The following appendices accompany the article

Dispersal of *Owenia fusiformis* larvae by wind-driven currents: turbulence, swimming behaviour and mortality in a three-dimensional stochastic model

K. Guizien[1,*], T. Brochier[1], J.-C. Duchêne[1], B.-S. Koh[1,3], P. Marsaleix[2]

[1]**Laboratoire d'Océanographie Biologique de Banyuls (CNRS/UPMC), BP 44, 66651 Banyuls-sur-Mer, France**
[2]**Laboratoire d'Aérologie de Toulouse (CNRS/UPS), 14 avenue E. Belin, 31400 Toulouse, France**
[3]**National Fisheries Research and Development Institute, Tidal-flat Research Center, 573-882 Gunsan, Republic of Korea**

[*]**Email: guizien@obs-banyuls.fr**

Marine Ecology Progress Series 311:47–66 (2006)

Appendix 1. Lagrangian versus Eulerian advection-diffusion simulations

A simulation of passive advection-diffusion tracers based on the Lagrangian stochastic approach is compared with the Eulerian approach based on the turbulent viscosity concept. Eulerian simulations consisted of injecting a tracer concentration at 1 mesh point in the southern bay on 18 March 1999 from 12:00 to 23:50 h with a fixed concentration of 1000. The time and spatial evolution of the tracer concentration C is computed over the entire domain solving the following advection-diffusion equation:

$$\frac{\partial C}{\partial t}+\vec{U}.\vec{\nabla}C = \frac{\partial}{\partial x}\left(K_x \frac{\partial C}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_y \frac{\partial C}{\partial y}\right)+\frac{\partial}{\partial z}\left(v_t \frac{\partial C}{\partial z}\right) \quad \text{(A1)}$$

where K_x,K_y are the horizontal and vertical turbulent diffusivities, v_t is the vertical one, t is time, and x, y, z are the 3D space variables. Eq. (A1) is solved using an upwind discretization scheme for horizontal advection terms to avoid numerical instabilities while reducing computation durations. Using an upwind scheme is the same as using a centred scheme for advection terms to which are added 'numerical' horizontal diffusion terms with diffusivities of the form $K_x = u\Delta x/2$ and $K_y = v\Delta y/2$. For a 100 m horizontal grid size (Δx or Δy) and with typical horizontal current (u or v) value of 0.2 m s^{-1}, 'numerical' diffusivity is about 10 m^2 s^{-1}, which is already a large value. Thus, no extra horizontal diffusion is added in Eq. (A1) and we take $K_x = K_y = 0$.

Lagrangian simulations consisted in releasing a number N of passive particles, i.e. the individual larva's velocity $\vec{u}_{sp}(\vec{X},t) = 0$ with $X = (x, y, z)$ the 3D space location, every 10 min, from the mesh point where the concentration was injected and over the same period. Their dispersal is computed solving Eq. (1) with and without turbulence. N should be large enough to ensure stable dispersal patterns. Preliminary tests showed that dispersal patterns obtained for $N = 10$ and $N = 30$ are very similar. Thus, N was set to 30. Finally, concentration fields are derived from particles positions by weighting the 8 neighbouring mesh points according to their distance from the particle.

Fig. A1 shows concentration fields obtained by solving Eq. (A1) (Eulerian, Fig. A1a) at various times within 3 d after the injection began. In Fig. A1b, the particle positions are plotted at the same times, obtained by solving the Lagrangian Eq. (1) without (black) and with (grey) turbulence. Fig. A1c shows concentration fields derived from particle positions at the same times. First, these computations clearly show that dispersal is dramatically enhanced by turbulence and that turbulence should be accounted for in Lagrangian dispersal models (Fig. A1b). Second, concentration fields assuming isotropic turbulence for Lagrangian dispersal agree rather well with the Eulerian concentration fields, although a stronger horizontal smearing occurs in the Eulerian computations. Indeed, the 'numerical' horizontal diffusivity, associated with the low time consuming upwind numerical scheme is larger than vertical diffusivity. This explains the discrepancies between the isotropic Lagrangian simulations and the Eulerian ones. It can be assumed that such 'numerical' horizontal diffusivity without physical grounds would differ from the effective physical horizontal diffusion. However, up to now, no field measurements of turbulence have been available to discriminate between the 2 computational results.

Although the Lagrangian approach can be limited by data storage requirements, it is expected to describe advection-diffusion dispersal more realistically, avoiding the extra computing cost of minimizing numerical diffusion in the discretisation of advection terms in the Eulerian approach.

Appendix 1 (continued)

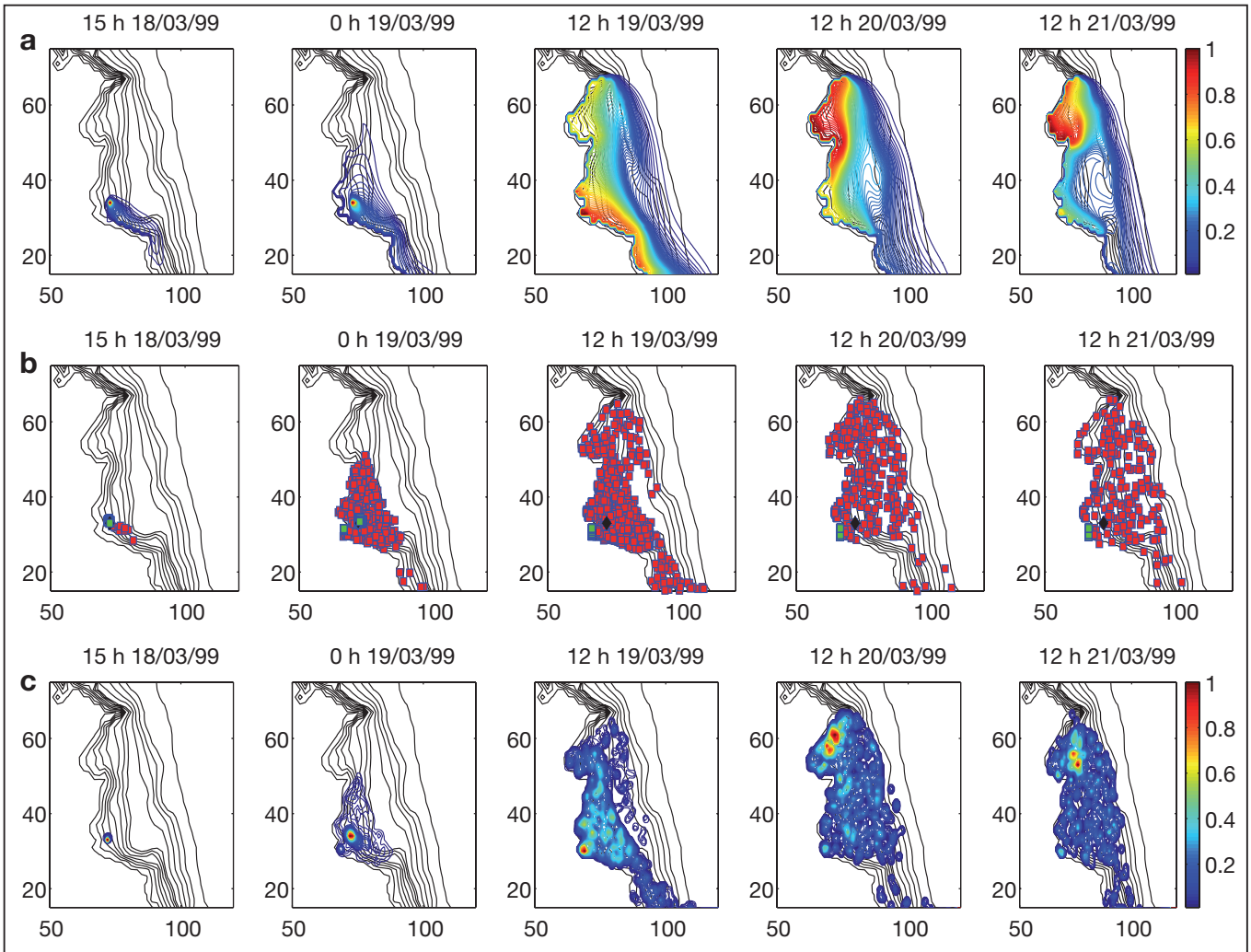

Fig. A1. (a) Time evolution of Eulerian advected-diffused concentration; (b) Lagrangian particle dispersals with isotropic turbulence in red and without in green; ♦: spawning locations; (c) concentration derived from Lagrangian particle dispersals. Particles and concentration are released from the same point from 12:00 to 23:50 h on 18 March 1999. Concentrations are non-dimensional, ranging between 0 and 1. Dates are d/mo/yr

Appendix 2. Open boundaries and nesting procedure

Open boundaries conditions

Let us consider 2 lateral open boundaries located at $x = 0$ and $x = L$ ($0 < L$) where L is the width of the computational grid between the 2 open boundaries. The boundary condition suggested by Oey & Chen (1992) gives a relation between the sea surface elevation η and the current transport normal to the boundary $\bar{u}$:

$$\left(\bar{u} - \bar{u}_F\right) = \pm\sqrt{gH}\,(\eta - \eta_F)$$

where g is gravity and H the water depth. The right hand side term is preceded by + (or –) for $x = L$ (or $x = 0$). The index F stands for the external forcings that can be derived from a larger grid or a climatology. A zero gradient condition is applied to the current transport tangent to the boundary and to baroclinic mode components. External temperature and salinity fields are advected into the numerical domain under inflow conditions.

These conditions applied on boundary nodes are combined with restoring terms added to the right-hand side of the momentum, temperature and salinity equations inside the domain over a restoring layer. Next to the $x = 0$ boundary, these restoring terms read:

$$\frac{\varphi_F - \varphi}{T_{res}} e^{-x/D}$$

where φ stands either for temperature, salinity, baroclinic velocities or transports and D is the restoring layer horizontal scale. The exponential decay scale factor depends on the grid size, typically $D = L/50$. The time scale factor is the time required for waves to cross the numerical domain, that is $T_{res} = L/c$ where c is the mean wave celerity. For barotropic modes, $c = \sqrt{gH}$ and for baroclinic modes, we take $c = 1$ m s^{-1} as a first guess that can be empirically adjusted later. The exponential decay and time scale factor used for the restoring layer in nested grids are given in Table A1.

Nested meshes

As detailed above, open boundary conditions combine model variables and external forcing terms. In the nesting procedure used here, these external forcing terms are provided by another simulation using a larger, coarser grid without feedback from the finer grid to the coarser one, which is a 1-way strategy. The largest grid is run first, periodical outputs being stored and then used to constrain the run of a nested grid. Due to grid discrepancies, largest grid outputs have to be interpolated on the finer grid. It is stressed that the interpolation of the external forcings not only concerns boundary nodes but also the restoring layer and initial state. In our computations, we use a linear interpolation scheme although it has been shown that this simple scheme can produce numerical instabilities at the initial state or near open boundaries if bottom slopes differ significantly from one grid to another (Auclair et al. 2000). Hence, with a linear interpolation scheme, the success of the nesting procedure will depend on grid matching, and thus the nested grid bathymetry and coastline must be carefully specified.

Practically, the nested grid bathymetry is constrained to be close to the larger one, especially where external forcings play a role, that is, in the restoring layer and at the open boundaries. Considering the $x = 0$ boundary, nested bathymetry h_{nested} is given by:

$$h_{nested} = (1 - e^{-x/D})h_{fine} + e^{-x/D}h_{coarse}$$

where h_{fine} is a first guess of the bathymetry and h_{coarse} its corresponding value in the coarser grid and D is the restoring layer horizontal scale.

High frequency variability at open boundaries is highly dependent on the time renewal of forcing terms (the smaller the better) but practical data storage constraints may lead to some compromises. Since high frequency variability processes on the modelled area are mainly near inertial motions and meteorologically induced circulations, the period of outputs storage T_{sto} is given by:

$$T_{sto} = \min(2\pi/Pf,\ T_{meteo})$$

where T_{meteo} is the time renewal of the meteorological forcings, f is Coriolis parameter and P a factor that controls the accuracy of inertial oscillations time sampling ($p \geq 4$). In our computations, we use the 3-hourly averaged meteorological fields outputs of the ALADIN weather forecast model (METEO FRANCE) and $P = 6$, which lead to $T_{sto} = 3$ h. Stored fields are averaged over T_{sto}. Finally, they are linearly interpolated in space and time to drive the nested grid.

Table A1. Nested computational grids resolution and size (numbers of points), horizontal and time scales for restoring layers

Grid	Horizontal cell size (m)	Number of vertical levels	Grid size	Restoring layer horizontal scale (m)	Barotropic restoring time scale (d)	Baroclinic restoring time scale (d)
Grid 1	1500	26	188 × 145	4500	0.1	0.1
Grid 2	700	21	82 × 150	2100	0.02	0.1
Grid 3	250	19	94 × 140	750	0.05	0.5
Grid 4	100	16	160 × 179	300	0.01	0.1