1$, bottom grid box,

$$
\begin{aligned}
H_1^{n+1} q_1^{n+1} = {} & H_1^n q_1^n + \Delta t \cdot \text{rhs}_1' + \Delta t \cdot A_{3/2} \frac{q_2^{n+1} - q_1^{n+1}}{\Delta z_{3/2}} - \frac{\Delta t}{\Delta \mathscr{A}} \left[\max\left(W_{3/2}^{(i)}, 0\right) q_1^{n+1} \right. \\
& \left. + \min\left(W_{3/2}^{(i)}, 0\right) q_2^{n+1} \right] \\
& - \Delta t \cdot r_D \cdot q_1^{n+1}
\end{aligned}
\tag{3.6}
$$

where the prime in $\mathrm{rhs}'_k = \mathrm{rhs}'_{i,j,k}$ means that the usual r.h.s. computed by ROMS code for the corresponding equations, except the replacement $W_{k+1/2} \to W^{(e)}_{k+1/2}$. H^n_k is the high of grid box around q^n_k and $\Delta\mathscr{A} = \Delta\mathscr{A}_{i,j}$ is the grid-box area as seen from above (hence their product $H^n_{i,j,k}\Delta\mathscr{A}_{i,j} = \Delta\mathscr{V}_{i,j,k}$ is the control volume for $q^n_{i,j,k}$, while the presence of time index in H^n_k is due to time dependency of grid-box height due to changing free surface); $A_{k+1/2}$ is vertical viscosity/diffusion coefficient (including the stabilization terms (Lemarié et al., 2012, Sec. 3) in the case when isoneutral lateral diffusion is used). The above system takes into account kinematic boundary conditions at surface and bottom, $W_{N+1/2} = W_{1/2} = 0$. *SRFRC* is surface forcing (wind stress for the momentum equations or heat/freshwater flux for tracers). The last term $\Delta t \cdot r_D q^{n+1}_1$ in (3.6) is applicable for momentum equations only, $q \in \{u, v\}$. r_D is the bottom drag coefficient, which is, generally speaking, nonlinear and depends bottom-most velocity magnitude, $r_D = r_D\left(\sqrt{(u_1)^2 + (v_1)^2}\right)$ computed from the most recent explicitly known velocity values. At the same time, it is expected that $\Delta t \cdot r_D/H_1 > 1$ so the implicit treatment of the whole bottom drag term $\Delta t \cdot r_D \cdot u_1$ is essential for numerical stability without imposing non-physical restriction on r_D or reducing time step. No-flux bottom b.c. are assumed for tracer equations, (3.6) is the same except that there is no counterpart for r_D term on the third line.

As in the original ROMS code, the new algorithm has simultaneous conservation and constancy preservation properties for tracers, despite the fact that grid box heights change due to changing free surface, $H^{n+1}_k \neq H^n_k$. This can be verified by substituting $q \equiv 1$ into (3.4)–(3.6), including the rhs'_k terms which contain both the horizontal advection fluxes and the $W^{(e)}$-part of vertical, then noting that $W^{(e)}$ and $W^{(i)}$ add up into W, and, finally, verifying that after this substitute the above becomes equivalent to continuity equation (1.17) from SM2005, hence all the conservation properties of mode splitting and coupling algorithm developed there are still fully respected.

The splitting in (3.2) works as follows:

At first, vertical velocity fluxes $W_{i,j,k+1/2}$ are computed the standard way [*cf.,* SM2005, Eqs. (1.18)-(1.19)]. Our next goal is to split them into two parts, $W^{(e)}$ and $W^{(i)}$, in such a way that the explicit part $W^{(e)}$ in combination with the horizontal velocity fluxes are guaranteed to stay within the limits of numerical stability of explicit 3D-algorithm. The "excess" vertical flux $W^{(i)} = W - W^{(e)}$ is excluded from the stability budget because of the implicit treatment (3.4)-(3.6). The *a priori* stability criterion of an advective algorithm depends on specific numerical detail of time stepping, spatial interpolation, nonuniformity of computational grid, and therefore is difficult to define universally. We therefore chose an approximate path. A relevant finite-volume Courant number $\alpha_{i,j,k}$ defined as the sum of fluxes outgoing from the grid box, normalized by the time step size Δt and grid-box volume $\Delta\mathscr{V}_{i,j,k}$,

$$
\alpha_{i,j,k} = \frac{\Delta t}{\Delta\mathscr{V}_{i,j,k}} \cdot \Big[\ \max(U_{i+1/2,j,k}, 0) - \min(U_{i-1/2,j,k}, 0)
$$
$$
+\max(V_{i,j+1/2,k}, 0) - \min(V_{i,j-1/2,k}, 0) \tag{3.7}
$$
$$
+\max(W_{i,j,k+1/2}, 0) - \min(W_{i,j,k-1/2}, 0)\Big]\ ,
$$

where the uppercase $U_{i+1/2,j,k}$, $V_{i,j+1/2,k}$, and $W_{i,j,k+1/2}$ denote finite-volume fluxes (velocity components multiplied by the cross-sections of contact surfaces of grid boxes adjacent in the respective directions). The above defined $\alpha_{i,j,k}$ can be interpreted as the fraction of fluid within the grid-box which is replaced during one time step. $\alpha_{i,j,k} \geq 1$ corresponds to the situation when forward-in-time, first-order upstream scheme looses its positive-definiteness property and becomes numerically unstable.

For any other numerical scheme one must define the maximum allowed value $\alpha_{\max}$ not necessarily equal to 1. Then to guarantee that $\alpha_{i,j,k} \leq \alpha_{\max}$, with $\alpha_{i,j,k} = \alpha\left(U, V, W^{(e)}\right)$ defined above, it is sufficient to restrict

$$W^{(e)}_{i,j,k+1/2} = \min\left\{ \frac{\alpha_{\max} - \epsilon\alpha_{\perp i,j,k}}{\Delta t} \Delta\mathscr{V}_{i,j,k} \, , \max\left[W_{i,j,k+1/2} \, , -\frac{\alpha_{\max} - \epsilon\alpha_{\perp i,j,k+1}}{\Delta t} \Delta\mathscr{V}_{i,j,k+1} \right] \right\} \quad (3.8)$$

where

$$\alpha_{\perp i,j,k} = \frac{\Delta t}{\Delta\mathscr{V}_{i,j,k}} \cdot \Big[\max(U_{i+1/2,j,k}, 0) - \min(U_{i-1/2,j,k}, 0)$$
$$+ \max(V_{i,j+1/2,k}, 0) - \min(V_{i,j-1/2,k}, 0) \Big] \, , \quad (3.9)$$

is the contribution of horizontal fluxes into $\alpha_{i,j,k}$ and we have introduced $\epsilon \sim 1$ which is a weighting coefficient to take into account the difference in stability limits of horizontal and vertical advection schemes considered separately (one-dimensionally). In the same numerical approximations are used in all direction then $\epsilon = 1$. If, for example, horizontal advection is 4th-order centered difference and vertical is 4th-order compact, then $\epsilon = \left[\sqrt{\sqrt{6} - 3/2} \cdot \left(1 + 1/\sqrt{6}\right)\right]/\sqrt{3} \approx 1.37222/1.73205 = 0.79225$ which is the ratio of the maxima of dispersive curves for 4th-order centered and compact schemes in Fig. 1. If horizontal advection uses 4rd-order centered scheme during predictor stage of LF-AM3 step and 3rd-order upstream-biased during corrector (hence no-longer pure imaginary Fourier image), while vertical is 4th-order compact during both, then ϵ is the ratio of the actual stability limits, $\epsilon \approx 0.915/1.003 \approx 0.91$, (*cf.,* Fig. 24, middle row, vs. Figs. 2 and 5, left) which takes into account not only spatial discretizations, but also their combined properties with the specific time stepping algorithm. In practice, optimal selection of ϵ may be adjusted from the ratio of formal stability limits because some explicit schemes may exhibit undesirable behavior when used close to their limits (*e.g.,* lack of dissipation combined with large phase errors) which needs to be avoided.

In (3.8) it assumed that $\epsilon\alpha_{\perp i,j,k} < \alpha_{\max}$, so both expressions $\alpha_{\max} - \epsilon\alpha_{\perp i,j,k}$ and $\alpha_{\max} - \epsilon\alpha_{\perp i,j,k+1}$ in (3.8) are positive. Thus, (3.8) is simply to check whether the value of $W_{i,j,k+1/2}$ is within the two bounds (one is always positive, the other is always negative) and limit it if otherwise. Once $W^{(e)}_{k+1/2}$ is known, the implicit is always computed as the remainder,

$$W^{(i)}_{i,j,k+1/2} = W_{i,j,k+1/2} - W^{(e)}_{i,j,k+1/2} \, . \quad (3.10)$$

Eq. (3.8) can be rewritten as

$$W^{(e)}_{i,j,k+1/2} = \frac{W_{i,j,k+1/2}}{f\left(\alpha^*_w, \alpha^*_{\max}\right)} \quad \text{where} \quad \begin{cases} \left.\begin{array}{l} \alpha^*_w = \Delta t \cdot W_{i,j,k+1/2}/\Delta\mathscr{V}_{i,j,k} \\[6pt] \alpha^*_{\max} = \alpha_{\max} - \epsilon\alpha_{\perp i,j,k} \end{array}\right\} \text{ if } W_{i,j,k+1/2} > 0 \\[18pt] \left.\begin{array}{l} \alpha^*_w = \Delta t \cdot \left|W_{i,j,k+1/2}\right|/\Delta\mathscr{V}_{i,j,k+1} \\[6pt] \alpha^*_{\max} = \alpha_{\max} - \epsilon\alpha_{\perp i,j,k+1} \end{array}\right\} \text{ if } W_{i,j,k+1/2} < 0 \end{cases} \quad (3.11)$$

and the limiting function

$$f(\alpha, \alpha_{\max}) = \begin{cases} 1 \, , & \text{if } \alpha \leq \alpha_{\max} \\[6pt] \alpha/\alpha_{\max} \, , & \text{if } \alpha > \alpha_{\max} \end{cases} \quad (3.12)$$

20

which is continuous with respect to α crossing the threshold value $\alpha_{\max}$, but is not differentiable there. It is advantageous for the reasons evident from the analysis below to modify (in fact, generalize) it as

$$f(\alpha, \alpha_{\max}) = \begin{cases} 1\,, & \text{if}\ \ \alpha \leq \alpha_{\min} \\[2mm] 1 + \dfrac{(\alpha - \alpha_{\min})^2}{4\alpha_{\max}\,(\alpha_{\max} - \alpha_{\min})}\,, & \text{if}\ \ \alpha_{\min} < \alpha < 2\alpha_{\max} - \alpha_{\min} \\[2mm] \alpha/\alpha_{\max}\,, & \text{if}\ \ \alpha \geq 2\alpha_{\max} - \alpha_{\min} \end{cases} \qquad (3.13)$$

which above consists of three segments – constant, parabolic, and linear – smoothly matched to each other (Fig. 9). The two values, $\alpha_{\min}$ and $\alpha_{\max}$, control the threshold below which the algorithm is

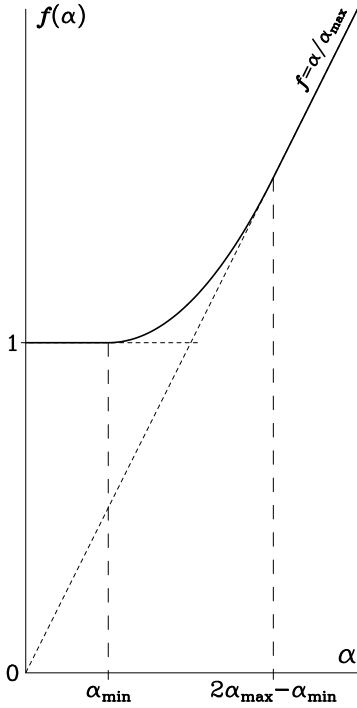Fig. 9. Limiting function for splitting vertical velocity into explicit and implicit parts.

fully explicit as is the original ROMS code, and the maximum allowed time step for the explicit part. The newly introduced $\alpha_{\min}$ is chosen to be a fixed fraction of $\alpha_{\max}$ – noted that application of limiting function via (3.11) implies that $\alpha_{\max}$ received by $f(\alpha, \alpha_{\max})$ as its second argument is adjusted (reduced) by the contribution of horizontal fluxes. In practical algorithm we define two parameters (fixed values) which set the initial values for $\alpha_{\min}$ and $\alpha_{\max}$. Then, for the purpose of computing (3.11) $\alpha_{\max}^* = \alpha_{\max} - \epsilon\alpha_{\perp}$ taken from the grid box above or below depending in the sign of vertical velocity, and $\alpha_{\min}^* = \alpha_{\min} \cdot \alpha_{\max}^*/\alpha_{\max}$ are used for computing limiting function $f(.)$. Selection $\alpha_{\min}$ and $\alpha_{\max}$, of their values is based on analysis of numerical stability and accuracy of the explicit part of the algorithm and will be detailed below.

The motivation for using upstream discretization (3.3) for the implicit part comes from the fact that it is monotonic and leads to a well posed, diagonally dominant discrete system. This choice is further 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. On the other hand, it is evident that a centered-in-space version

21

508   of (3.4)-(3.6) is obtained by replacing

$$509 \qquad \max\left(W^{(i)}_{k+1/2}, 0\right) q^{n+1}_k + \min\left(W^{(i)}_{k+1/2}, 0\right) q^{n+1}_{k+1} \quad \rightarrow \quad W^{(i)}_{k+1/2} \cdot \frac{q^{n+1}_k + q^{n+1}_{k+1}}{2} \qquad (3.14)$$

510   everywhere where $\max$ and $\min$ occur. Because of predominantly backward-Euler time stepping for
511   Courant numbers exceeding threshold value $\alpha_{\max}$ it is also dissipative, thought to a lesser degree than
512   the upstream under the same conditions. However, stability analysis (next section) leads to different
513   settings of threshold values $\alpha_{\min}$, $\alpha_{\max}$ for centered vs. upstream versions of the implicit part to ensure
514   the unconditional stability (transition to implicitness must occur earlier and at a heavier weight in the
515   centered case) resulting in virtually identical overall dissipation and phase errors for both adaptive
516   algorithms, despite the fact that the centered scheme is second-order accurate in space, while upstream
517   is only first. This further negates a motivation to abandon diagonal dominance of algorithm described
518   above.

519   Selecting α^* as the upstream value between α_k and α_{k+1} in (3.11) in combination with computing
520   Courant number from outgoing fluxes via (3.7) yields α^* dominated by the contribution from $W_{k+1/2}$
521   itself in the case where vertical Courant number becomes large and the adaptive limiting is activated,
522   i.e., where it matters. This applies to both upward and downward $W_{k+1/2}$, while the only difference
523   that contribution from the horizontal fluxes is taken from either the grid box below or above the in-
524   terface $k + 1/2$. Overall, the algorithm is designed to avoid spatial averaging in any direction when
525   computing Courant number: this is motivated by the desire to avoid an underestimate in situation when
526   velocity field is not smooth on grid-scale (*e.g.,* if alternatively to (3.7) one can average all three veloc-
527   ity components toward the center of grid box $H_{i,j,k}$ and compute α using these values. The outcome
528   may be substantially different when the velocity field is a checker-board mode in all three directions
529   as pair-wise averaging of opposite sign values leads to a smaller computed Courant number, delay in
530   triggering adaptive algorithm, and failure to prevent numerical instability). Alternatively to using the
531   upstream value for α^* in (3.11), a more restrictive option $\alpha^* = \max\{\alpha_k, \alpha_{k+1}\}$ was tried as well. In
532   practice neither option have shown any advantage over the other, nor yielded a noticeable difference
533   in the results.

534   *3.2   Fourier analysis, stability, and parameter optimization*

535   In the case of uniform grid spacing, $\Delta x = const$, and in the absence of physically-motivated diffu-
536   sion, $A = 0$, the combination of LF-AM3 time-stepping (2.6)-(2.7) with adaptively-implicit advection
537   algorithm (3.5) becomes [6] :

$$538 \qquad q^{n+1/2}_j = \left(\frac{1}{2} - 2\gamma\right) q^{n-1}_j + \left(\frac{1}{2} + 2\gamma\right) q^n_j - (1 - 2\gamma)\left[\; \alpha'\left(\tilde{q}^n_{j+1/2} - \tilde{q}^n_{j-1/2}\right) \right.$$
$$\left. + \alpha''\left(q^{n+1/2}_j - q^{n+1/2}_{j-1}\right)\right] \qquad (3.15)$$

539   followed by

$$540 \qquad q^{n+1}_j = q^n_j - \alpha'\left(\tilde{q}^{n+1/2}_{j+1/2} - \tilde{q}^{n+1/2}_{j-1/2}\right) - \alpha''\left(q^{n+1}_j - q^{n+1}_{j-1}\right) \qquad (3.16)$$

---

[6]   For simplicity we limit ourselves to the analysis to the one-dimension case, as all the motion takes place in
vertical direction only. Generalization to multiple dimensions – essentially having horizontal advection using an
explicit algorithms in addition to vertical is considered in Appendix C.

where

$$\alpha = \frac{c\Delta t}{\Delta x}\,, \qquad\qquad \alpha' = \frac{\alpha}{f(\alpha)}\,, \qquad\qquad \alpha'' = \alpha - \alpha' \qquad\qquad (3.17)$$

are Courant number, its explicit portion limited by (3.13), and the remainder to be treated implicitly. $\gamma = 1/12$ is needed to achieve the third-order temporal accuracy; alternative setting $\gamma = 0$ corresponds to LF-TR step which is second-order.

Inserting Fourier component $q_j^n = \lambda^n \cdot \widehat{q}_k \cdot e^{ik\Delta xj}$ into above and replacing of spatial differences with their Fourier images,

$$\tilde{q}_{j+1/2} - \tilde{q}_{j-1/2} = i\mathscr{K}\Delta x \cdot \widehat{q}_k \cdot e^{ik\Delta xj}\,, \qquad\qquad q_j - q_{j-1} = \left(1 - e^{-ik\Delta x}\right)\cdot \widehat{q}_k \cdot e^{ik\Delta xj}\,, \qquad (3.18)$$

where $i\mathscr{K}\Delta x$ is the same as in (2.12), yields the characteristic equation

$$\begin{aligned}
\left[1 + \alpha''\left(1 - e^{-ik\Delta x}\right)\right]\lambda = 1 &- i\alpha'\mathscr{K}\Delta x \cdot \frac{\left(\frac{1}{2} + 2\gamma\right) - i\alpha'\mathscr{K}\Delta x\,(1 - 2\gamma)}{1 + \alpha''\left(1 - e^{-ik\Delta x}\right)(1 - 2\gamma)} \\
&- i\alpha'\mathscr{K}\Delta x \cdot \frac{\left(\frac{1}{2} - 2\gamma\right)}{1 + \alpha''\left(1 - e^{-ik\Delta x}\right)(1 - 2\gamma)}\cdot \lambda^{-1}\,.
\end{aligned} \qquad (3.19)$$

It is easy to verify that the above reverts back to (2.14) if $\alpha' \to \alpha$ and $\alpha'' \to 0$. In the opposite limit, setting $\alpha' \to 0$ and $\alpha'' \to \alpha$ yields only one root, $\lambda = \left[1 + \alpha''\left(1 - e^{-ik\Delta x}\right)\right]^{-1}$ as the equation degenerates from the second- to the first-order. This λ corresponds to backward-Euler-in-time, upstream-in-space scheme, which is $|\lambda| \leq 1$ for the entire range of values, $0 \leq \alpha < +\infty$, monotonic, but only first-order in time and in space. Since $0 \leq k\Delta x < -\pi$, therefore $e^{-ik\Delta x}$ assumes complex values on the lower half of unit circle (has negative imaginary part), no Fourier component is allowed to have phase increment per time step Δt to exceed $\pi/4$, which means phase delay for all Fourier components. In the general case when both $\alpha' \neq 0$ and $\alpha'' \neq 0$, the values of λ become some blending (generally speaking nonlinear) of these two extreme cases. Below in this part we show that with an appropriate choice of limiting function $f = f(\alpha)$ the algorithm can be made unconditionally stable, but at the same time retain good accuracy for the Fourier components which frequencies are resolved in time.

Intuitively the upper threshold $\alpha_{\max}$ has the meaning of "never exceed speed" for the explicit component, hence should be chosen not larger than its limit of stability, *e.g.,* $\alpha = 0.915$ in the case of LF-AM3 stepping combined with 4th-order compact spatial discretization. Also, because the implicit component of the adaptive algorithm is expected to be less accurate than the explicit, both $\alpha_{\min}$ and $\alpha_{\max}$ should be chosen as large as possible to minimize numerical errors. This leads to a natural first guess of $\alpha_{\min} = \alpha_{\max} = 0.915$. The resultant $|\lambda|^{1/\alpha}$ and $c^*/c^{(\text{exact})}$ are shown in Fig. 10. The algorithm is unconditionally stable, but it has evident drawbacks: both dissipation and phase speed are changing abruptly once α exceeds the threshold value; there is also sharp raise toward $|\lambda| = 1$ in the vicinity of $k\Delta x \approx 2\pi/3$, $\alpha = 0.915$, which may lead to undamped and unresolved modes resulting in numerical artifacts. In part this sharp transition is due to the fact that choosing $\alpha_{\min} = \alpha_{\max}$ causes (3.13) switch sharply between the constant and linear segments. The other contributor to the non-smoothness is the structure of physical mode of the explicit version of LF-AM3 itself: at first the physical mode follows the units circle while gradually departing inward, then when approaching the stability limit it bends outward and crosses the unit circle at significant angle. The sharp raise in toward 1 in $|\lambda|^{1/\alpha}$ panel Fig. 10 is traced back to this property of LF-AM3. Inserting the parabolic segment into (3.13) (hence
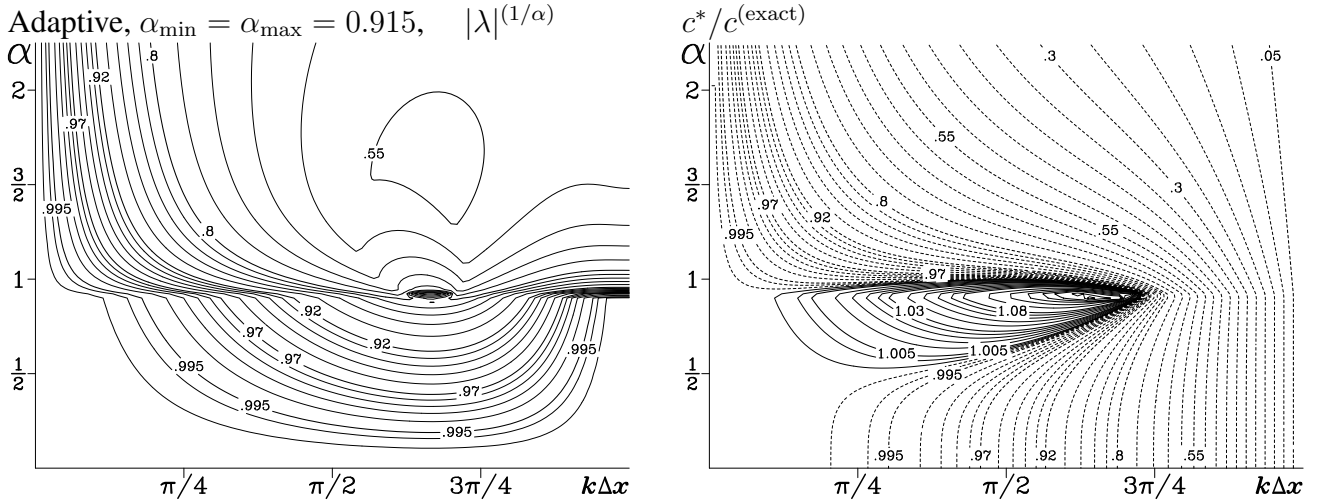
23

Fig. 10. Amplitude multiplier per $1\Delta x$ travel, $|\lambda|^{(1/\alpha)}$ (*left*), and numerical-to-ideal phase-speed ratio, $c^*/c^{(\text{exact})}$ (*right*) adaptive advection scheme with $\alpha_{\min} = \alpha_{\max} = 0.915$.

choosing $\alpha_{\min} < \alpha_{\max}$) should make smooth transition in the properties of the adaptive algorithm, but, as we will see soon, also helps to modify the behavior of the algorithm in the vicinity of Courant numbers close to stability limit of the explicit version of LF-AM3.

Another complication comes from the fact that both fully-explicit and adaptively-implicit versions of LF-AM3 time stepping have one physical and one computational mode. In the explicit case the computational mode stays well inside the unit circle for the entire range of stability (SM2005, see Fig. 20, third diagram on the bottom row). Therefore the computational mode is not a concern in Fig. 2, where both $|\lambda|^{(1/\alpha)}$ and $c^*/c^{(\text{exact})}$ belong to the physical mode. It also happens that the computational mode stays well within $|\lambda| < 1$ for the adaptively-implicit algorithm with parameter choice shown in Fig. 10. However, we will see soon, this is not always the case for the adaptive algorithm, and the computational mode may have larger amplitude of $|\lambda|$ than physical for some combinations of $k\Delta x$ and α, thus both modes require consideration, and, because the behavior of computational mode is less predictable than of physical, one needs to perform a full survey of parameter space rather than rely on intuitive settings.

To search for suitable settings $\alpha_{\min}$ and $\alpha_{\max}$ we note that stability of the explicit version depends entirely on the behavior of its characteristic equation (2.14) along the line $k\Delta x = 2\pi/3$ on the $(k\Delta x, \alpha)$-plane corresponding to the maximum possible value of $\mathcal{K}\Delta x = \mathcal{K}\Delta x(k\Delta x)$ for all $k\Delta x$ – the highest point on bold curve in Fig. 1, resulting in $K\Delta x = \sqrt{3}$. Therefore, it is useful to study behavior of the characteristic equation of the adaptive algorithm (3.19) along the same line. This is presented in Fig. 11. At first, we remind ourselves the properties of explicit LF-AM3 algorithm, *top-left*. It is identical to the third panel in bottom row in Fig. 20 from SM2005, with the exception that the original meaning of polar angle was $\omega\Delta t$ (time step normalized by frequency of oscillation), and now it is $\varphi = \alpha \cdot \mathcal{K}\Delta x$. LF-AM3 becomes unstable when its physical mode leaves the unit circle at $\varphi \approx 1.58745$, which corresponds to $\alpha = 0.915$. The computational mode for this algorithm stays well inside the unit circle and does not affect stability. Then, starting with $\alpha_{\min} = 0.915$ and decreasing it toward 0, we find the maximum possible values of $\alpha_{\max} = \alpha_{\max}(\alpha_{\min})$ for which the characteristic roots of (3.19) stay within the unit circle. In practice (except for $\alpha_{\min} = 0.915$) this means that either the physical or the computational mode approaches and touches the unit circle and turns inward (a small increase in $\alpha_{\max}$ relative to its stated value would cause exit and re-entry of the unit circle in
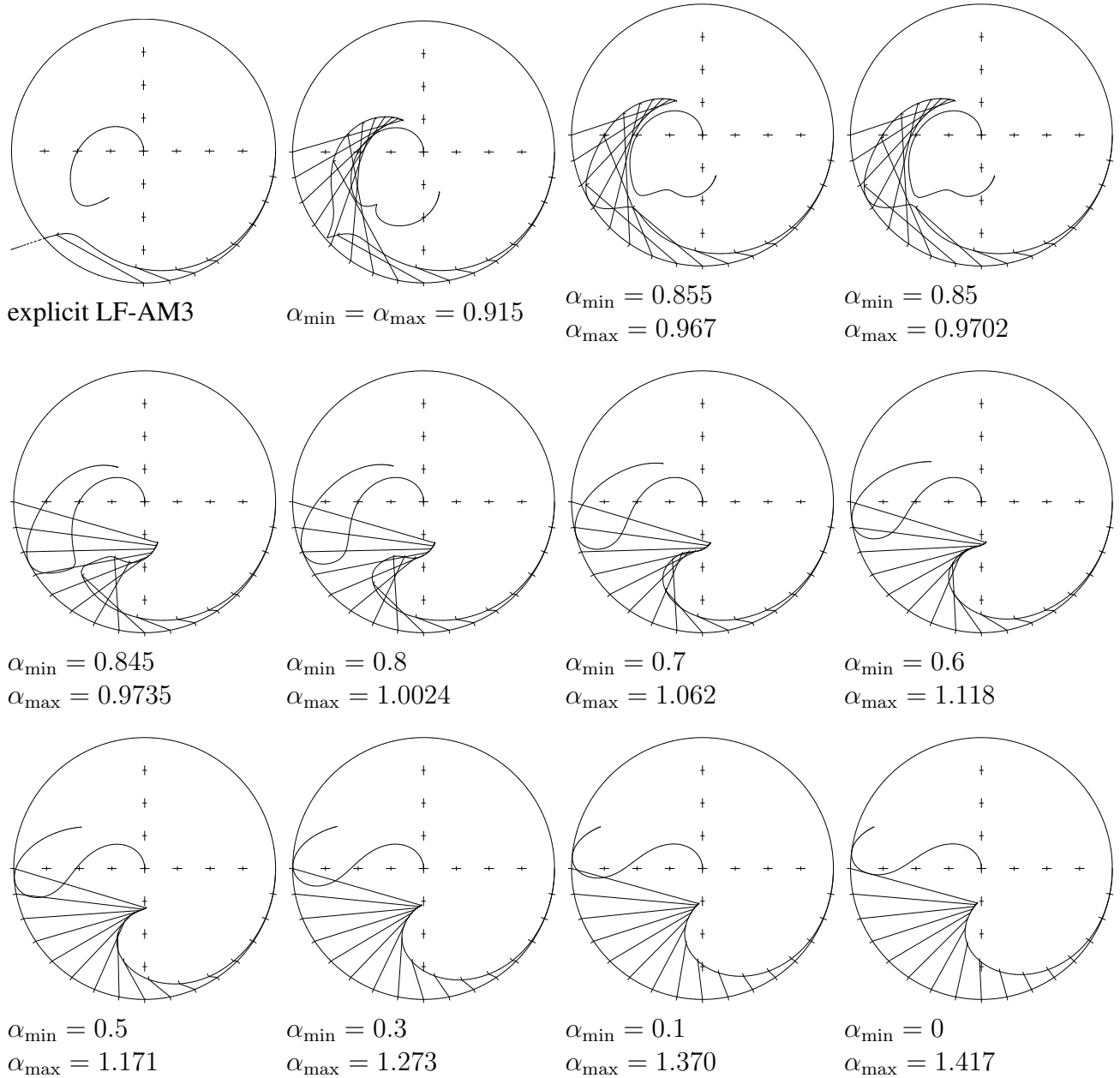
explicit LF-AM3

$\alpha_{\min} = \alpha_{\max} = 0.915$

$\alpha_{\min} = 0.855$
$\alpha_{\max} = 0.967$

$\alpha_{\min} = 0.85$
$\alpha_{\max} = 0.9702$

$\alpha_{\min} = 0.845$
$\alpha_{\max} = 0.9735$

$\alpha_{\min} = 0.8$
$\alpha_{\max} = 1.0024$

$\alpha_{\min} = 0.7$
$\alpha_{\max} = 1.062$

$\alpha_{\min} = 0.6$
$\alpha_{\max} = 1.118$

$\alpha_{\min} = 0.5$
$\alpha_{\max} = 1.171$

$\alpha_{\min} = 0.3$
$\alpha_{\max} = 1.273$

$\alpha_{\min} = 0.1$
$\alpha_{\max} = 1.370$

$\alpha_{\min} = 0$
$\alpha_{\max} = 1.417$

Fig. 11. *Top left*: characteristic roots λ plotted against unit-circle on complex-plane for explicit LF-AM3 step-ping. The argument φ (clockwise angle from positive direction of x-axis) corresponds to $\alpha \cdot \sqrt{3}$ where α is Courant number, and multiplier $\sqrt{3}$ corresponds to the highest point on bold $\mathscr{K}\Delta x = \mathscr{K}\Delta x(k\Delta x)$ curve in Fig. 1. Bold line corresponds to physical mode, thin line to computational. Outer notches on the unit circle indicate the locations of "ideal" roots; notches on the inner side of bold line indicate the actual roots corresponding to each ideal. Thin straight lines connecting the two roots illustrate the numerical error: displacement along the unit circle manifests numerical dispersion, moving inward – dissipation. [It should be noted that *ideal* in this context means ideal from the point of view of *time stepping analysis only*, since the exact phase multiplier for $k\Delta x = 2\pi/3$ mode should be $e^{i\alpha \cdot 2\pi/3}$ in the case of ideal spatial differencing rather than $e^{i\alpha \cdot \sqrt{3}}$ used as the basis for comparison here.] *All other panels*: for each $\alpha_{\min} = 0.915, 0.855, ... 0.1, 0$ we find the maximum possible $\alpha_{\max}$ for which both the physical and the computational characteristic roots λ of the adaptive algorithm (3.19) of stay entirely within the unit circle.

each case). The entire sequence shown in Fig. 11 exhibits only one qualitative change in behavior at $\alpha_{\min} = 0.85$ before which it is the physical mode approaches the unit circle, while the computational stays inside. Beyond 0.85 they switch roles, and all other diagrams look alike, and differ mainly by the level physical mode dissipation. This sequence indicates that selecting $\alpha_{\min}$ smaller than the stability limit of the explicit scheme, 0.915, also allows to select $\alpha_{\max}$ beyond it, which is, in fact, contrary to the "never exceed speed" rationale. The explanation comes from the fact that once Courant number exceeds $\alpha = \alpha_{\min}$, the explicit scheme is blended with a highly dissipative implicit which introduces numerical damping of the high wavenumber Fourier components – those which would be unstable first in the case of fully explicit algorithm. This applies to both physical and computational modes.
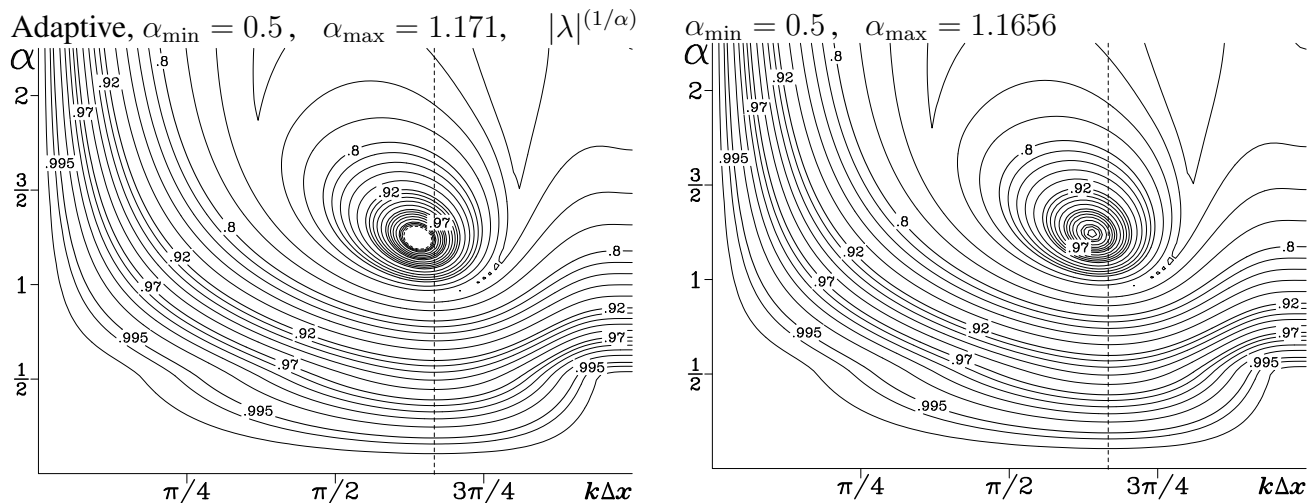

Fig. 12. *Left*: $|\lambda|^{(1/\alpha)}$ for an adaptive scheme with arbitrarily picked value of $\alpha_{\min}$ (0.5 in this case) and maximum possible $\alpha_{\max} = 1.171$ as predicted by the 1D-analysis along $k\Delta x = 2\pi/3$ line (indicated on this plot as vertical dashed line). Note that $|\lambda|^{(1/\alpha)}$ approaches 1 from below when moving along $k\Delta x = 2\pi/3$ line at $\alpha \approx 1.2$ (consistently with the analysis in Fig. 11), however the actual two-dimensional maximum of $|\lambda|^{(1/\alpha)}$ lies slightly on the left from the line, so the values inside the dashed contour exceed 1, rendering this algorithm be numerically unstable within this "hole" on $(k\Delta x, \alpha)$-plane. *Right*: same, but with $\alpha_{\max}$ adjusted to lower down the value of 2D maximum of $|\lambda|^{(1/\alpha)}$ to not exceed 1, hence eliminating the hole.

While the 1D analysis in Fig. 11 exposes mutual influences of $\alpha_{\min}$ vs. $\alpha_{\max}$ settings on numerical stability, its outcome is not automatically applicable to the entire $(k\Delta x, \alpha)$-plane because algorithm (3.15)-(3.16) blends two spatial discretizations with qualitatively different Fourier images of the first spatial derivative (*cf.,* $i\mathscr{K}\Delta x$ for centered compact fourth-order differencing vs. $(1 - e^{-ik\Delta x})$ for the upstream). This means that stability analysis of (3.19) cannot be, in principle, reduced to separate analysis of spatial and temporal schemes as it is possible in the case of (2.14) where α and $i\mathscr{K}\Delta x$ appear exclusively as the product of $i\alpha\mathscr{K}\Delta x$ and never separately [7], hence the unconditional stability along the $k\Delta x = 2\pi/3$-line presented in Fig. 11 does not guarantee the unconditional stability for the entire two-dimensional $(k\Delta x, \alpha)$-plane. This is illustrated in Fig. 12. The maximum values of $\alpha_{\max} = \alpha_{\max}(\alpha_{\min})$ from Fig. 11 are therefore need to be adjusted for each $\alpha_{\min}$ using full 2D analysis. They become

---

[7] Strictly speaking, for this reason alone the adaptive algorithm can no longer be classified as a *method of lines*, even thought the underlying explicit algorithm is.
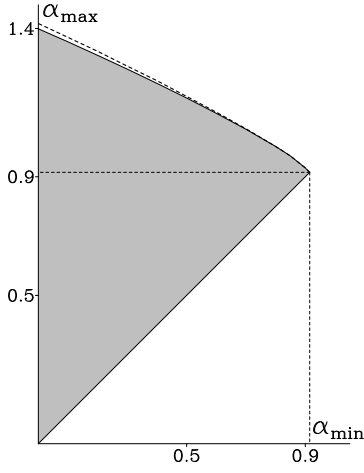
Fig. 13. Region of unconditional stability of algorithm (3.15)-(3.16) shown as shaded area on $(\alpha_{\min}, \alpha_{\max})$-plane. Dashed line on the top corresponds to the sequence of the maximum possible $\alpha_{\max} = \alpha_{\max}(\alpha_{\min})$ settings resulting from the 1D analysis along $k\Delta x = 2\pi/3$ line from Fig. 11. Note the gap between the shaded area and the line.

$\alpha_{\min}$	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	.85	.9	.915
$\alpha^*_{\max}$	1.417	1.37	1.323	1.273	1.224	1.171	1.118	1.062	1.0024	.9702	.93	.915
$\alpha^{**}_{\max}$	1.399	1.355	1.3095	1.263	1.215	1.1656	1.114	1.0603	1.0024	.9702	.93	.915

where $\alpha^*_{\max}$ are from Fig. 11 included for comparison, and $\alpha^{**}_{\max}$ are the final values. The fact that the two sets of values become indistinguishable as $\alpha_{\min}$ approaches 0.915 is explained by the fact the location of instability hole is centered on the $k\Delta x = 2\pi/3$-line for $\alpha_{\min} = 0.915$, and remains nearly centered (gradually moves to the left) when $\alpha_{\min}$ departs from 0.915, which means that the analysis in Fig. 11 is still very accurate there. The qualitative behavior of physical and computational mode characteristic roots λ shown in Fig. 11 remains the same if a similar analysis is performed not for $k\Delta x = 2\pi/3$, but for $k\Delta x = const$-lines crossing the actual 2D-maximum of $|\lambda|^{(1/\alpha)}$ in each $\alpha_{\min}, \alpha^{**}_{\max}$ setting from the table above. Therefore the table defines the envelope of unconditional region in $(\alpha_{\min}, \alpha_{\max})$-parameter space as

$$0 \le \alpha_{\min} \le 0.915 \qquad\qquad \alpha_{\min} \le \alpha_{\max} \le \alpha^{**}_{\max}(\alpha_{\min}), \qquad\qquad (3.20)$$

which is illustrated in Fig. 13.

Besides stability, two other considerations are involved in choosing threshold values: (*i*) providing sufficient numerical damping to the 2D maximum of $|\lambda|^{(1/\alpha)}$ (which requires that $\alpha_{\max}$ is further reduced from the $\alpha^{**}_{\max}$ value in the table above, and (*ii*) continuity and smoothness of the properties of the numerical scheme as the actual Courant number crosses the threshold of implicit components (satisfying this needs to widen the gap between $\alpha_{\min}$ and $\alpha_{\max}$, which in its turn leads to selection of smaller values of $\alpha_{\min}$ less than 0.915 – the maximum allowed).

To ensure sufficient numerical damping – item (*i*) above, we note that the undesirable transient uplift of $|\lambda|$ toward 1 in the vicinity of $(k\Delta x = 2/\pi/3, \alpha = 0.915)$ on Fig. 10 is traced back to the behavior of physical mode in second panel on top row of Fig. 11, an can be eliminated completely by adjusting the values of $\alpha_{\min}, \alpha_{\max}$, as shown in Fig. 14. With such choice of parameters it is possible to encircle the computational mode by the physical, while at the same time to achieve monotonic decrease of amplitude of physical $|\lambda|$ with increasing Courant number α (thought $|\lambda|^{(1/\alpha)}$ still has some weak non-monotonic behavior as manifested by the closed contour .55), and also push back the phase delay of high-wavenumber modes as α increases. However, the latter applies only in the vicinity of $k\Delta x = 2\pi/3$, while there is still sharp transition in both dissipative and dispersive properties for

27

$\alpha_{\min} = 0.885$
$\alpha_{\max} = 0.900$

$\alpha_{\min} = 0.885$
$\alpha_{\max} = 0.905$

$\alpha_{\min} = 0.885$
$\alpha_{\max} = 0.910$

$\alpha_{\min} = 0.885$
$\alpha_{\max} = 0.915$

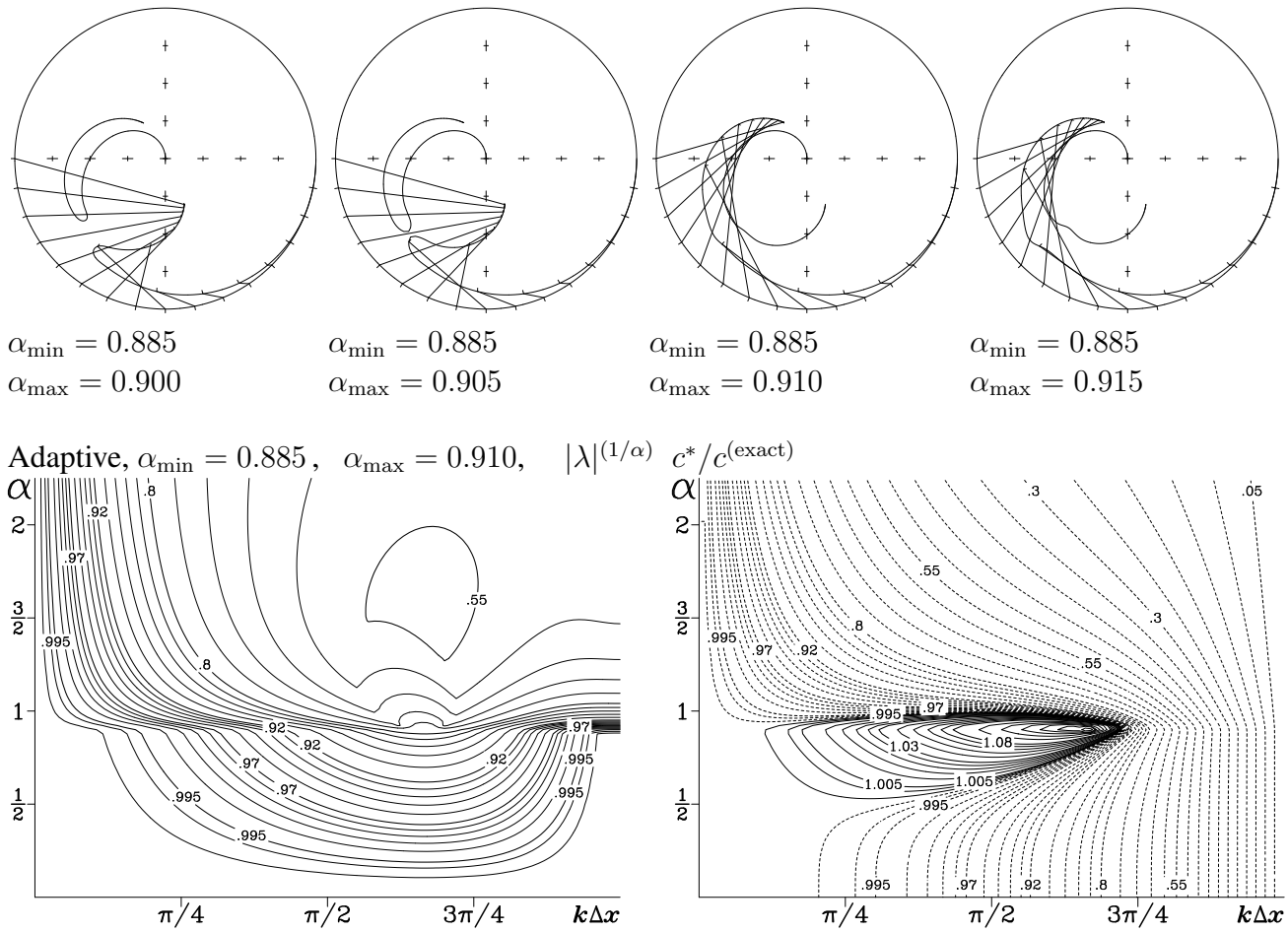Adaptive, $\alpha_{\min} = 0.885$, $\alpha_{\max} = 0.910$, $|\lambda|^{(1/\alpha)}$ $c^*/c^{(\text{exact})}$

Fig. 14. Addressing the non-smooth transition in dissipative properties of the $\alpha_{\min} = \alpha_{\max} = 0.915$ algorithm in Fig. 10: *Top row*: evolution of characteristic roots λ for physical and computational modes computed along $k\Delta x = 2\pi/3$ line (same as in Fig. 11) with changing $\alpha_{\max}$. Note that the two modes meet each other and reconnect when $\alpha_{\max} = 0.905 \rightarrow 0.910$ resulting in a scheme with computational mode amplitude $|\lambda|$ be consistency smaller than physical, and with monotonically increasing dissipation of physical mode as it spirals inward the unit circle (further increase in $\alpha_{\max}$ causes bulging of the physical mode toward the unit circle, making the overall pattern be similar to third panel in top row of Fig. 11). *Bottom*: $|\lambda|^{(1/\alpha)}$ and $c^*/c^{(\text{exact})}$ for $\alpha_{\min} = 0.885$ $\alpha_{\max} = 0.910$ in the same format as in Fig. 10.

resolved wavenumbers ($k\Delta x < \sim \pi/3$, , $\alpha \sim 0.9$). (Sharp transition for the highest wavenumbers, $k\Delta x > 3\pi/4$ is much less important because dispersive errors are very large there any way). To eliminate the sharp transition one needs to widen the gap between $\alpha_{\min}$ and $alpha_{\max}$. Unfortunately this is impossible to achieve while maintaining the qualitative pattern observed on the third circle diagram of Fig. 10 (keeping the computational mode inside the physical, and having monotonic dissipation of the latter), hence a compromise needs to be made.

In practice, given that the explicit LF-AM3 already start exhibiting noticeable dissipative behavior for $\alpha > 0.6$ (*cf.,* Figs. 4 and 5), the first threshold value $\alpha_{\min}$ should be chosen in this vicinity as well (as picking a higher value does not lead to a scheme with a significantly smaller dissipation for these Courant numbers. The second threshold, $alpha_{\max}$, is then selected to "sink" the 2D maximum of $|\lambda|^{(1/\alpha)}$ down from 1 to an acceptable value, so the scheme does not leave undamped Fourier components if the actual Courant number happens to be
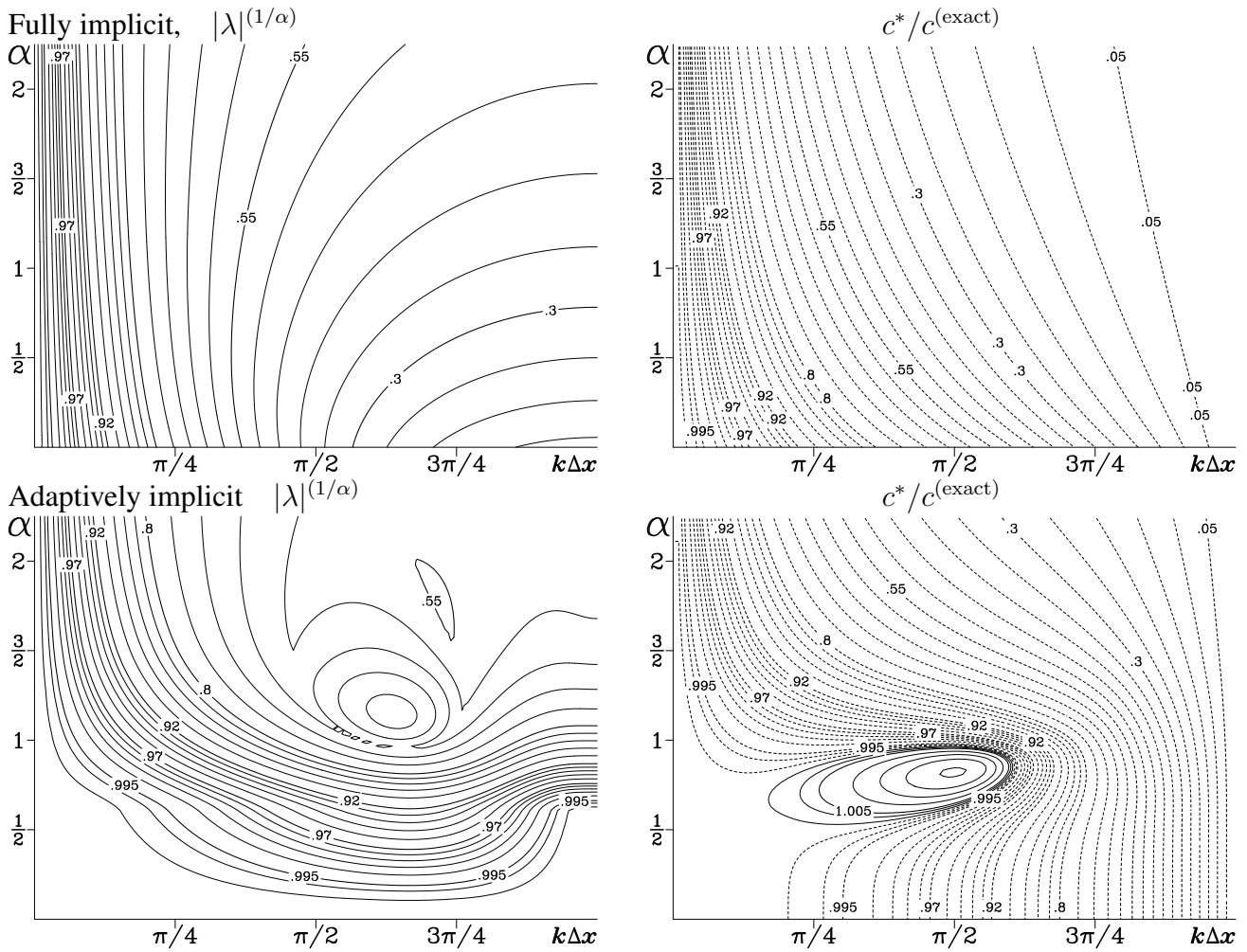
Fig. 15. Amplitude multiplier per $1\Delta x$ travel $|\lambda|^{(1/\alpha)}$ (*left column*) and the ratio of numerical to "ideal" phase speeds (*right column*) for fully-implicit (backward-Euler in time, upstream in space, *upper row*) and adaptively-implicit (using LF-AM3 stepping for the explicit part; *bottom row*) advection algorithm. The threshold values in the adaptive algorithm are $\alpha_{\min} = 0.6$, $\alpha_{\max} = 1.0$. Note that the $0 < \alpha < \alpha_{\min}$ portions of both lower panels are exactly the same as in the corresponding panels in Fig. 2. Because LF-AM3 algorithm have one physical and one computational mode, plotted on the $|\lambda|^{(1/\alpha)}$ panel above is the maximum of the two modes. However, $c^*/c^{(\text{exact})}$ is still for physical mode only. The eye-like pattern in $|\lambda|^{(1/\alpha)}$ panel centered around $(k\Delta x = 2\pi/3,\ \alpha \approx 1.2)$ with an apparent non-smooth transition from the rest of the plot is due to the fact that in the vicinity of this point the amplitude of computational mode exceeds that of the physical mode, thought both of them are well within the unit circle – the local maximum value there is $|\lambda|^{(1/\alpha)} \approx 0.72$.

666 close that of the maximum. An example of such setting is illustrated in Fig. 15, *bottom row*, where we
667 have chosen $\alpha_{\min} = 0.6,\ \alpha_{\max} = 1.0$. For comparison we also include similar panels for fully implicit
668 backward Euler, upstream-in-space advection, which is unconditionally stable, monotonic, but is only
669 first-order accurate in space and time, and, accordingly, is highly dissipative. In the asymptotic limit
670 of infinitely large Courant number, $\alpha \to \infty$, the properties of the adaptive and fully implicit schemes
671 become the same. On the other hand, for all regimes within $\alpha \le \alpha_{\min}$ the adaptive scheme is identical
672 to that on Fig. 2, left column. The transition zone begins at $\alpha = \alpha_{\min}$, and because LF-AM3 stepping
673 has tendency for phase acceleration toward the end of its stability limit, while implicit step causes
674 delay, a proper choice of $\alpha_{\min}, \alpha_{\max}$ can utilize compensation of these effects resulting in a wider
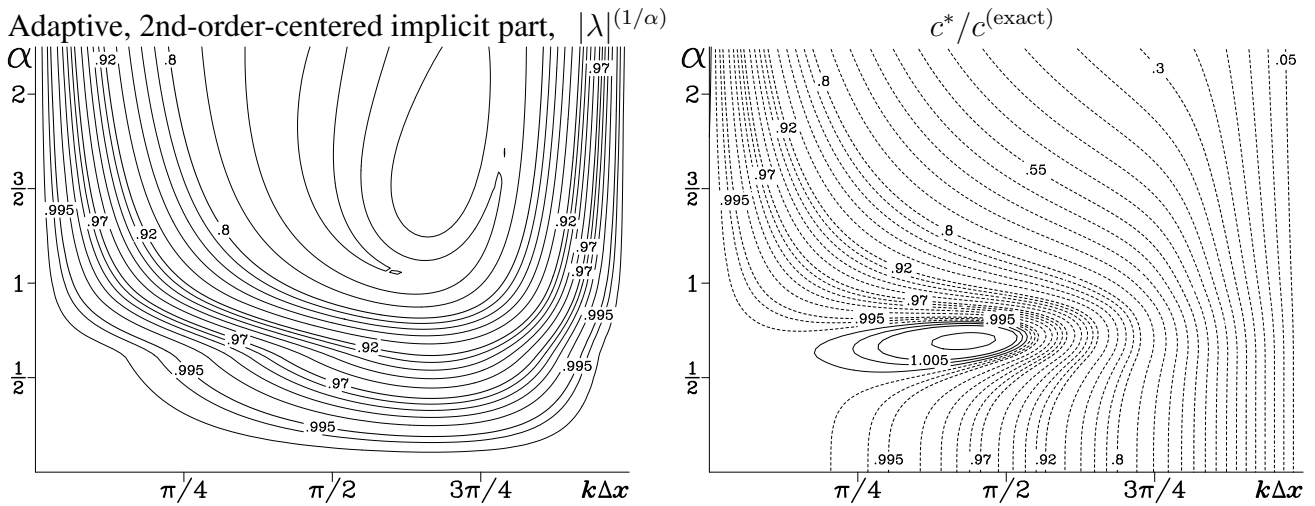
Fig. 16. Same as Fig. 15, but using second-order centered scheme for the implicit part instead of upstream, $\alpha_{\min} = 0.6$, and $\alpha_{\max} = 0.75$.

zone where the adaptive scheme is accurate in phase speed (*cf.,* Fig. 15 vs. Fig. 6). At last, it is worth to note that for the resolved wavenumbers, $k\Delta x <\sim \pi/3$, and after entering into implicit regime, $\sim 1 < \alpha <\sim 3/3$, both the dissipation and the phase delay in Fig. 15 are actually less than in Figs. 10 and 14 despite the fact that parameter choice in the latter two was motivated solely to delay the transition to implicitness as much as possible.

It is also useful to compare this with using a second-order centered scheme instead of upstream for the implicit part [*cf.,* Eq. (3.14) and the paragraph in which it occurred]. The centered-implicit scheme is naturally less dissipative, and produces smaller phase delay [*cf.,* Appendix B] than the upstream, however to achieve the same degree of damping of the local maximum associated with the computational mode of LF-AM3 stepping – keeping $|\lambda|^{(1/\alpha)}$ just below 0.8 there – one needs to choose $\alpha_{\min} = 0.6$ and $\alpha_{\max} = 0.75$, which leads a heavier weighing on implicitness under the same conditions. The properties of the resulting scheme are shown in Fig. 16. Judging by the dissipation and phase errors for the resolved wavenumbers, $k\Delta x < \pi/2$, and Courant numbers entering into implicit zone, $\alpha > 0.6$, the two algorithms remarkable similar. The most significant difference is that the centered scheme does not dissipate wavenumbers approaching $k\Delta x \to \pi$ while not propagation them correctly either. The upstream strongly dampens them once Courant number is large enough to activate implicitness. This property is desirable. This comparison indicates that selection of the upstream scheme for the implicit part motivated by the reasons stated above (monotonicity, diagonal dominance) also leads to no loss of accuracy for the resolved wavenumbers, despite having smaller order spatial discretization, first vs. second.

Fig. 17 shows the results for advection of narrow pulse using adaptive scheme for a wide range of Courant numbers. The original LF-AM3 algorithm is included for comparison on the left side (only for values of α where it is numerically stable), while fully implicit results are shown on the right. With our choice of parameters adaptive scheme is identical to LF-AM3 for $\alpha \leq 0.6$, while beyond that it becomes progressively more dissipative, and, as expected, for the largest values of α the results are similar to fully implicit. While the adaptive scheme does not have formally monotonicity property, it is sufficiently dissipative to avoid spurious oscillations when the implicit part takes over, while at the same time it retains the accuracy of the original explicit step for the range of Courant numbers which are resolved in time.

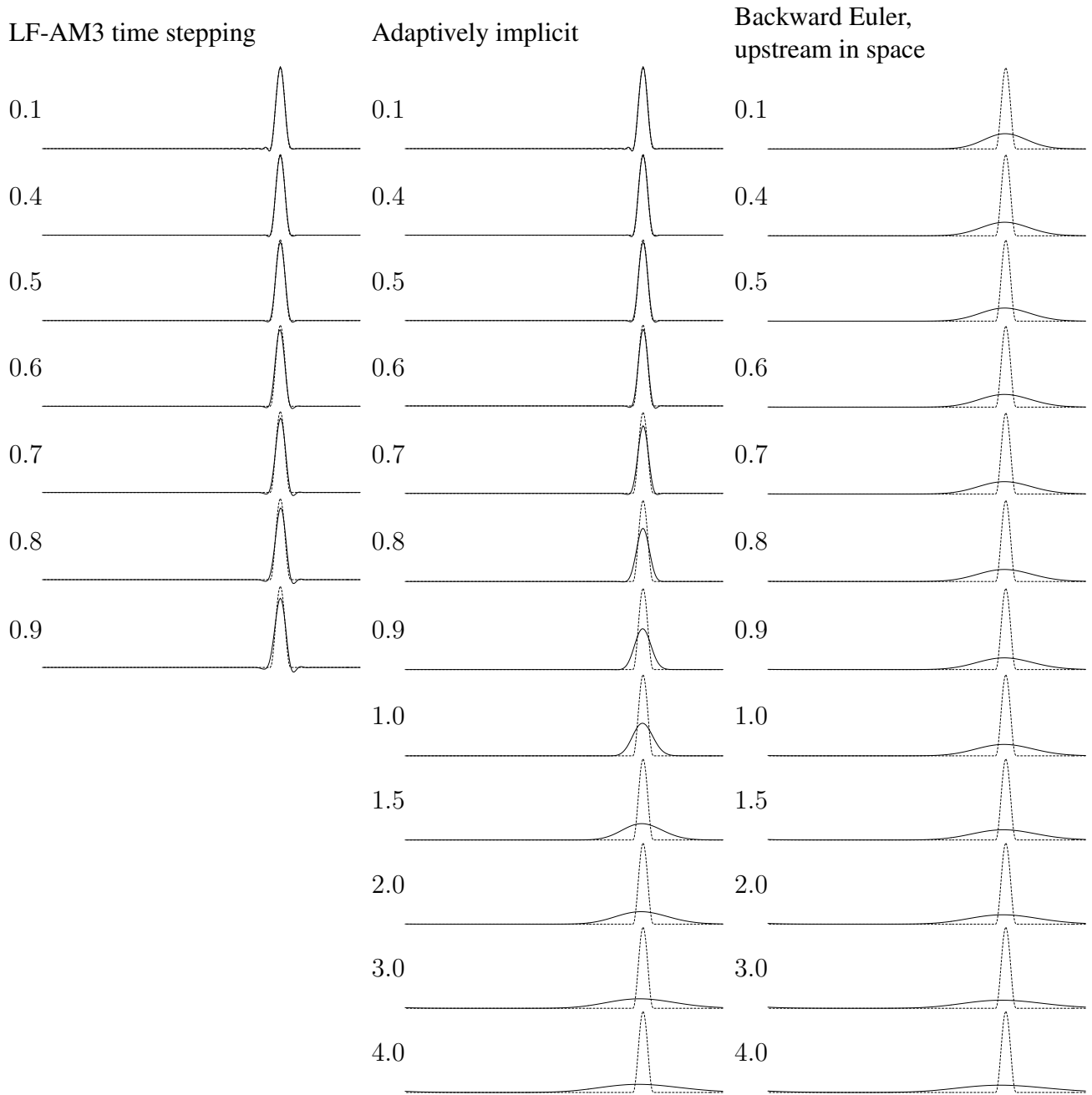LF-AM3 time stepping   Adaptively implicit   Backward Euler,
                                                           upstream in space

0.1   0.1   0.1

0.4   0.4   0.4

0.5   0.5   0.5

0.6   0.6   0.6

0.7   0.7   0.7

0.8   0.8   0.8

0.9   0.9   0.9

                              1.0   1.0

                              1.5   1.5

                              2.0   2.0

                              3.0   3.0

                              4.0   4.0

Fig. 17. Comparison of LF-AM3 algorithm (*left column*, same as on Fig. 5), adaptively implicit (*middle column*, threshold Courant numbers settings $\alpha_{\min} = 0.6$, $\alpha_{\max} = 1.0$, same as in Fig. 15), and fully-implicit backward Euler upstream in space advection (*right column*). All the conditions are the same as in Fig. 5. Number on the left of each panel indicates Courant number.

704 The adaptively implicit algorithm can be combined with semi-Lagrangian advection as well. As-
705 suming the same $\alpha \to \alpha' + \alpha''$ limiting and splitting (3.17), (3.13), the characteristic equation of the
706 adaptive version becomes

707
$$\left[1 + \alpha'' \left(1 - e^{ik\Delta x}\right)\right] \lambda = \Lambda\left(\alpha'\right) \qquad (3.21)$$

708 where $\Lambda\left(\alpha'\right)$ is the expression in the r.h.s. of (2.18) with α substituted by α'. The properties of (2.18)
709 are shown in Fig. 18 for $\alpha_{\min} = 0.6$, $\alpha_{\max} = 1.0$. There is no computational mode, and, in fact, this
710 choice of $\alpha_{\min}$, $\alpha_{\max}$ is not optimal considering semi-Lagrangian advection alone: both parameters can
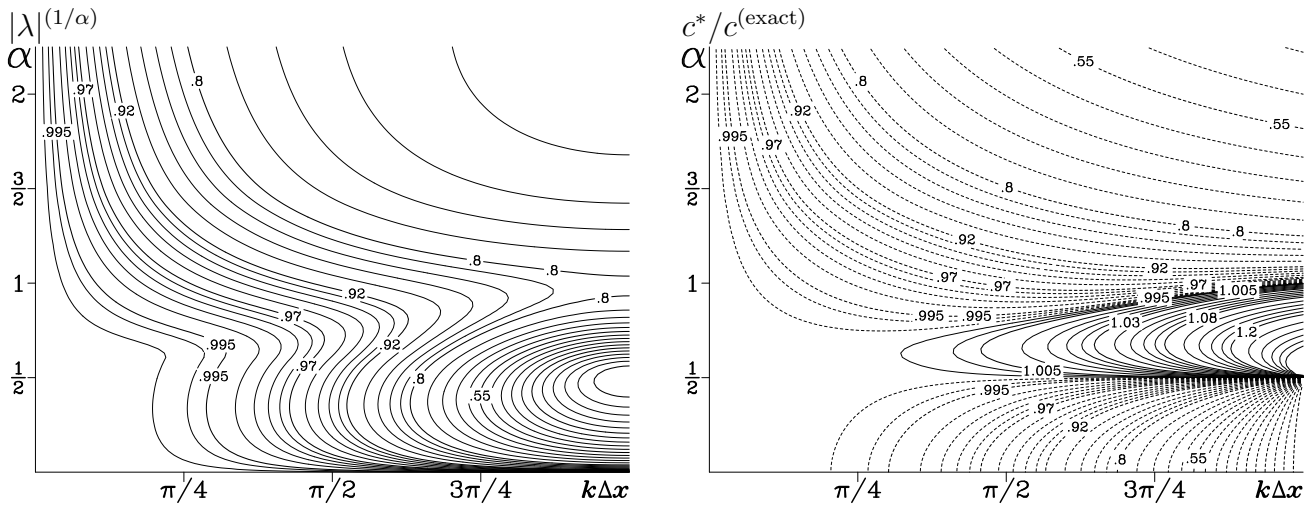
31

Fig. 18. Same as two lower panels of Fig. ADPTLambdaFig, but for using semi-Lagrangian explicit part instead of LF-AM3 stepping.

be increased to some extent. However, in the present versions of ROMS code alternative (including semi-Lagrangian) advection schemes are used only for extra tracer fields associated with biological and sediment models, but not for temperature and salinity, and definitely not for momentum equations. It is structurally too cumbersome to keep a dual set of $\alpha_{min}, \alpha_{max}$ in the code, so we restrict ourselves by pointing out that the choice of $\alpha_{min}, \alpha_{max}$ motivated by LF-AM3 stepping does not face an additional stability restriction if some of the tracers use alternative advection schemes.

3.3 Changes to the code and computational cost

The adaptive algorithm fits into the existing infrastructure of ROMS code with some changes:

(*i*) Computation of sigma-coordinate vertical velocity is combined with computing the 3D advective Courant number, limiting algorithm, and splitting procedure, so the outcome is $W^{(e)}$ and $W^{(i)}$ instead of W. Because the inputs for Courant number are the same finite-volume fluxes as for the vertical velocity itself, the additional computations are naturally done within the same loops resulting in a more efficient code due to avoidance of extra load and store operations, which partially offsets the increased count of arithmetic operations;

(*ii*) The implicit tri-diagonal solvers for vertical viscosity and diffusion terms are modified to incorporate implicit part of vertical advection: once again, no extra loops have been added;

(*iii*) The original ROMS code of SM2005 skips computation of all viscous and diffusive terms (vertical and lateral) during the predictor stage of its time stepping algorithm for the baroclinic mode (thus, keeping only advection, Coriolis, and pressure-gradient terms). The modified code has implicit vertical solver on both stages, which is unavoidable due to the necessity to include vertical advection;

(*iv*) Once the above changes were made it was realized that the original code structure for the momentum equations consisting of `pre_step3d`, `step3d_uv`, and `rhs3d` subprograms (with the latter involving all computations common for predictor and corrector stages, *e.g.,* advection and Coriolis terms) is no longer optimal: while the combined implicit vertical advection-viscosity solver has significant commonality between the predictor and corrector stages, the differences are too large to use the same code without introducing extra logic or intermediate variables. Instead

₇₃₈ `rhs3d` was eliminated as a subprogram, its computations are now inserted directly (via CPP in-
clude command) into `pre_step3d` and `step3d_uv` where the predictor and corrector versions
of the implicit solver are done individually and consolidated with other computations – vertical
integrations associated with mode coupling, implicit bottom drag, etc. – these are the ones which
cause the differences. As the result no new provisional storage, no extra logic were introduced,
and the number of 3D-sweeps of storage arrays is kept to the minimum (hence optimal utilization
of CPU cache). An additional bonus of this approach is the possibility to have different horizontal
advection schemes for the momentum equations during predictor and corrector stages (4th-order
centered and 3rd-order upstream respectively) resulting in increase of stability limit relative to
3rd-order upstream during both stages, Appendix A, Fig. 24.

(*v*) The code is instrumented to monitor the largest vertical and horizontal Courant numbers in order
to provide on-line diagnostics about which one is the most limiting for each particular application
and to establish numerical safety margins [8] for Δt settings. This part can be switched off without
affecting model solution as it is completely outside the model algorithm.

The overall strategy of ROMS design to keep expensive parameterizations for vertical mixing coeffi-
cients and lateral mixing terms outside the predictor-corrector algorithm algorithm (thence compute
them only once per time step) is retained. In practice we observe only negligible increase of CPU time
needed to complete one time step: no more that 5%, which is well offset by the ability to run the model
using larger time step.

4 Computational examples

4.1 *Gravitational adjustment of density front ("lock-exchange" problem)*

To demonstrate viability of the proposed algorithm we present test results from gravitational adjust-
ment of density front, Fig. 19. The setup is the same as in Ilıcak et al. (2012), which is also a standard
ROMS test problem, Haidvogel and Beckmann (1999, see Sec. 6.3 there) in its turn inspired by the
classical work of Benjamin (1968) and early modeling study of Wang (1984). At the initial state the
basin is divided in two halves by a vertical diaphragm which separates two uniform water masses, one
with $T = 5^0 C$ on the left and the other with $T = 35^0 C$ on the right. Thermal expansion coefficient
(not normalized by density) is $0.2 kg/m^3/^0 C$, so the density contrast between the halves is $5 kg/m^3$.
The length of the domain is $64 km$, depth is $20m$, grid resolution $\Delta x = 400m$, $\Delta z = 0.5m$. The
vertical-to-horizontal aspect ratios, both $\Delta z/\Delta x = 1/800 \ll 1$ and $h/\Delta x = 1/20 \ll 1$, are small,
which means that this grid does admit nonhydrostatic effects. Nevertheless, this problem is known
to generate sharp fronts with vertical velocities playing by far the dominant restriction of the size of
time step allowed by numerical stability. This can be understood from the fact that once sharp front
propagates horizontally in the bottom half of the water column, so it takes time $\tau = \Delta x/u$ to cover
one horizontal grid interval, and water before the front is stagnant, during the same time τ the entire
grid-box-wide, $h/2$-deep volume should be uplifted by $h/2$ to give room for incoming water, hence it

---

[8] We should emphasize that for the purpose of adaptive control Courant number is 3D Courant number as it is
dictated by the overall budget of numerical stability for advection scheme. This means that the "explicitness"
available for the vertical direction is what is left after it has been "taxed" by the horizontal.

leads to an estimate

$$h/(2w) \sim \Delta x/u \qquad \text{or} \quad w \sim u \cdot h/(2\Delta x), \tag{4.1}$$

which has tendency to increase with $\Delta x \to 0$. (In a nonhydrostatic model this tendency will be eventually halted because the nonhydrostatic pressure makes water "feel" the approaching front beyond just one horizontal grid interval).
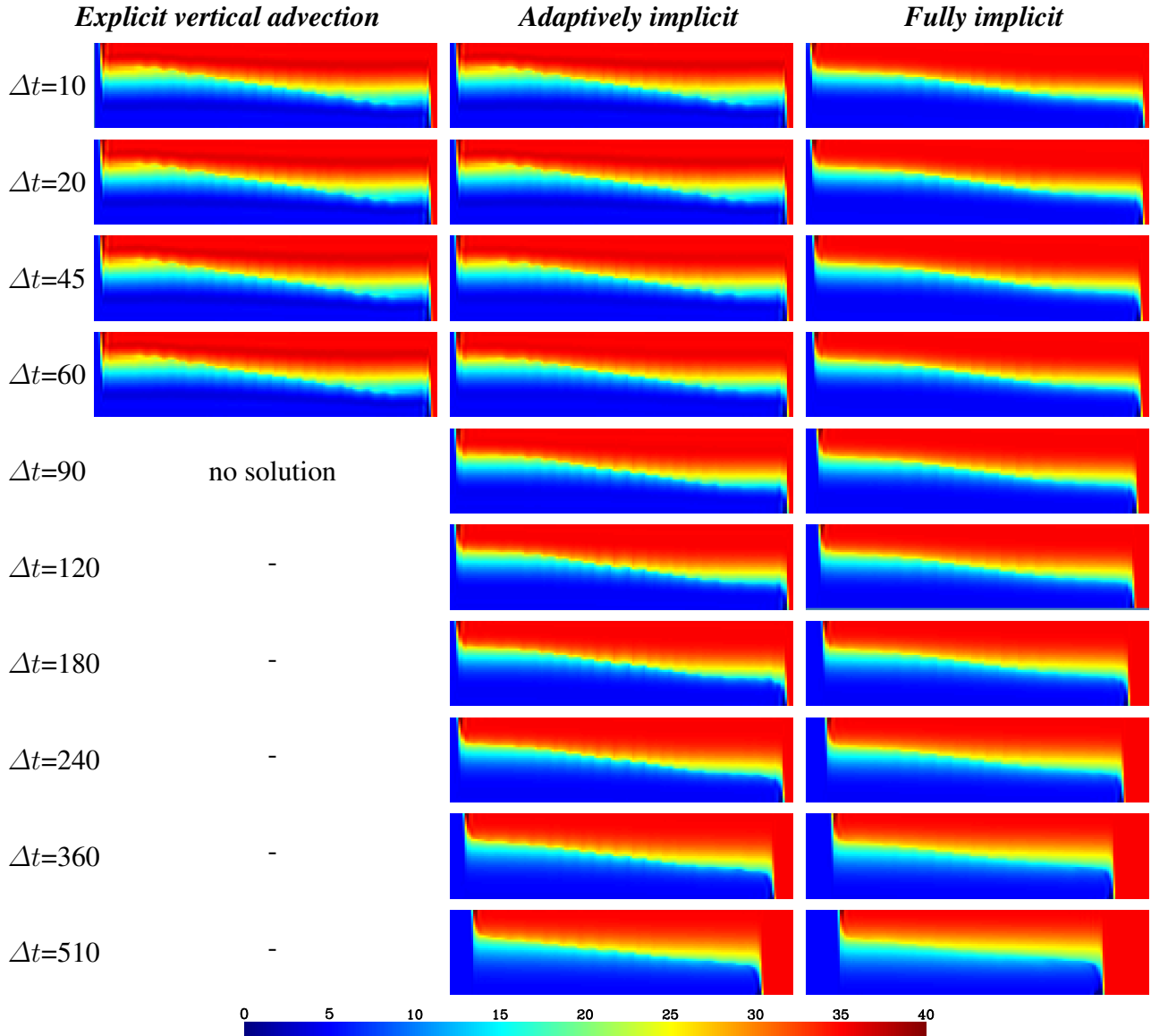

Fig. 19. Comparison of lock-exchange test solutions using explicit, adaptively implicit, and fully-implicit vertical advection algorithms.

Another remark to be made is that while Ilıcak et al. (2012) considers GOLD simulation as the "etalon" because the Lagrangian nature of vertical coordinate of this model makes it possible to obtain a solution without any numerically induced mixing at all, the hydrostatic posing of the problem makes it fundamentally non-convergent because decrease of horizontal grid spacing unavoidably leads to a situation when Δx becomes sufficiently small in comparison with the total depth and the thickness of denser fluid at which point the non-hydrostatic effects become resolved. Both laboratory experiments Rottman and Simpson (1983, see Fig. 2 there), Lowe et al. (2005, Fig. 10), and theoretical studies

Klemp et al. (1994) leading to a refinement of the conceptual model of Benjamin (1968) indicate that some mixing of the two fluids is inherent to this phenomenon in the case of small density contrast (e.g., where Boussinesq approximation in valid) due to the formation of the head with significant inertial motions in vertical direction (e.g., non-hydrostatic effects) followed by formation of a bore. However, the bulk outcome, such as the speed of propagation of the front can still be predicted by hydrostatic theory (*cf.,* Birman et al., 2005), perhaps with some closure assumption of how much water should be physically mixed.

The solution (vertical along-channel xz cross-section of temperature field) is shown at 17 hours since initialization. This matches Fig. 2 and Fig. 5 from Ilıcak et al. (2012). Explicit solution can be obtained for $\Delta t \leq 60s$, beyond which the code becomes numerically unstable. The adaptively implicit algorithm allows dramatic increase of the allowed time step. Unlike Ilıcak et al. (2012) who selected a Smolarkiewicz scheme for tracer advection (the best fit for this particular problem, but is too diffusive for realistic long-term simulations), we use a third-order upstream scheme in the horizontal, and parabolic spline in the vertical (for the explicit part) direction. They also choose to perform their tests with $\Delta t = 1s$ resulting in vanishingly small CFL, while our goal here is to push it to the limit. For the smallest settings of Δt, adaptive and explicit solutions are identical (which is expected), while for the largest Δt adaptive solution becomes more similar to fully implicit (backward Euler in time, upstream in space). Also note the progressive delay in the front propagation for the largest Δt – neither adaptive, nor fully implicit scheme is expected to be accurate at this regime ($\Delta t = 240...510s$), but still adaptive shows slightly less delay and less mixing.

4.2 Generation of large vertical velocities by vertical mixing: merging surface and bottom boundary layers

Vertical mixing parameterization schemes are an inherent part of any oceanic model. Normally they are physically formulated to act within each vertical column independently, and, with the exception of interpolations due to the necessity to bring horizontally staggered variables to a common location, these schemes do not imply any horizontal connectivity. This, and the highly nonlinear dynamical nature of these parameterizations may potentially create horizontal discontinuities in primary variables, and, large vertical velocities as the result of subsequent dynamical adjustment. Here we bring an example of such situation.

The experimental setup here is similar to the standard ROMS upwelling/downwelling test problem, except that constant-in-time analytical vertical viscosity/diffusivity profile is replaced with KPP boundary layer model, enabling both surface and bottom boundary layers. The model domain is a channel with free-slip wall boundaries on north and south sides, with depth raising from $150m$ to $25m$ toward the walls (see Fig. 20), and is uniform in east-west direction, where periodic boundary condition are are assumed. The width of the channel is $80km$, f-plane with Coriolis parameter $f = 10^{-4}$ (positive, Northern hemisphere) is assumed. Grid resolution $\Delta x = \Delta y = 500m$, resulting in 160 grid points across the channel and 48 non-uniformly spaced vertical layers with refinement toward both free surface and bottom. The initial condition is rest state ($u = v = 0$) with horizontally uniform positively stratified temperature profile,

$$\Theta\Big|_{t=0} = 14^0C + 4^0C \cdot \tanh\left(\frac{z + 35}{25}\right) + \frac{z + 120}{75} \; . \tag{4.2}$$

35

An idealized equation of state is assumed with thermal expansion coefficient $0.28 kg/\left(m^3 \cdot ^0C\right)$. Salinity is uniform and has no dynamical effect. At time $t = 0$ wind starts blowing in positive along-channel direction, according to

$$\tau_x = 0.1 \left[\frac{N}{m^2}\right] \cdot \begin{cases} \sin\left(\frac{\pi}{4} \cdot t[\text{days}]\right), & t < 2[\text{days}] \\ 1, & \text{otherwise}, \end{cases} \tag{4.3}$$

thus, smoothly reaching its full strength after 2 days, and remaining constant thereafter. There is no heat forcing at the surface.

K-Profile Parameterization (KPP) canonically referenced to Large et al. (1994) interprets turbulent mixing as a quasi-equilibrium process adapting to forcing conditions (surface wind stress and thermodynamic fluxes), and relies on bulk Richardson number criterion in determining the extent of boundary layer (hence treating the boundary layer as a whole). For our experiments we use an integral criterion (Shchepetkin, 2005) with a 3-way weighting procedure for vertical shear of horizontal velocity vs. buoyancy stratification vs. Coriolis force. The latter is recognized as a stabilizing factor acting against shear, and, in principle, able to balance it alone even in the absence of stratification, resulting in finite depth of boundary layer. The neutrally stratified case with neutral buoyancy forcing is tuned to yield the correct Ekman depth of $0.7u^*/f$ as the result of the solution (as opposite to a posteriori imposed limit in the original KPP; u^* is friction velocity associated with surface wind stress forcing). If no Coriolis force, the shear vs. positive buoyancy stratification case is tuned to match bulk Richardson number criterion for boundary layer thickness. Finally, negative buoyancy forcing in the absence of wind and Coriolis produces deepening matching the empirical rule of convection (Eqs. (21), (24), and (23) from Large et al., 1994 respectively). If the most general case when all three factors are present, their influences are considered in continuously-weighted manner (as opposite to using logical on-and-off switches) resulting in mostly "if-less" algorithm. The bottom boundary condition is described in Shchepetkin et al. (2009) and the resultant bottom drag drives the bottom boundary layer using similar rules with the exception that there is no buoyancy forcing. If the surface and the bottom boundary layers overlap, their viscosities/diffusivities are added as

$$A = \sqrt{A_{\text{surf}}^2 + A_{\text{bot}}^2} \tag{4.4}$$

where A_{surf}, A_{bot} are surface- and bottom-induced turbulent viscosities and diffusivities associated with the respective boundary layers. This merging rule is motivated by the fact that

$$A \sim w^* \cdot L \tag{4.5}$$

where L is length scale and w^* is turbulent velocity scale, which in its turn is proportional to $\sqrt{TKE}$, where TKE is turbulent kinetic energy. When boundary layers merge, their length scales became common, while TKEs are added up.

The results are shown in Fig. 20**a**, set of 5 panels in upper-left. Soon after the initialization wind causes formation of surface Ekman spiral and surface boundary layer, however as the near-surface velocity is pushed to the right relative to wind direction (that is to the left on the plot), it is restricted by the wall, and immediately produces displacement of free surface resulting in opposing barotropic pressure gradient. This causes compensating flow in the water column below, which in its turn interacts with the bottom resulting in formation of bottom boundary layer. The compensation flow (most evident
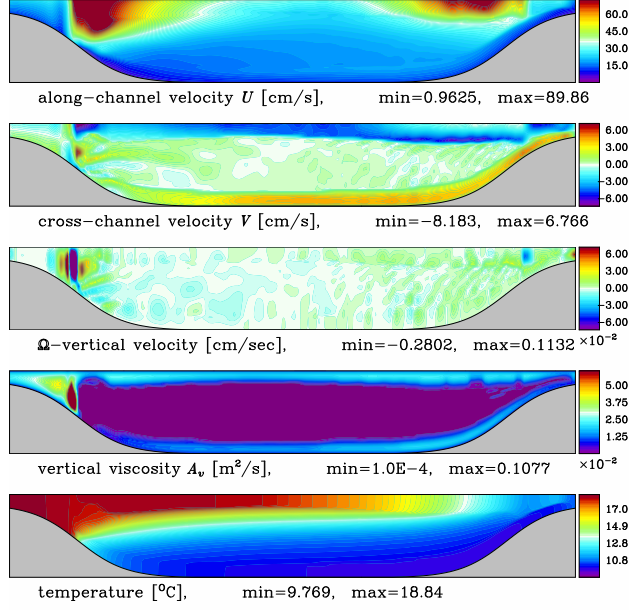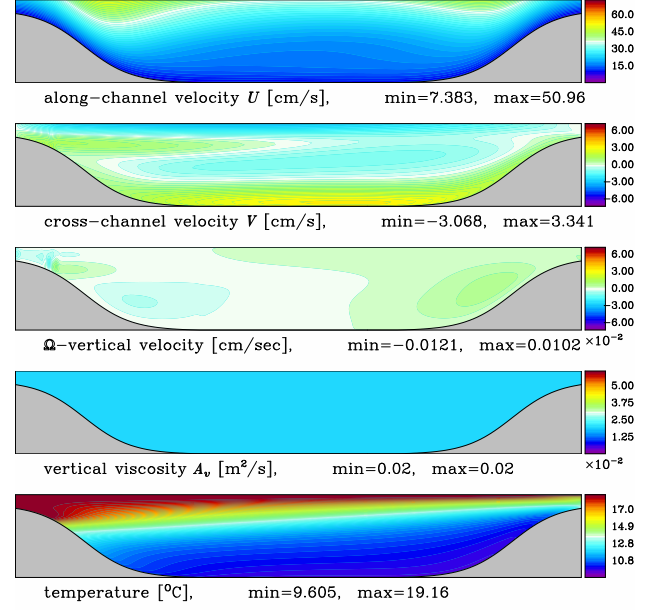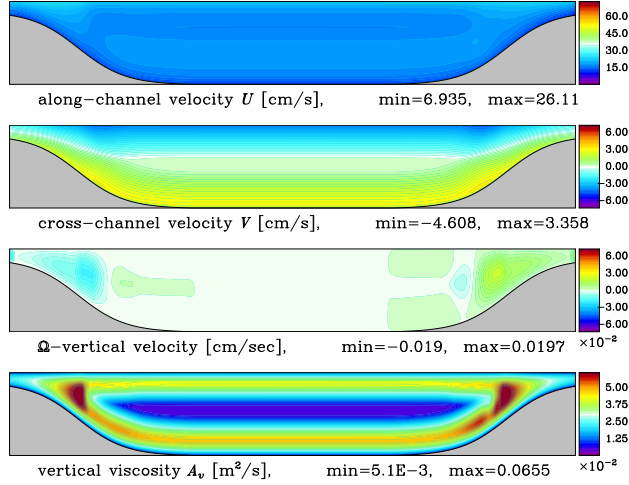
a: KPP, surface and bottom

along−channel velocity U [cm/s],   min=0.9625,   max=89.86

cross−channel velocity V [cm/s],   min=−8.183,   max=6.766

Ω−vertical velocity [cm/sec],   min=−0.2802,   max=0.1132 ×10⁻²

vertical viscosity A_v [m²/s],   min=1.0E−4,   max=0.1077   ×10⁻²

temperature [⁰C],   min=9.769,   max=18.84

c: $A_v = const$, $A_\Theta = 0$

along−channel velocity U [cm/s],   min=7.383,   max=50.96

cross−channel velocity V [cm/s],   min=−3.068,   max=3.341

Ω−vertical velocity [cm/sec],   min=−0.0121,   max=0.0102 ×10⁻²

vertical viscosity A_v [m²/s],   min=0.02,   max=0.02   ×10⁻²

temperature [⁰C],   min=9.605,   max=19.16

b: KPP, no stratification

along−channel velocity U [cm/s],   min=6.935,   max=26.11

cross−channel velocity V [cm/s],   min=−4.608,   max=3.358

Ω−vertical velocity [cm/sec],   min=−0.019,   max=0.0197 ×10⁻²

vertical viscosity A_v [m²/s],   min=5.1E−3,   max=0.0655   ×10⁻²

Fig. 20. Generation of large vertical velocities by merging surface and bottom boundary layers in wind–driven upwelling/downwelling problem. Shown on all panels here are vertical sections in cross-channel direction of the fields indicated under each plot. All are instantaneous snapshots are taken at $t = 8$ days since model initialization. *upper-left* – **a:** KPP surface and bottom boundary layer model with stratified initial condition; *lower-left* – **b:** same as **a**, but no stratification (temperature is set to uniform $\Theta = 14^0 C$, not shown); *upper-right* – **c:** same as **a**, but uniform vertical vertical viscosity instead of KPP. Note that only case **a** develops flow with large vertical velocities.

on v-panel) is confined to the bottom boundary layer, which is also manifested by "braking" effect on u-velocity near the bottom (more detailed examination reveals formation of bottom Ekman spiral there) and by erosion of stratification within the bottom layer. With all the parameters specified above the surface-layer Ekman depth is estimated as

$$0.7 u^* / f = 0.7 \sqrt{\tau_x / \rho_0} \Big/ f = 0.7 \cdot \sqrt{(0.1 \, N \cdot m^{-2}) / (1000 \, kg \cdot m^{-3})} \Big/ \left(10^{-4} c^{-1} \right) = 70 m \qquad (4.6)$$

and the expected thickness of surface boundary layer should be smaller because it is positively stratified. The thickness bottom layer is expected to comparable as the bottom stress should be, in principle, the same when along-channel flow reaches equilibrium, and less before that. Therefore the two boundary layers cannot reach each other in the deep part in middle of the channel, however they can do so in

the shallow. When this happens, the resultant surface-to-bottom mixing eliminates the remaining stratification, leaving behind top-to-bottom turbulent Ekman spiral (the left-most portion in the upper-left set of panels in Fig. 20). This interacts with the adjacent areas where boundary layers did not merge yet, and the discontinuities in u and v form large vertical velocities. To a large extent this front acts like a wall causing downward flow necessary to close the circulation, but it also generates internal waves due to loss of nearly geostrophic balance by sudden mixing effect. Merging of boundary layers also occurs on upwelling side of the channel, but vertical velocities there a weaker and occur at later time. This mechanism of generating large vertical velocities is nonlinear, and fundamentally relies on both dynamical feedback of influence of stratification onto boundary layers and vice versa. Thus, the loss of stratification by itself in not sufficient. Fig. 20**b** presents the same experiment, but with non-stratified setup. In this case very soon after initialization an equilibrium Ekman flow is formed. Vertical velocities are very weak, despite the fact that boundary layers do merge and there are places with sufficiently sharp contrast in vertical viscosity in horizontal direction. In the opposite case, Fig. 20**c** there is stratification, but the dynamical feedback of the flow and stratification onto vertical viscosity/diffusivity is eliminated – a constant value is used instead of KPP. There is upwelling with incline of thermocline comparable to the case **a** and also there is formation of non-stratified region on the downwelling side. However vertical remains very small. In all three cases we use time step $dt = 450s$ with vertical Courant numbers exceeding 1 in the case **a**, where, obviously, adaptively-implicit algorithm is essential [9]. In the other two cases, **b,c**, the threshold for implicitness was never reached.

4.3 Pacific Ocean model

An example for eddy-permitting resolution modeling is a 0.22-degree North-Equatorial Pacific model. The entire domain covers from 48^0S to 65^0N and from 99^0E to 290^0E with isotropic grid resolution, $\Delta x = \Delta y \sim 21km$ on the Equator to $10km$ at the extreme north of the grid. This is a limited-area configuration with open boundaries on the southern and western sides. The boundary forcing data was derived from SODA Ocean Climate Reanalysis (Carton and Giese, 2008) and applied via off-line nesting procedure (Mason et al., 2010). The model is forced by winds and thermodynamic fluxes derived from NCEP2 reanalysis (Kanamitsu et al., 2002) (in some runs wind stresses are substituted by ERA-40 (Uppala et al., 2005) available via SODA solution). Only a portion, 12^0S to 42^0N and 105^0E to 160^0E, of the entire model domain (approximately 1/3 in east-west and 1/2 in north-south direction) in the western part is shown in Fig. 21. Shown on the two right panels are the vertical maxima of horizontal, c_x, and vertical (in sigma-coordinate sense), c_w, Courant numbers: these two are computed separately by keeping only U, V- terms in (3.7), and, conversely, by keeping only W at each grid point, and then take maximum value over each vertical column. All the fields are instantaneous. The square pixels on each panel correspond to the actual model grid boxes – there is no interpolation or post processing of any kind.

Time step is $\Delta t = 2250\,s$ in this simulation, which results in horizontal Courant number staying well within the safe range of $\alpha \leq 0.5$ (in fact, the color scale on the upper right panel goes up to 0.25, however the extreme values in the panel go slightly beyond that, $\alpha_{\max} \sim 0.3$, where the red

---

[9] It should be noted that the perfect uniformity in along-channel direction for this setup makes it effectively two-dimensional in cross-channel vertical plane. As the result, horizontal Courant number due to u-velocity alone reaches 0.9, however u-velocity does not affect the numerical stability at all.

color changes toward more white-orange. In contrast, spatial distribution of maximum vertical Courant number is very different. Note that both right panels use the same color scheme, but in the case of c_w the color map is stretched to highlight the contrast for small values. So everything which is blue to green corresponds to very small values, and, in fact, c_w is small everywhere, expect just in a few "hot" spots. At such resolution the correlation between large values of c_x and c_w is not simple (as the
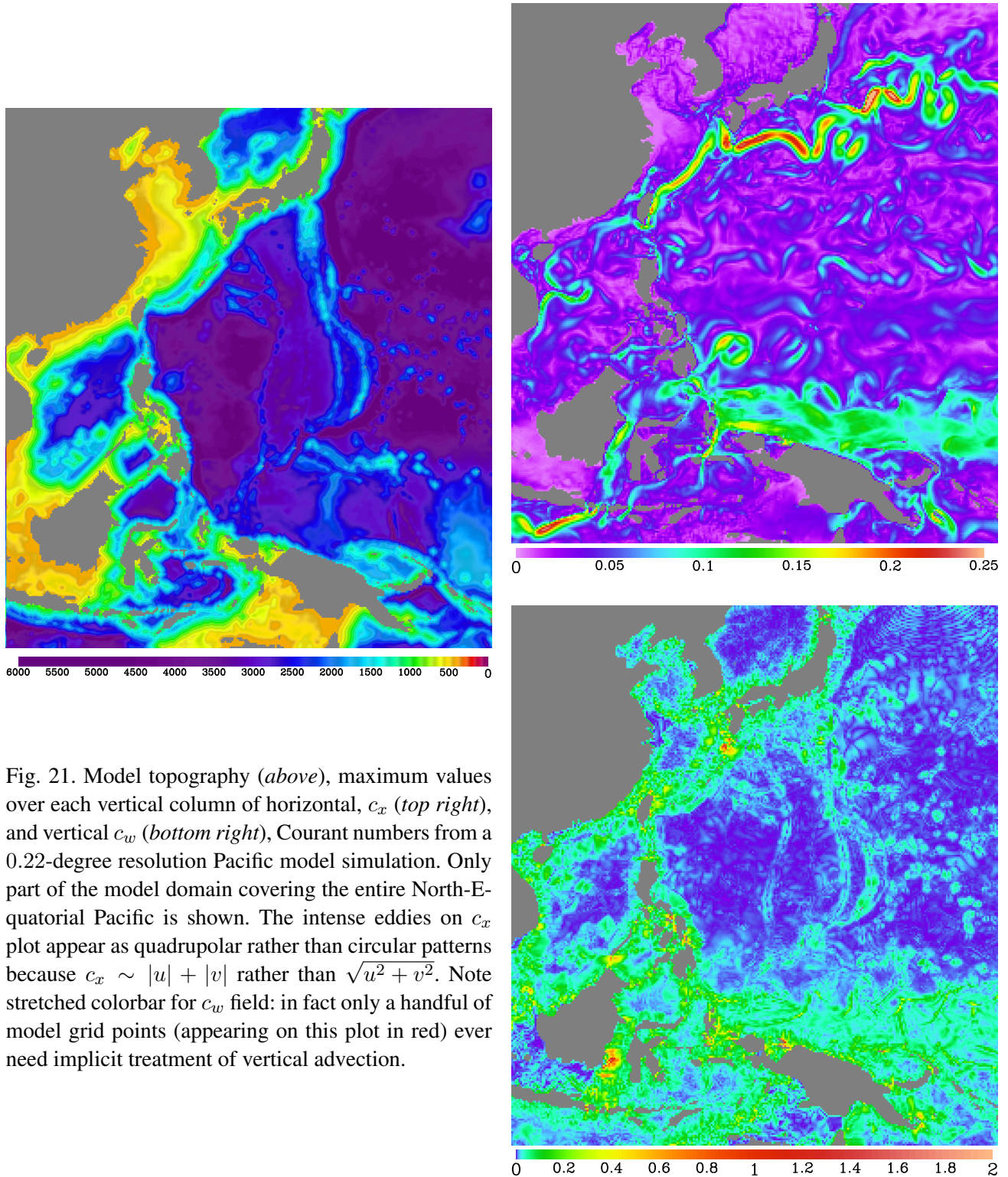
Fig. 21. Model topography (*above*), maximum values over each vertical column of horizontal, c_x (*top right*), and vertical c_w (*bottom right*), Courant numbers from a 0.22-degree resolution Pacific model simulation. Only part of the model domain covering the entire North-Equatorial Pacific is shown. The intense eddies on c_x plot appear as quadrupolar rather than circular patterns because $c_x \sim |u| + |v|$ rather than $\sqrt{u^2 + v^2}$. Note stretched colorbar for c_w field: in fact only a handful of model grid points (appearing on this plot in red) ever need implicit treatment of vertical advection.

39

extreme values of c_w do not necessarily occur where currents are the strongest), however it is clear that topographic slopes play a major role. The largest values of c_w occur intermittently (not always present in specific place and any time), however they are generally bound to appear only within the specific places in the grid. We estimate gain in computational efficiency of a factor of 1.5 or more (but within 2) relative to the fully explicit code. Most importantly the use of vertically adaptive advection makes this simulation to be very robust, as it relieves from the necessity to keep safety margin in Δt setting as vertical velocities are hard to predict a priori, so in practice one has to make several trial simulations before deciding it.

4.4 Palos Verdes Configuration

An example of very fine resolution ROMS configuration is Palos Verdes domain, Fig. 22. The physical size of the model domain is $120 \times 45 km$ in along-shore and offshore directions, adjacent to general Los Angeles area and oriented north-west – south-east. Horizontal grid spacing is only $75m$. Land mask is shown in gray color, the peninsula in the middle is Palos Verdes, left from it is Santa Monica Bay (with Marina del Ray visible as carved area into the mask; also identifiable along the coastline are port of Los Angeles on the right edge of peninsula, and Newport Beach marina). This particular domain is the innermost member of the 4-level set of one-way nested domains (hence it is designated as PVL4 domain) set up by Yusuke Uchiyama with the intent do fine-resolution studies of coastal response to wind bursts. The outer domains are $\Delta x = 250m$, $750m$ covering progressively larger areas, and finally $2.5km$ US West Coast configuration. Off-line nesting technique (Mason et al., 2010) is applied, with the exception that the L2 domain also receives tidal forcing. The PVL4 grid has $1600 \times 600 \times 32$ points. The maximum depth in this area is only $900m$. The minimum is restricted to $2m$. There is no wetting and drying (as there are no tidal flats in in this area), however tidal amplitudes may achieve a substantial fraction of this minimum depth. It is forced by WRF modeled winds (atmospheric model $\Delta x = 6km$ specifically computed for this purpose.)

The primary computational challenge in this configuration is the combination of very high intermittency of wind and tidal forcing (the latter are due to spring tides, where lunar and solar components are in phase), fully-developed bottom boundary layer (which may merge with the upper boundary layer resulting in vanishing stratification in the entire water column), also constriction of the tidally generated flows by man-made obstacles (water breakers).

Time step $\Delta t = 28.8s$ with 60 barotropic steps during each baroclinic using the adaptively-implicit vertical advection with parabolic splines for the explicit part for both momentum and tracer equations. This is nearly the maximum Δt for which we were able to run this simulation (it subject to the restriction of divisibility of time interval between the outputs by integer number of time steps, 2 hours and 250 respectively). Slightly larger settings, $\Delta t = 30s$ and $32s$, are possible, but require tedious babysitting because the wind forcing for configuration is characterized by large temporal variation in strength so completion of a substantial portion of the run at a certain Δt setting does not always guarantee numerical stability for the entire duration. A fully-explicit code runs for Δt up to $20s$, but only if vertical splines are replaced with fourth-order scheme with harmonic averaging of consecutive differences (Shchepetkin and McWilliams, 2003, Sec. 4, Eqs.(4.9)-(4.12)), known in ROMS community as AKIMA advection. Switching to splines forces further reduction of time step to $\Delta t = \sim 17...18s$. This can be explained be the fact, that, while it is nonlinear, in the case of monotonic profile of the advected field AKIMA becomes similar to the conventional 4th-order finite-difference scheme, and therefore
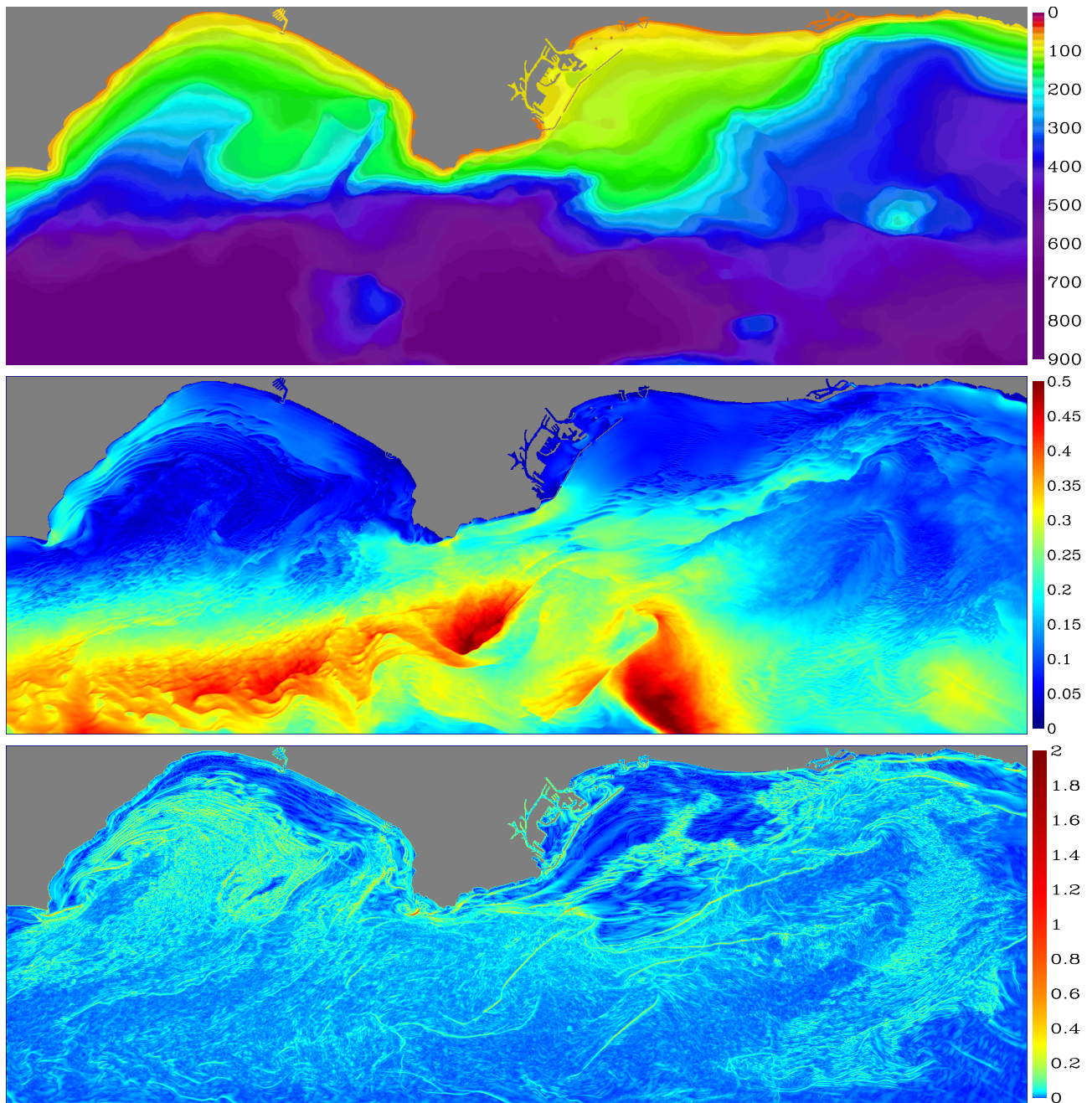
Fig. 22. US West Coast Palos Verdes Level 4 (PVL4) configuration. Model topography (*top panel*), maximum values over each vertical column of horizontal, c_x (*middle*), and vertical, c_w, (*bottom panel*) Courant numbers. Grid resolution is $\Delta x = 75m$, $1600 \times 600 \times 32$ points. The entire domain is shown. Maximum depth within the domain is $900m$, minimum is only $2m$. This solution was obtained using WRF-modeled winds and thermodynamic fluxes (atmospheric model resolution $\Delta x = 6km$) and it is tidally forced. Note stretched colorbar of c_w — vertical Courant numbers are actually very small everywhere except within the frontal structures. The maximum c_w in this Fig. occurs near the south-western tip of PV peninsula.

958 its dispersion curve (Fig. 1) has lower maximum value than that of compact differencing, resulting
959 in larger stability limit if the same time stepping algorithm is used. On the other hand, stability of
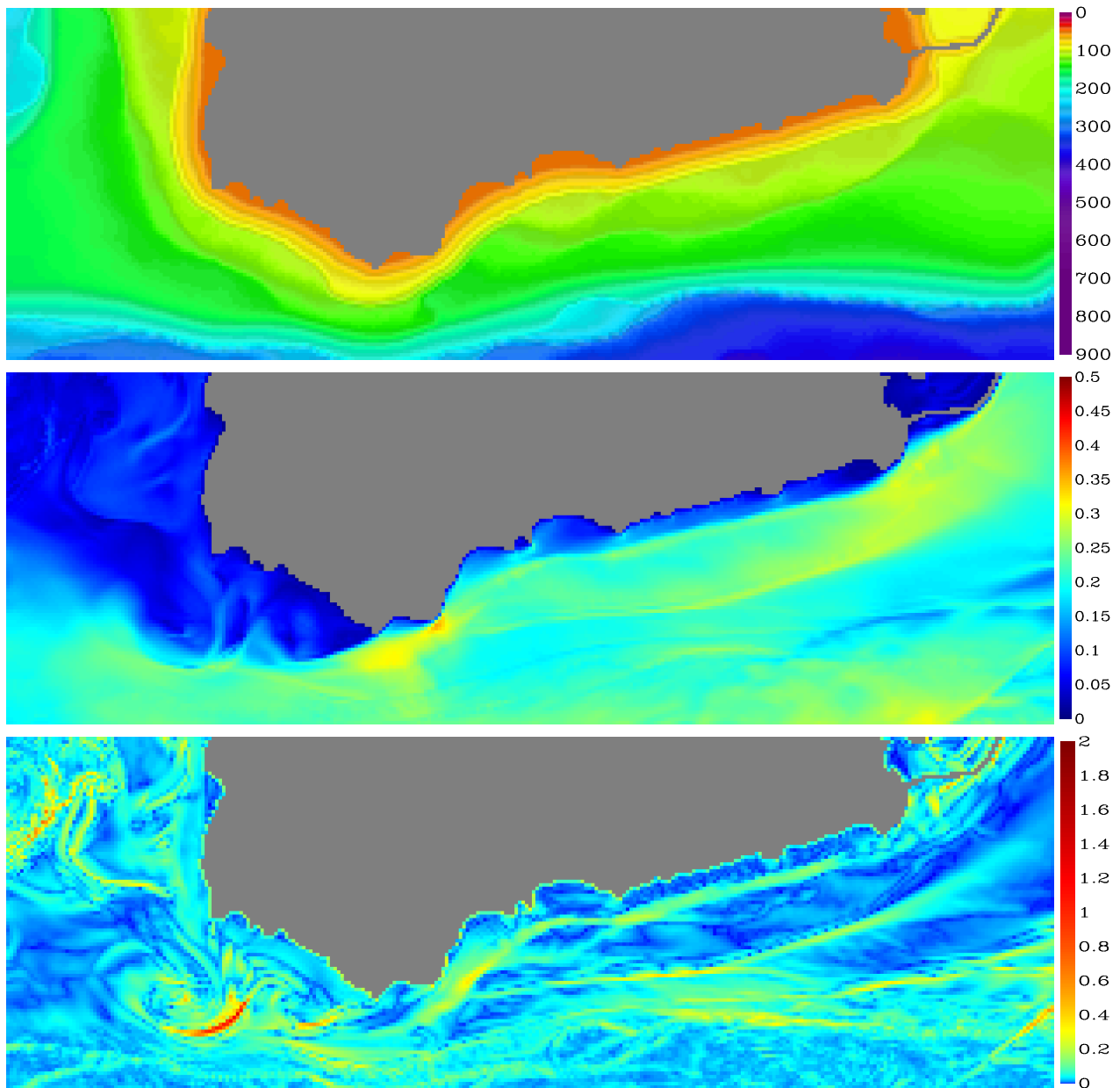960 adaptively implicit code is virtually insensitive to the choice between splines and AKIMA.

41

Fig. 23. An enlarged central portion of the previous figure. All fields, value ranges, and color schemes are the same. The square pixels of these plots correspond to the actual grid boxes of the model grid – there is no interpolation, averaging, or post-processing of any kind. The maximum measured value of c_w in this snapshot is 1.328 occurring within the curved frontal structure off the south-western shore of PV peninsula.

Shown on the two lower panels Fig. 22 are water-column maxima of horizontal and vertical Courant numbers (defined similarly to that on Fig. 21) during a wind burst effect. All the fields are instantaneous at a moment selected when the model reports large vertical Courant number. Note that similarly to Fig. 21 the color bar for c_w is stretched. Once again, the values of vertical Courant numbers are very small everywhere except occasionally within the frontal structures, which are generated by the inherent ocean dynamics – it should be noted that the surface forcing fields are very smooth relative to the resolution of this oceanic grid. Unlike in the Pacific model, occurrences of extreme values of c_w are not topographically locked (although, obviously, topography plays role), but are mainly event driven.

In this particular snapshot the location of extreme value of c_w was traced back to the event near southwestern tip of PV peninsula, and shown in more detail in Fig. 23. The flow is predominantly wind driven and is moving from the right to the left passing the tip of PV peninsula at which point it separates, leaving a nearly stagnant "shadow" area between the coast and the separated flow. This situation forces coastal upwelling in the area, resulting in sharp horizontal contrast in density field at the front, which eventually leads to large vertical velocity, and the extreme of vertical Courant number way beyond the the limit of what can be handled by the explicit code.

5   Conclusion

Using implicit algorithms is common for diffusive processes, however they are rarely selected for advection. The explanation of this comes from realization that viscous and diffusive processes often lead to a situation where the solution adapts to external conditions (through forcing, or boundaries other terms in equations) which results in an equilibrium, so the time evolution is slow in comparison with the time interval needed to establish the equilibrium. Advection is different in principle the underlying process is propagation, hence accurate modeling requires resolving it in time, so the time step size must be chosen to be sufficiently small to allow meaningful representation of the phase changes of the highest wavenumber admitted by the computational grid. This translates into staying within $c\Delta t/\Delta x < 1$.

Explicit advection algorithms can be designed in such a way that using them in the computational regime close to the largest possible time step allowed by numerical stability does not compromise the accuracy of the solution. Implicit algorithms promise to circumvent the CFL limitation, however their drawbacks are two fold: at first, if used in the $c\Delta t/\Delta x > 1$ regime, the Courant number itself also sets the fundamental limitation on the largest spatial wavenumber which can be propagated with the proper phase speed. Secondly, Dahlquist (1963) set a fundamental limitation that any linear (with constant coefficient) implicit method capable of unconditional stability is limited to at most second-order temporal accuracy – a Crank-Nicholson step over one or more Δt-intervals is the only option for the second-order accuracy, while modifying it by placing more weight onto the new-time-step r.h.s. terms changes it into θ-method, which is formally only first-order accurate – if used within $c\Delta t/\Delta x \sim 1/2$, neither option yields numerical accuracy to comparable that of to the best explicit schemes.

Adaptive implicit advection proposed in this article designed as an extension to the explicit, where weighting between the explicit and the implicit parts adjusts automatically to local flow condition based on Courant number, thus in goes beyond the class of constant-coefficient time stepping algorithms, allowing it to circumvent the Dahlquist (1963) limitation. For small Courant numbers and up to the useful portion of the stability range of the explicit part of the overall algorithm the accuracy is fully retained without any compromise at all. Implicitness activates itself only where and when it is necessary in a seamless manner by smoothly changing toward being more and more implicit. The implicit part is deliberately designed to have dissipatively-dominant truncation error - primarily to avoid dispersive ripples, which may be detected as negative stratification and interfere with vertical mixing parameterization in a non-controllable way.

The motivation, and, in fact justification, of such approach comes from the observation that in practical oceanic modeling results using ROMS code vertical velocities and vertical Courant numbers

43

are generally very small almost everywhere throughout the computational domain, except in just a tiny fraction of it, sometimes few dozen grid points, where the extreme values are triggered by either loss of stratification due to local overturning, mixing events, propagation gravity fronts in density fields, specific topographic places causing focusing and trapping of vertical motions, internal wave breaking, tidally-induced mixing in shallow areas, etc – all of which are characterized by strong vertical mixing, and where the elevated numerical dissipation due to the advection scheme is acceptable.

The practical benefit of the adaptive algorithm is the ability to run model with a substantially larger time step without compromising numerical quality of the solution. The gains are application dependent, primarily sensitive to roughness of bottom topography and intensity of the flow. Typically we observe increasing ratios of time steps allowed by the adaptive code and its explicit prototype when going to finer horizontal resolutions, with exceeding a factor of 3 for a $\Delta x \sim 750m$ inner Gulf Stream nest model practically observed. They are more modest for US West Coast configurations (traditional for UCLA ROMS), where currents are generally weaker and internal wave phase speed is typically the most limited. This situation, however, changes with the inclusion of tides.

Appendix A: Finite Courant number behavior of some of the commonly used advection schemes

Fourth-order centered finite-difference in space with LF-AM3 time stepping: the algorithm essentially repeats (2.6)-(2.7) except that midpoint values $\tilde{q}_{j+1/2}$ are computed as

$$\tilde{q}_{j+1/2} = -\frac{1}{12}q_{j-1} + \frac{7}{12}q_j + \frac{7}{12}q_{j+1} - \frac{1}{12}q_{j+2} \tag{A.1}$$

instead of parabolic spline interpolation (2.5). The corresponding finite-difference approximation for the first derivative

$$\frac{\tilde{q}_{j+1/2} - \tilde{q}_{j+1/2}}{\Delta x} = \frac{-q_{j+2} + 8q_{j+1} - 8q_{j-1} + q_{j-2}}{12\Delta x} \tag{A.2}$$

has Fourier image

$$i\mathscr{K}\Delta x = \frac{4}{3}i\sin(k\Delta x) - \frac{1}{6}i\sin(2k\Delta x) = ik\Delta x - i\frac{(k\Delta x)^5}{30} + ... \tag{A.3}$$

which indicates the fourth-order spatial accuracy. The function $\mathscr{K}(k)\Delta x = \frac{4}{3}\sin(k\Delta x) - \frac{1}{6}\sin(2k\Delta x)$ reaches its maximum value of $\mathscr{K}\Delta x = \sqrt{\sqrt{6} - 3/2} \cdot (1 + 1/\sqrt{6}) \approx 1.37222197$ when $k\Delta x = \mathrm{acos}\left(1 - \sqrt{3/2}\right) \approx 0.57215487 \cdot \pi$. This maximum value is smaller and is on the left from the corresponding $\mathscr{K}\Delta x = \sqrt{3}$ at $k\Delta x = 2\pi/3$ for the 4th-order accurate Padè derivative on Fig. 1.

The characteristic equation is essentially the same as (2.14) except that $i\mathscr{K}\Delta x$ now has different meaning, however it is still purely imaginary. There are two characteristic roots, one for the physical mode and the other one is for computational. The upper row on Fig. 24 shows dissipation per $1\Delta x$ travel and numerical to ideal phase speed ratio for this algorithm in the same format as the corresponding panels on Fig. 2. Qualitatively both $|lambda|^{(1/\alpha)}$ and $c^*/c^{(\mathrm{exact})}$ are very similar for both algorithms, however Fig. 24 has noticeable narrower

region where the phase speed ratio falls within the range of 1 ± 0.005 which is explained by a larger truncation error of (A.3) vs. (2.13). Fig. 24 also has slightly larger stability limit and smaller dissipation, both of which are due to lower maximum value of $\mathscr{K}\Delta x = \mathscr{K}\Delta x(k\Delta x)$ for the finite difference scheme.
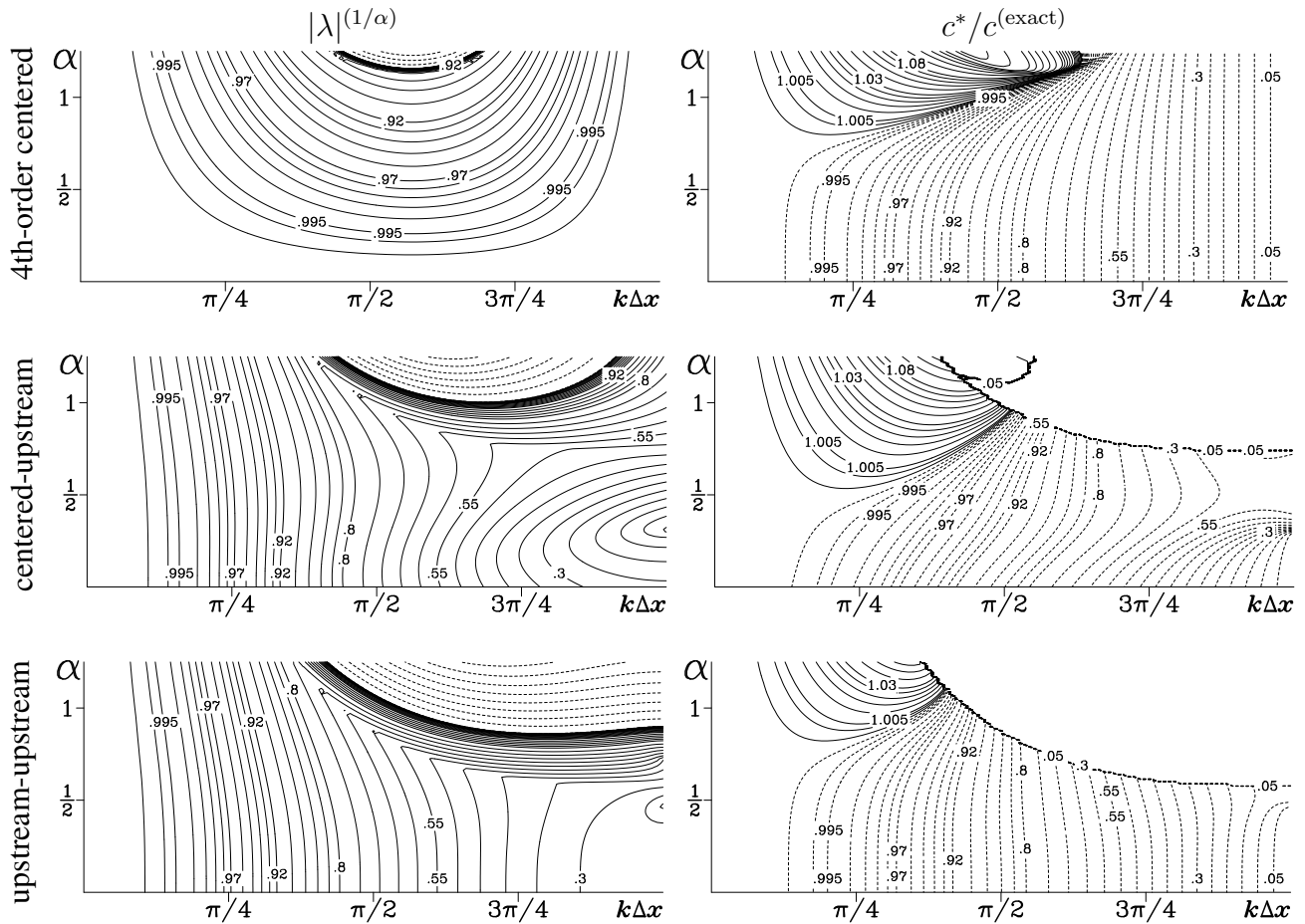

Fig. 24. Dissipation per $1\Delta x$ travel, $|\lambda|^{(1/\alpha)}$ (*left column*), and numerical to ideal phase speed ratio, $c^*/c^{(\mathrm{exact})}$ (*right*), for advection algorithm using LF-AM3 time stepping combined with 4th-order centered (A.1) and/or third-order upstream-biased (A.4) discretization in space: *upper row* – centered for both predictor and corrector stages; *medium row* – centered for predictor stage, upstream for corrector; *bottom row* – upstream for both stages. Note that stability ranges are smaller for both cases involving the use of upstream-biased scheme ($\alpha_{\max} \approx 1.156,\ 1.003,\ 0.861$, top to bottom respectively), and in comparison with the centered case, $k\Delta x$ for which instability occurs first is shifted toward the higher wavenumbers beyond $k\Delta x$ of the maximum on the corresponding dispersive curve in Fig. 1 ($k\Delta x \approx 0.572\pi,\ 0.685\pi,\ 0.795\pi$ respectively).

Another widely adopted scheme is the third-order upstream-biased. It used an asymmetric stencil to compute the mid-point value $\tilde{q}_{j+1/2}$

$$\tilde{q}_{j+1/2} = -\frac{1}{6}q_{j-1} + \frac{5}{6}q_j + \frac{1}{3}q_{j+1} \tag{A.4}$$

in such a way that it uses one more point on the upstream side (*i.e.,* the above formula is for positive advecting velocity; for negative it should involve q_{j+2} instead of q_{j-1}). The approximation for the first derivative

$$\frac{\tilde{q}_{j+1/2} - \tilde{q}_{j-1/2}}{\Delta x} = \frac{2q_{j+1} + 3q_j - 6q_{j-1} + q_{j-2}}{6\Delta x} \tag{A.5}$$

45

has Fourier image

$$
\begin{aligned}
i\mathcal{K}\Delta x &= \frac{4}{3}i\sin(k\Delta x) - \frac{1}{6}i\sin(2k\Delta x) + \frac{1}{2} - \frac{2}{3}\cos(k\Delta x) + \frac{1}{6}\cos(2k\Delta x) \\
&= i\sin(k\Delta x)\cdot\left(1 + \frac{1 - \cos(k\Delta x)}{3}\right) + \frac{\left(1 - \cos(k\Delta x)\right)^2}{3} \\
&= ik\Delta x + \frac{(k\Delta x)^4}{12} - i\frac{(k\Delta x)^5}{30} + \dots
\end{aligned}
\tag{A.6}
$$

which, unlike (A.3) is no longer purely imaginary. The $-(k\Delta x)^4/12$ term introduces numerical dissipation of hyperdiffusive type, which is present even for infinitely small Courant numbers. There are two variants: either the upstream-biased scheme is only used during the corrector stage of LF-AM3 step, while the centered is used for predictor, or the upstream is used for both. Their properties are shown on the middle and lower rows of Fig. 24. The two variants are virtually indistinguishable within the resolved portions ($k\Delta x < \pi/4$, $\alpha < 1$) of the $(k\Delta x, \alpha)$-plane, however the use of upstream-biased scheme during corrector stage only results in a larger stability limit (note that the use of of upstream-biased scheme leads to some decrease of the stability limit in comparison with the centered-only scheme in both cases, but it is more pronounced in the second), and somewhat decrease of phase error, which now has tendency to decrease with increase of Courant number.

QUICKEST semi-Lagrangian advection algorithm of Leonard (1979) interprets the given discrete values q_j as grid-box averages (rather than instantaneous values at points x_j) and uses parabolic reconstruction of the advected field within each cell

$$
q(x') = q_j + \frac{q_{j+1} - q_{j-1}}{2\Delta x}\cdot x' + \frac{1}{2}\cdot\frac{q_{j+1} - 2q_j + q_{j-1}}{\Delta x^2}\cdot\left(x'^2 - \frac{\Delta x^2}{12}\right)
\tag{A.7}
$$

where x' is a local coordinate defined within each cell, $-\Delta x/2 \leq x' \leq +\Delta x/2$. The parabola is constructed in such a way that its integral within the cell matches its given value,

$$
\frac{1}{\Delta x}\int\limits_{-\Delta x/2}^{+\Delta x/2} q(x')\,dx' = q_j\,,
\tag{A.8}
$$

and its first and second derivatives match the corresponding finite-difference approximations based on the neighboring discrete values. The finite-volume flux going from cell j into cell $j+1$ (assuming that the advecting velocity c is positive) is the computed as

$$
F_{j+1/2} = \int\limits_{+\Delta x/2 - c\Delta t}^{+\Delta x/2} q(x')\,dx' = c\Delta t\left\{q_j + \frac{q_{+1} - q_{j-1}}{4}\left(1 - \alpha\right) + (q_{+1} - 2q_j + q_{j-1})\left[\frac{1}{12} - \frac{\alpha}{4} + \frac{\alpha^2}{6}\right]\right\}
\tag{A.9}
$$

where $\alpha = c\Delta t/\Delta x$. The updated values are than

$$
q_j^{n+1} = q_j^n - \left[F_{j+1/2}^n - F_{j-1/2}^n\right]/\Delta x\,.
\tag{A.10}
$$

Note that unlike (2.8) the above expression for $F_{j+1/2}$ does not imply computing the interface values $\tilde{q}_{j+1/2}$ first. In fact, the interface values are not even uniquely defined, as the right and the left side limits of distributions (A.7) from the two adjacent cells are not equal to each other,

$$
\left[\lim_{x'\to+\Delta x/2} q(x')\right]_j \neq \left[\lim_{x'\to-\Delta x/2} q(x')\right]_{j+1}\,.
\tag{A.11}
$$

Assuming that q_j^n is a Fourier component, $q_j^n = \lambda^n \cdot e^i k \Delta x \cdot j$, finding the step multiplier for this algorithm is tedious, but straightforward,

$$
\begin{aligned}
\lambda = 1 - \alpha^2 \big(1 - \cos(k\Delta x)\big) - \alpha \left(1 - \alpha^2\right) \left[\frac{1}{2} - \frac{2}{3}\cos(k\Delta x) + \frac{1}{6}\cos(2k\Delta x)\right] \\
- i\alpha \sin(k\Delta x) \left[1 + \frac{1 - \alpha^2}{3}\big(1 - \cos(k\Delta x)\big)\right].
\end{aligned}
\tag{A.12}
$$

Its Taylor expansion in powers of $k\Delta x$ yields

$$
\lambda = 1 - i\alpha k\Delta x - \alpha^2\frac{(k\Delta x)^2}{2} + i\alpha^3\frac{(k\Delta x)^3}{6} - \left(2\alpha - \alpha^2 - 2\alpha^3\right)\frac{(k\Delta x)^4}{24} + i\alpha\left(\frac{8}{3} - \frac{11}{3}\alpha^2\right)\frac{(k\Delta x)^5}{120} + ...
\tag{A.13}
$$

where the terms for powers up to and including $(k\Delta x)^3$ match the "ideal" multiplier $\lambda = e^{-i\alpha k\Delta x}$. The deviation of $(k\Delta x)^4$ term from $+\alpha^4(k\Delta x)^4/24$ (in fact, appearance there of powers of α less that the fourth) indicates the dissipative (hyper-diffusive) nature of the leading-order truncation term, which affects the solution even if Courant number α becomes vanishingly small, as it starts with the first power of α.

Piecewise Parabolic Method (PPM) (Colella and Woodward, 1984) in another semi-Lagrangian algorithm based on parabolic reconstruction. Leaving its monotonicity limiters aside, it is equivalent to using (2.8) and (2.2), however the interface $\tilde{q}_{j+1/2}$ values are computed via local interpolation (A.1) instead of solving (2.5). Its step multiplier

$$
\begin{aligned}
\lambda = 1 - i\alpha\left(1 - \frac{\alpha}{2}\big(1 - \cos(k\Delta x)\big)\right) \cdot 2\sin\frac{k\Delta x}{2}\left(\frac{7}{6}\cos\frac{k\Delta x}{2} - \frac{1}{6}\cos\frac{3k\Delta x}{2}\right) \\
+ i\alpha^2(3 - 2\alpha)\sin(k\Delta x)\left[\cos\frac{k\Delta x}{2}\left(\frac{7}{6}\cos\frac{k\Delta x}{2} - \frac{1}{6}\cos\frac{3k\Delta x}{2}\right) - 1\right] \\
- \alpha^2\sin(k\Delta x)\cdot\sin\frac{k\Delta x}{2}\left(\frac{7}{6}\cos\frac{k\Delta x}{2} - \frac{1}{6}\cos\frac{3k\Delta x}{2}\right) \\
+ \alpha^2(3 - 2\alpha)\big(1 - \cos(k\Delta x)\big)\left[\cos\frac{k\Delta x}{2}\left(\frac{7}{6}\cos\frac{k\Delta x}{2} - \frac{1}{6}\cos\frac{3k\Delta x}{2}\right) - 1\right].
\end{aligned}
\tag{A.14}
$$

Once again, it can be verified that substitution of $\alpha = 1$ into above turns it into $\lambda = \cos(k\Delta x) - i\sin(k\Delta x)$, which is the exact value. Its Taylor expansion for small $k\Delta x$ is

$$
\lambda = 1 - i\alpha k\Delta x - \alpha^2\frac{(k\Delta x)^2}{2} + i\alpha^3\frac{(k\Delta x)^3}{6} + \alpha^2(2\alpha - 1)\frac{(k\Delta x)^4}{24} + i\alpha\left(\frac{1}{90} - \frac{\alpha}{12} + \frac{\alpha^2}{24}\right)(k\Delta x)^5 + ...
\tag{A.15}
$$

to derive which we have used expansions of the common expressions

$$
\begin{aligned}
2\sin\frac{k\Delta x}{2}\left(\frac{7}{6}\cos\frac{k\Delta x}{2} - \frac{1}{6}\cos\frac{3k\Delta x}{2}\right) &= k\Delta x\left[1 - \frac{(k\Delta x)^4}{90} + ...\right] \\
\cos\frac{k\Delta x}{2}\left(\frac{7}{6}\cos\frac{k\Delta x}{2} - \frac{1}{6}\cos\frac{3k\Delta x}{2}\right) - 1 &= -\frac{(k\Delta x)^2}{12}\left[1 + \frac{5}{12}(k\Delta x)^2 + ...\right].
\end{aligned}
\tag{A.16}
$$

Eq. (A.15) matches the ideal multiplier $e^{-i\alpha k\Delta x}$ for up to $(k\Delta x)^3$-term. The the first mismatch occurs in $(k\Delta x)^4$, but now the lowest power of α in this term is the second, which is principally different from (A.13), where its starts with the first power of α.

The properties of (A.12) and (A.14) are compared on Fig. 25. While the two algorithms are superficially very similar (both semi-Lagrangian, flux-integrated, based on piecewise parabolic reconstruction), and are of the same order of accuracy (the third), their subtle distinctions in their properties may lead to crucial distinctions in the solutions. The leading-order truncation error of PPM is dissipative (hyper-diffusive), however it is Courant

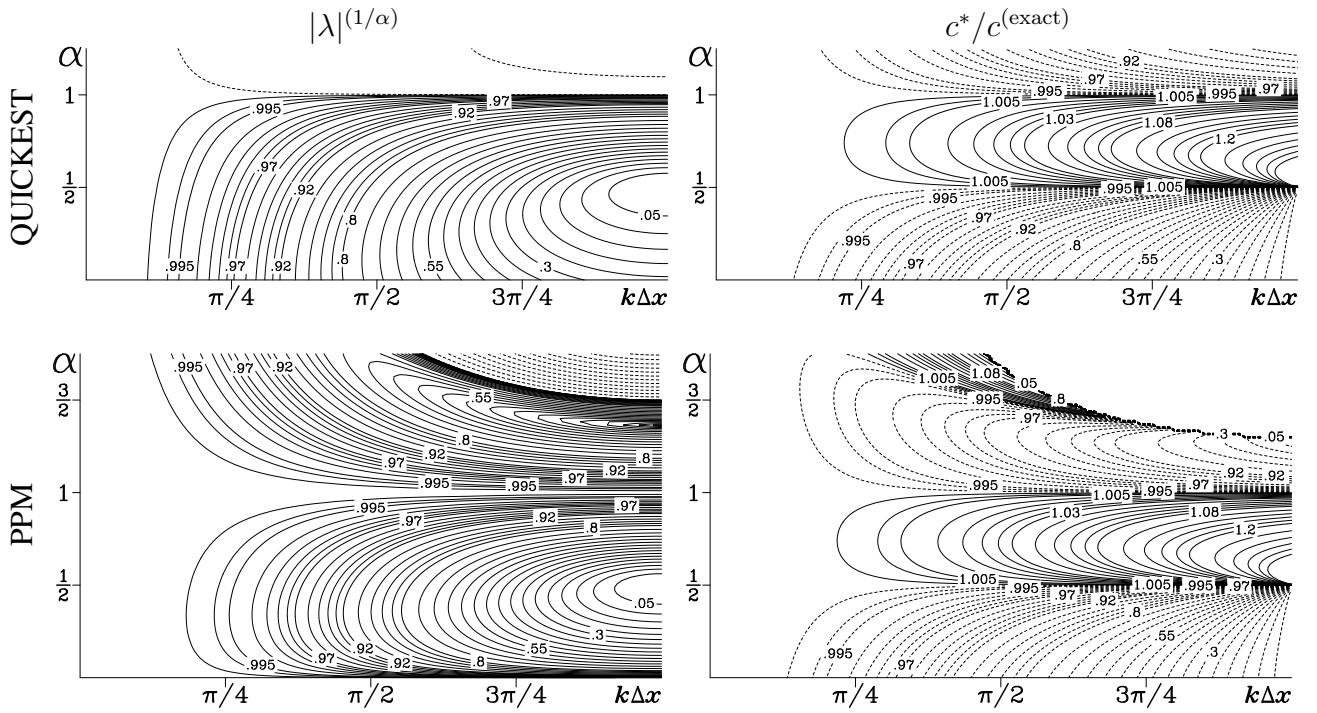Fig. 25. Dissipation per $1\Delta x$ travel, $|\lambda|^{(1/\alpha)}$ (*left column*), and numerical to ideal phase speed ratio, $c^*/c^{\text{(exact)}}$ (*right*), for QUICKEST (*upper row*) and PPM (*bottom*) semi-Lagrangian advection algorithms.

number dependent, and numerical dissipation of PPM vanishes when $\alpha \to 0$. QUICKEST dissipation is only weakly dependent of Courant number. The stability limit is $\alpha_{\max} = 3/2$, however this applies only for the version of PPM with all monotonicity limiters turned off. Once limiters are back on, the stability is guaranteed only until $\alpha_{\max} = 1$. QUICKEST stability limit is $\alpha_{\max} = 1$. On the other hand, the dispersive properties of the two schemes are virtually identical to each other, and furthermore, are very similar to LF-AM3 with centered-upstream discretization, see Fig. 24, middle row (Note that all the isolines of $c^*/c^{\text{(exact)}}$ approach $k\Delta x$ axis at exactly the same locations). The bottom row of Fig. 25 is qualitatively similar to that of Fig. 4, however spline reconstruction leads to a smaller leading-order truncation error of spatial discretization, so the contour-free area on the left portion of both lower panels on Fig. 4 is substantially wider than on Fig. 25.

Appendix B: Maximum phase increments per time step for common implicit advection schemes

Backward Euler in time, centered in space algorithm,

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} + c\frac{q_{j+1}^{n+1} - q_{j-1}^{n+1}}{2\Delta x} = 0\,, \tag{B.1}$$

after substituting Fourier component $q_j^n = \widetilde{q} \cdot \lambda^n \cdot e^{ik\Delta x}$ yields step multiplier

$$\lambda = \frac{1}{1 + i\alpha \sin\xi} = \frac{1 - i\alpha \sin\xi}{1 + \alpha^2 \sin^2\xi} \tag{B.2}$$

where we have introduced $\alpha = c\Delta t/\Delta x$ is Courant number and $\xi = k\Delta x$ is wavenumber normalized by grid interval. The ratio of imaginary to real parts of λ yields tangent of the phase increment,

$$\tan\phi = \frac{\alpha \sin\xi}{1}. \tag{B.3}$$

48

Ideally $\phi^{\text{(exact)}} = \omega\Delta t$, where $\omega = ck$, hence $\phi^{\text{(exact)}} = ck\Delta t = k\Delta x \cdot c\Delta t/\Delta x = \alpha\xi$, so the above is numerically consistent with the desired value, since both $\sin\xi \sim \xi$ when $\xi \to 0$ and $\tan\phi \sim \phi$ when $\phi \to 0$. However, departure from both infinitely small wavenumber and from infinitely small Courant number results in smaller-than-ideal phase increment for both these reasons. Since $0 \leq k\Delta x \leq \pi$, so $\sin\xi$ reaches its maximum value when $\xi = \pi/2$, which corresponds to $4\Delta x$-waves. This means that the phase increment for any wavenumber and Courant number is bounded by $0 \leq \phi \leq \pi/2$ regardless of how large is the value of α.

Backward Euler in time, upstream in space,

$$\frac{q_j^{n+1} - q_j^n}{\Delta t} + c\frac{q_j^{n+1} - q_{j-1}^{n+1}}{\Delta x} = 0\,, \tag{B.4}$$

has step multiplier,

$$\lambda = \frac{1}{1 + i\alpha\left(1 - e^{-i\xi}\right)} = \frac{1 + \alpha(1 - \cos\xi) - i\alpha\sin\xi}{1 + 2\alpha(1 + \alpha)(1 - \cos\xi)}\,, \tag{B.5}$$

which phase increment is

$$\tan\phi = \frac{\alpha\sin\xi}{1 + \alpha(1 - \cos\xi)}\,. \tag{B.6}$$

For each given α this function reaches its maximum value if $\cos\xi = \alpha/(1 + \alpha)$ resulting in

$$\max[\tan\phi] = \frac{\alpha}{\sqrt{1 + 2\alpha}} \begin{array}{l} \nearrow 1/\sqrt{2}\,, \quad \alpha \to +\infty \\ \searrow \alpha\,, \qquad \alpha \to 0 \end{array} \tag{B.7}$$

which bounds ϕ as

$$0 \leq \phi < \operatorname{atan}\left(1/\sqrt{2}\right) \approx 0.195913276 \cdot \pi\,. \tag{B.8}$$

This is narrower than similar range for (B.3), and the maximum phase increment occurs for a smaller wavenumber, $k\Delta x = \operatorname{acos}\left(\alpha/(1 + \alpha)\right)$ instead of $\pi/2$.

Appendix C: Two-dimensional stability analysis of one-dimensionally implicit adaptive algorithm

One potential concern about the one-dimensional analysis of the adaptively implicit algorithm in Sec. 3.2 is that does not cover the possibility of numerical instability of essentially a flux-splitting type (*cf.,* Leonard et al., 1996) associated with simultaneous application of advective fluxes computed one-dimensionally, separately in each direction, and using different mathematical expressions. This applies to the selection of upstream vs. centered scheme for the implicit part as well as the use of 3rd-order upstream-biased scheme in two horizontal directions, while reverting to centered in vertical. Here we explore these issues.

The two-dimensional analog of (3.15)-(3.16) is

$$q_{j,l}^{n+1/2} = \left(\frac{1}{2} - 2\gamma\right) q_{j,l}^{n-1} + \left(\frac{1}{2} + 2\gamma\right) q_{j,l}^n - (1 - 2\gamma)\left[\alpha'\left(\tilde{q}_{j+1/2,l}^n - \tilde{q}_{j-1/2,l}^n\right)\right. \\ \left. + \alpha''\left(q_{j,l}^{n+1/2} - q_{j-1,l}^{n+1/2}\right) + \beta\left(\tilde{q}_{j,l+1/2}^n - \tilde{q}_{j,l-1/2}^n\right)\right] \tag{C.1}$$

and

$$q_{j,l}^{n+1} = q_{j,l}^n - \alpha'\left(\tilde{q}_{j+1/2,l}^{n+1/2} - \tilde{q}_{j-1/2,l}^{n+1/2}\right) - \alpha''\left(q_{j,l}^{n+1} - q_{j-1,l}^{n+1}\right) - \beta\left(\tilde{q}_{j,l+1/2}^{n+1/2} - \tilde{q}_{j,l-1/2}^{n+1/2}\right)\,, \tag{C.2}$$

where all symbols are the same as in (3.15)-(3.16), with the newly appearing β is the Courant number in the transversal direction (corresponding to the second spatial index l acquired by $q_{j,l}$),

$$\alpha = \alpha' + \alpha'' = c_x\Delta t/\Delta x \qquad \beta = c_y\Delta t/\Delta y \tag{C.3}$$

As in the case of (3.18), Fourier transform maps the finite difference expressions above into

$$\tilde{q}_{j+1/2,l} - \tilde{q}_{j-1/2,l} = i\mathscr{K}\Delta x \cdot \widehat{q}_{k,m} \cdot e^{ik\Delta x j + im\Delta y l}$$

$$\tilde{q}_{j,l+1/2} - \tilde{q}_{j,l-1/2} = \left(i\mathscr{M}\Delta y + \mathscr{N}\Delta y^4\right) \cdot \widehat{q}_{k,m} \cdot e^{ik\Delta x j + im\Delta y l} \tag{C.4}$$

$$q_{j,l} - q_{j-1,l} = \left(1 - e^{-ik\Delta x}\right) \cdot \widehat{q}_{k,m} \cdot e^{ik\Delta x j + im\Delta y l},$$

where $\mathbf{k} = (k, m)$ is two-dimensional wavenumber; $i\mathscr{M}\Delta y + \mathscr{N}\Delta y^4$ correspond respectively to the imaginary and real parts of $i\mathscr{K}\Delta x$ from (A.6) in the case when of 3rd-order upstream-biased scheme, or the same $i\mathscr{M}\Delta y$ but $N \equiv 0$ in the case of 4th-order centered. A Fourier transform (C.1)-(C.2) become

$$\underbrace{\left[1 + \alpha''\left(1 - e^{ik\Delta x}\right)(1 - 2\gamma)\right]}_{[\,1\,]}\widehat{q}_{\mathbf{k}}^{n+1/2} = \left(\frac{1}{2} - 2\gamma\right)\widehat{q}_{\mathbf{k}}^{n-1} + \left(\frac{1}{2} + 2\gamma\right)\widehat{q}_{\mathbf{k}}^{n} \tag{C.5}$$

$$- (1 - 2\gamma)\left[i\alpha'\mathscr{K}\Delta x + i\beta\mathscr{M}\Delta y\right]\widehat{q}_{\mathbf{k}}^{n}$$

$$\underbrace{\left[1 + \alpha''\left(1 - e^{ik\Delta x}\right)\right]}_{[\,2\,]}\widehat{q}_{\mathbf{k}}^{n+1} = \widehat{q}_{\mathbf{k}}^{n} - \left[i\alpha'\mathscr{K}\Delta x + i\beta\mathscr{M}\Delta y + \beta\mathscr{N}\Delta y^4\right]\widehat{q}_{\mathbf{k}}^{n+1/2}, \tag{C.6}$$

where $[\,1\,]$ and $[\,2\,]$ are simply shorthands for the respective expressions in square brackets. Note that the r.h.s. of the second equation contains $\beta\mathscr{N}\Delta y^4$ while there is no such term in the first equation. This is because the 3rd-order upstream-biased advection is used during corrector stage only – 4th-order centered is used instead during predictor. There is an advantage in this selection resulting in a larger stability limit and more uniform numerical dissipation with respect value of Courant number β (as discussed in Appendix A, Fig. 24). If both stages use the same advection scheme (upstream or centered), the the expressions $\beta\mathscr{N}\Delta y^4$ be present or absent in r.h.ss of both equations accordingly. Another variation of the adaptive algorithm is the use of 2nd-order centered differencing instead of upstream for the implicit α'' terms, hence

$$\frac{q_{j+1,l} - q_{j-1,l}}{2} = i\sin(k\Delta x) \cdot \widehat{q}_{k,m} \cdot e^{ik\Delta x j + im\Delta y l}, \tag{C.7}$$

in place of the last line of (C.4), resulting in

$$[\,1\,] = \left[1 + i\alpha''i\sin(k\Delta x) \cdot (1 - 2\gamma)\right] \qquad \text{and} \qquad [\,2\,] = \left[1 + i\alpha''i\sin(k\Delta x)\right] \tag{C.8}$$

respectively.

Using (C.5) to eliminate $\widehat{q}_{\mathbf{k}}^{n+1/2}$ from (C.6) and substituting $\widehat{q}_{\mathbf{k}}^{n} = \lambda^n e^{i\mathbf{k}\cdot\mathbf{x}}$ leads to the characteristic equation

$$[\,2\,] \cdot \lambda = 1 - \frac{i\alpha'\mathscr{K}\Delta x + i\beta\mathscr{M}\Delta y + \beta\mathscr{N}\Delta y^4}{[\,1\,]} \cdot \left[\left(\frac{1}{2} + 2\gamma\right) - (1 - 2\gamma)\left(i\alpha'\mathscr{K}\Delta x + i\beta\mathscr{M}\Delta y\right)\right]$$

$$- \frac{i\alpha'\mathscr{K}\Delta x + i\beta\mathscr{M}\Delta y + \beta\mathscr{N}\Delta y^4}{[\,1\,]} \cdot \left(\frac{1}{2} - 2\gamma\right) \cdot \lambda^{-1}. \tag{C.9}$$

This is structurally similar to (3.19), especially in the case of when centered differencing is used for the explicit part during both predictor and corrector: $\beta\mathscr{N}\Delta y^4$ term vanishes leaving $i\alpha'\mathscr{K}\Delta x + i\beta\mathscr{M}\Delta y$ pure imaginary as $i\alpha'\mathscr{K}\Delta x$ in (3.19). In this case, in comparison with its one-dimensional counterpart, now α' should be limited in such a way that the sum $\alpha' + \epsilon\beta$ should not exceed the desired $\alpha_{\max}$. [Here $\epsilon = \max\{\mathscr{M}\Delta y\}/\max\{\mathscr{K}\Delta x\} \sim 1$ is a multiplier to account for the different possible maximum values of $\mathscr{M}\Delta y$ and $\mathscr{K}\Delta x$ due to the use of different algorithms in each direction.] This parallels and, in fact, explains the subtraction of horizontal Courant number $\alpha_{\perp i,j,k}$ from $\alpha_{\max}$ before applying it to limit vertical velocity in (3.8) and (3.11).
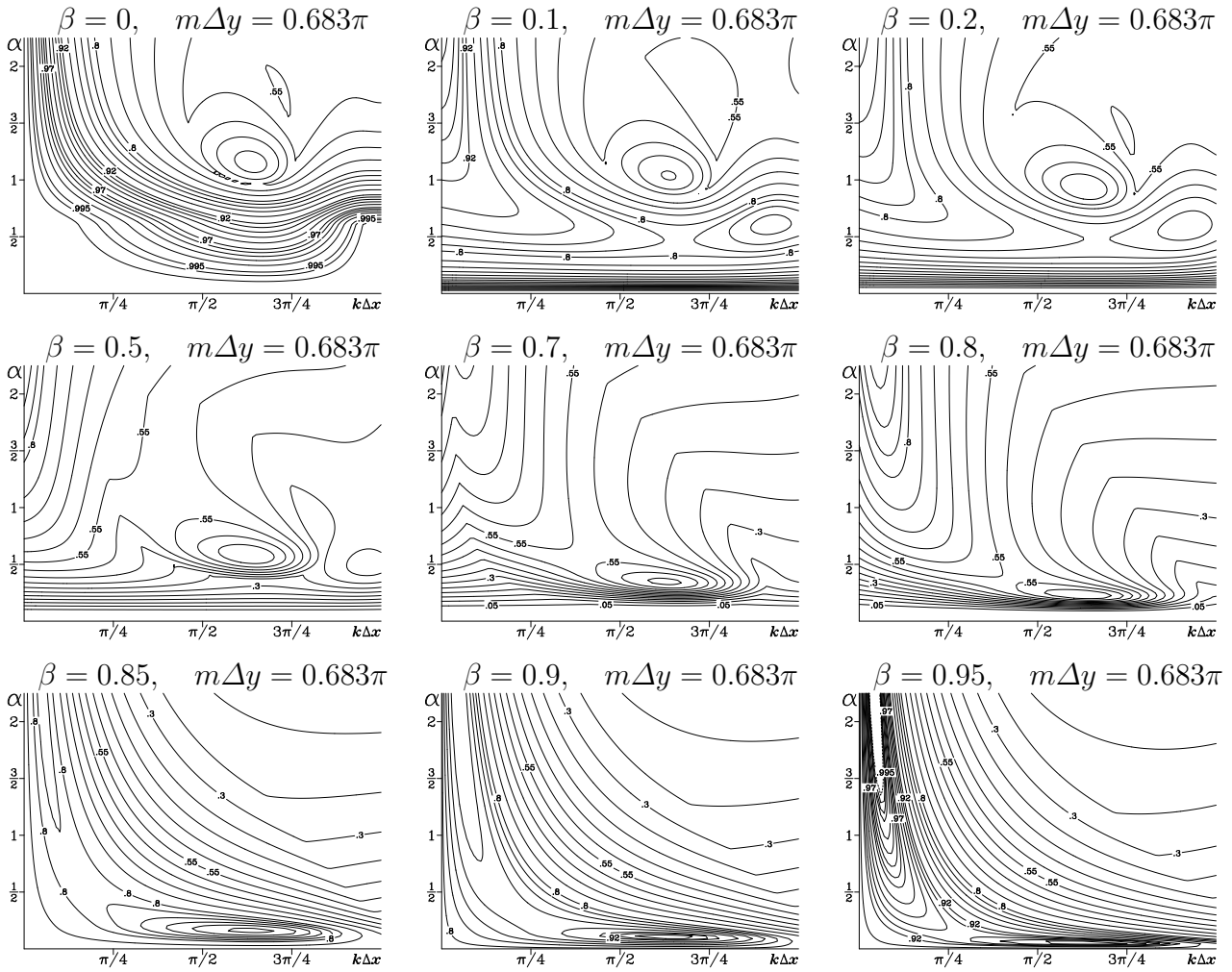
Fig. 26. Amplification per $1\Delta x$ travel $|\lambda|^{(1/\alpha)}$ for characteristic equation (C.9) plotted on $(k\Delta x, \alpha)$-plane for different values of the "horizontal" Courant number β, a single given wavenumber in horizontal direction $m\Delta y = 0.683\pi$ which becomes unstable first once β exceeds its maximum allowed value, and using third-order upstream-biased advection in horizontal direction (corrector stage only, cf. Fig. 24, middle row). The first panel, $\beta = 0$, is identical to Fig. 15, lower-left, which is expected because the threshold values of the limiting algorithm $\alpha_{\max} = 0.6$ and $\alpha_{\max} = 1$ are the same in both cases.

In the case of third-order upstream biased scheme, hence $\mathcal{N}\Delta y^4 \neq 0$, the stability analysis of (C.9) can no longer be reduced to 1D-case of (3.19), and is therefore more complicated as the $(k\Delta x, \alpha)$-parameter space now becomes essentially a 4-dimensional one, $(k\Delta x, \alpha, m\Delta y, \beta)$. To reduce we note that it is sufficient to consider only horizontal wavenumber $m\Delta y$ which becomes unstable first as β increases – it is expected to be in the vicinity of $m\Delta y \sim 0.6$ and $\beta \sim 0.9$ as follows from Fig. 24, middle left. So by varying β and $m\Delta y$ in the vicinity of these values, we find that the instability occurs first when $m\Delta y \approx (0.683 \pm 0.0005)\pi$ and $\beta \approx 0.92$. Then we keep $m\Delta y$ fixed at this setting, and examine behavior of $|\lambda|$ as function of $(k\Delta x, \alpha)$ (covering the entire plane) and slowly varying β within the permissible range. The algorithm is fully-explicit in horizontal direction, so β is expected to have an upper limit. Our goal is to verify that in the algorithm does not impose any additional restriction on Courant number α in the second direction, which is treated adaptively, as long as β stays within the allowed range.

The results are presented in Fig. 26. Note that the third-order upstream-biased scheme results in a strong damping for high wavenumbers, $|\lambda|^{(1/\alpha)} \sim 0.5$ for $m\Delta y = 0.683\pi$, so all the values presented here are expected

51

to be significantly smaller that unity for this reason alone. The tendency of the local maximum observed in all panels here to become closer and closer to abscissa axis (and narrower as well) with increase of β is due to the property of the limiting algorithm (3.11) with β playing the role of $\alpha_\perp$: the horizontal Courant number is subtracted from the maximum allowed vertical before splitting of vertical velocity, hence the lesser fraction vertical velocity is treated explicitly (it vanishes completely when $\beta = \alpha_{\max}$) resulting in a more dissipative algorithm. The property of unconditional stability with respect to α for the entire range of $k\Delta x$ is therefore confirmed for all β presented in Fig. 26 except $\beta = 0.95$ which is beyond the stability limit of horizontal advection. Note that the maximum value within the local maximum near abscissa does not have tendency to grow with increase of β, while instability occurs first for small $k\Delta x$ and large α, with rather sharp growth of $|\lambda|$ as β exceeds 0.9.

Similar studies can be completed for other possible combinations of horizontal (UP3 = 3rd-order upstream biased; C4 = 4th-order centered) and vertical implicit component (UP1 = upstream; C2 = 2nd-order centered, Eq. (3.14), Fig. 16) schemes. Compact 4th-order scheme is used for vertical explicit component in all cases. Given $\alpha_{\min}$, $\alpha_{\max}$, ϵ one determine the maximum horizontal Courant number $\beta_{\max}$ and most critical horizontal wavenumber $m\Delta y$ compare them to their one-dimensional counterparts. Conversely, sensitivity of $\beta_{\max}$ from ϵ leads to optimization of the latter: setting smaller value of ϵ causes increase of the maximum close to horizontal axis in Fig. 26 relative to the other maximum close to vertical axis. The largest possible $\beta_{\max}$ is achieved when the two maxima reach unity at the same time. Examples of optimized parameters are in the table:

Horiz./Impl. vert.	$\alpha_{\min}$	$\alpha_{\max}$	ϵ	$\beta_{\max}$	$m\Delta y$	
UP3/UP1	0.6	1.0	1.0	0.942	0.683π	see Fig. 26
UP3/C2	0.6	0.75	1.0	0.926	0.683π	
C4/UP1	0.6	1.0	0.9	1.110	0.573π	
C4/C2	0.6	0.75	0.8	1.091	0.573π	

Appendix D: Practical algorithm for $W \to W^{(e)} + W^{(i)}$ splitting

Although the procedure of limiting the explicit part and splitting vertical velocity into $W^{(e)}$ and $W^{(i)}$ can be outlined as the sequence (3.9) $\to$ (3.13) $\to$ (3.11) $\to$ (3.10), its translation into an efficient practical algorithm leaves little visual resemblance with these formulae. Therefore we expose it here in more detail.

In the original fully-explicit code computing of $W_{i,j,k+1/2}$ consists of two stages: vertical bottom-up integration of the divergence of horizontal fluxes (*cf.,* Eqs. (1.18)-(1.19) from SM2005) followed by a corrective step to account for the fact that vertical velocity in generalized sigma coordinates is defined relatively to the *moving* grid-box interfaces which follow up-and-down displacement of free surface. The latter step also enforces the exact match in kinematic boundary condition at the surface. The $W \to W^{(e)} + W^{(i)}$ splitting procedure is "implanted" into the existing code without adding a single extra do-loop. Thus, the computation of horizontal Courant number (symbol `CX(i,k)` below) takes place within the first step, while limiting and splitting of full W is combined with the second.

$$\texttt{Wi(i,j,k)} = ... \qquad\qquad\qquad \text{vertical integration of } \mathrm{div}(U,V)$$

$$\texttt{CX(i,k)} = \max\left(U_{i+1/2,j,k}, 0\right) - \min\left(U_{i-1/2,j,k}, 0\right)$$

$$+\max\left(V_{i,j+1/2,k}, 0\right) - \min\left(V_{i,j-1/2,k}, 0\right)$$

$$\texttt{CX(i,0)} = \epsilon \cdot \Delta t / \Delta \mathscr{A}_{i,j} \qquad\qquad \text{just a conversion factor independent of } k$$

```
Wi(i,j,k) = W_{i,j,k+1/2}                          → fully computed W with correct b.c.
if (Wi(i,j,k) > 0) then
  c2d=CX(i,k)   ; dh=H_{i,j,k}                      select values from the grid box above or
else                                                below, upstream from interface W_{i,j,k+1/2}
  c2d=CX(i,k+1) ; dh=H_{i,j,k+1}
endif
cw_max=α_max*dh-c2d*CX(i,0)                         cw_max/dh is the maximum allowed
if (cw_max > 0) then                                           vertical Courant number
  cw_max2=cw_max*cw_max
  cw_min=cw_max*cmnx_ratio
  cw=abs(Wi(i,j,k))*CX(i,0)                         cw/dh is vertical Courant number
  if (cw < cw_min) then
    cff=cw_max2                                     3-way selection, cf., Eq. (3.13)
  elseif (cw < cutoff*cw_max) then
    cff=cw_max2 + r4cmx*(cw-cw_min)**2
  else
    cff=cw_max*cw
  endif
  We(i,j,k)=cw_max2*Wi(i,j,k)/cff                   → W^{(e)}_{i,j,k+1/2}
  Wi(i,j,k)=Wi(i,j,k)-We(i,j,k)                     → W^{(i)}_{i,j,k+1/2}
else
  We(i,j,k)=0                                       Wi(i,j,k) remains unchanged
endif
```

The italicized symbols, $cmnx\_ratio = \alpha_{\min}/\alpha_{\max}$, $cutoff = 2 - \alpha_{\min}/\alpha_{\max}$, $r4cmx = 1/(4 - 4\alpha_{\min}/\alpha_{\max})$, are precomputed constants (via parameter statements) and correspond to the switching threshold values in (3.13). Note that neither Courant number α, nor limiting function $f(\alpha, \alpha_{\max})$ explicitly appear anywhere in the code. There is only a single division per grid point associated with the entire splitting algorithm. It is logically protected to avoid division by zero: cw_max may become non-positive only if the horizontal Courant number becomes too large. In this case vertical advection reverts to implicit backward-Euler time step. Because the horizontal advection algorithm is explicit, its stability limit is not expected to exceed the chosen value of $\alpha_{\max}$ in a significant manner (if at all) but still this protective logic was found to be useful in some computational examples. Alternatively to selecting vertically upstream value for the horizontal contribution to net Courant number, we also try

$$c2d=\max[CX(i,k),CX(i,k+1)] \; ; \; dh=\min[H_{i,j,k}, H_{i,j,k+1}]$$

which is logically more restrictive, but does not lead to a noticeable difference in practice.

References:

Benjamin, T. B., 1968: Gravity currents and related phenomena. *Journal of Fluid Mechanics*, **31**, 209–248.

Birman, V. K., J. E. Martin, and E. Meiburg, 2005: The non-boussinesq lock-exchange problem. part 2. high-resolution simulations. *Journal of Fluid Mechanics*, **537**, 125–144, doi:10.1017/S0022112005005033.

Buzbee, B. L., G. H. Golub, and C. W. Nielson, 1970: On direct methods for solving poisson's equations. *SIAM J. Numer. Analysis*, **7**, 627–656, doi:10.1137/0707049.

Carton, J. A. and B. Giese, 2008: A reanalysis of ocean climate using Simple Ocean Data Assimilation (SODA).

Month. Weath. Rev., **136**, 29993017, `doi:10.1175/2007MWR1978.1`.

Colella, P. and P. R. Woodward, 1984: The Piecewise Parabolic Method (PPM) for gas dynamical simulations. *J. Comput. Phys.*, **54**, 174–201, `doi:10.1016/0021-9991(84)90143-8`.

Crank, J. and P. Nicolson, 1947: A practical method for numerical evaluation of solutions of partial differential equations of the heat conduction type. *Proc. Camb. Phil. Soc.*, **43**, 50–67, republished in *Adv. Comput. Math.*, 1996, Vol. 6, pp. 207-226, `doi:10.1007/BF02127704`.

Dahlquist, G. G., 1963: A special stability problem for linear multistep methods. *BIT*, **3**, 27–43, `doi:10.1007/BF01963532`, ISSN 0006-3835.

De Boor, C., 1978: *A Practical Guide to Splines*. Springer-Verlag, 392 pp.

Griffies, S. M. and A. J. Adcroft, 2008: Formulating the equations of ocean models. *Ocean Modeling in an Eddying Regime, Geophysical Monograph 177*, M. W. Hecht and H. Hasumi, eds., American Geophysical Union, Washington, DC, 281–318.

Haidvogel, D. B. and A. Beckmann, 1999: *Numerical ocean circulation modeling*. Imperial College Press, London, 318 pp.

Hecht, M. W., 2010: Cautionary tales of persistent accumulation of numerical error: Dispersive centered advection. *Ocean Modell.*, **35**, 270–276, `doi:10.1016/j.ocemod.2010.07.005`.

Hirsh, R. S., 1975: Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique. *J. Comput. Phys.*, **19**, 90–109, `doi:10.1016/0021-9991(75)90118-7`.

Hyman, J. M., 1979: A method of lines aproach to the numerical solution of conservation laws. *Advances in computer methods for partial differential equations*, R. Vichnevetsky and R. S. Stepleman, eds., (Pub, IMACS), volume III, 313–321.

Ilıcak, M., A. J. Adcroft, S. M. Griffies, and R. W. Hallberg, 2012: Spurious dianeutral mixing and the role of momentum closure. *Ocean Modell.*, **45-46**, 37–58, `doi:10.1016/j.ocemod.2011.10.003`.

Kanamitsu, M., W. Ebisuzaki, J. Woollen, S.-K. Yang, J. J. Hnilo, M. Fiorino, and G. L. Potter, 2002: NCEP-DOE AMIP-II reanalysis (R-2). *Bulletin of the American Meteorological Society*, **83**, 1631–1643, wOS:000179554000016 `doi:10.1175/BAMS-83-11-1631`, `http://www.esrl.noaa.gov/psd/`.

Klemp, J. B., R. Rotunno, and W. C. Skamarock, 1994: On the dynamics of gravity currents in a channel. *Journal of Fluid Mechanics*, **269**, 169–198, `doi:10.1017/S0022112094001527`.

Large, W., J. C. McWilliams, and S. C. Doney, 1994: Oceanic vertical mixing: a review and a model with a nonlocal boundary layer parameterization. *Rev. Geophys.*, **32**, 363–403, `doi:10.1029/94RG01872`.

Lele, S. K., 1992: Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.*, **103**, 16–42, `doi:10.1016/0021-9991(92)90324-R`.

Lemarié, F., L. Debreu, A. F. Shchepetkin, and J. C. McWilliams, 2012: On the stability and accuracy of the harmonic and biharmonic adiabatic mixing operators in ocean models. *Ocean Modell.*, **52-53**, 9–35, `doi:10.1016/j.ocemod.2012.04.007`.

Leonard, B. P., 1979: A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Comput. Methods Appl. Mech. Eng.*, **19**, 59–98, `doi:10.1016/0045-7825(79)90034-3`.

— 1988: Universal limiter for transient interpolation method of the advective transport equations: The ULTIMATE conservative difference scheme. *NASA Technical Memorandum 100916 ICOMP-88-11*, 115pp.

— 1991: The ultimate conservative difference scheme applied to unsteady one-dimensional advection. *Computer Meth. Appl. Mech. & Engineering*, **88**, 17–74, `doi:10.1016/0045-7825(91)90232-U`.

Leonard, B. P., A. P. Lock, and M. K. McVean, 1996: Conservative explicit unrestricted-time-step constancy preserving advection schemes. *Month. Weath. Rev.*, **124**, 2588–2606, `doi:10.1175/1520-0493(1996)124<2588:CEUTSM>2.0.CO;2`.

Lowe, R. J., J. W. Rottman, and P. F. Linden, 2005: The non-boussinesq lock-exchange problem. part 1. theory and experiments. *Journal of Fluid Mechanics*, **537**, 101–124, `doi:10.1017/S0022112005005069`.

Mason, E., M. J. Molemaker, A. Shchepetkin, F. Colas, J. C. McWilliams, and P. Sangrà,

2010: Procedures for offline grid nesting in regional ocean models. *Ocean Modell.*, **35**, 1–15, doi:10.1016/j.ocemod.2010.05.007.

Orszag, S. A. and M. Israeli, 1974: Numerical simulation of viscous incompressible flows. *Ann. Rev. fluid mechanics*, **6**, 281–318, doi:10.1146/annurev.fl.06.010174.001433.

Richtmyer, R. D. and K. W. Morton, 1967: *Difference methods for initial-value Problems. Second edition. Wiley-Interscience*, 405pp.

Rottman, J. W. and J. E. Simpson, 1983: Gravity currents produced by instantaneous releases of a heavy fluid in a rectangular channel. *Journal of Fluid Mechanics*, **135**, 95–110, doi:10.1017/S0022112083002979.

Rueda, F. J., E. Sanmiguel-Rojas, and B. R. Hodges, 2007: Baroclinic stability for a family of two-level, semi-implicit numerical methods for the 3d shallow water equations. *Int. J. Num. Meth. Fluids*, **54**, 237–268, doi:10.1002/fld.1391.

Shchepetkin, A. F., 2005: If-less KPP. *ROMS/TOMS User Workshop: Adjoint Modeling and Applications*, https://www.myroms.org/Workshops/ROMS2005/Nonlinear/AlexanderShchepetkin.pdf, La Jolla, CA, October 24-26, 2005.

Shchepetkin, A. F. and J. C. McWilliams, 2003: A method for computing horizontal pressure-gradient force in an oceanic model with a nonaligned vertical coordinate. *J. Geophys. Res.*, **108**, 3090, doi:10.1029/2001JC001047.

— 2005: The regional oceanic modeling system (ROMS): A split-explicit, free-surface, topography-following-coordinate oceanic model. *Ocean Modell.*, **9**, 347–404, doi:10.1016/j.ocemod.2004.08.002.

— 2009: Computational kernel algorithms for fine-scale, multi-process, long-term oceanic simulations. *Handbook of Numerical Analysis: Computational Methods for the Ocean and the Atmosphere*, R. Temam and J. Tribbia, eds., Elsevier Science, volume 14, 121–183, doi:10.1016/S1570-8659(08)01202-0.

Shchepetkin, A. F., J. Molemaker, E. M. F. Colas, Y. Uchiyama, J. Kurian, and J. C. McWilliams, 2009: Recent developments of ROMS at UCLA. *Physical Oceanography Review Symposium*, http://www.atmos.ucla.edu/~alex/ROMS/Chicago2009Talk.pdf, Chicago, IL, June 7-13, 2009.

Uppala, S. M., P. W. Kallberg, A. J. Simmons, U. Andrae, V. D. Bechtold, M. Fiorino, J. K. Gibson, J. Haseler, A. Hernandez, G. A. Kelly, X. Li, K. Onogi, S. Saarinen, N. Sokka, R. P. Allan, E. Andersson, K. Arpe, M. A. Balmaseda, A. C. M. Beljaars, L. Van De Berg, J. Bidlot, N. Bormann, S. Caires, F. Chevallier, A. Dethof, M. Dragosavac, M. Fisher, M. Fuentes, S. Hagemann, E. Holm, B. J. Hoskins, L. Isaksen, P. A. E. M. Janssen, R. Jenne, A. P. McNally, J. F. Mahfouf, J. J. Morcrette, N. A. Rayner, R. W. Saunders, P. Simon, A. Sterl, K. E. Trenberth, A. Untch, D. Vasiljevic, P. Viterbo, and J. Woollen, 2005: The ERA-40 re-analysis. *Q. J. R. Meteor. Soc.*, **131**, 2961–3012, doi:10.1256/qj.04.176.

van Leer, B., 1979: Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J. Comput. Phys.*, **32**, 101–136, doi:10.1016/0021-9991(79)90145-1.

Wang, D.-P., 1984: Mutual intrusion of a gravity current and density front formation. *J. Phys. Oceanogr.*, **14**, 1191–1199, doi:10.1175/1520-0485(1984)014<1191:MIOAGC>2.0.CO;2.