INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids (2013)
Published online in Wiley Online Library (wileyonlinelibrary.com/journal/nmf). DOI: 10.1002/fld.3774

Numerical diffusion for flow-aligned unstructured grids with application to estuarine modeling

Rusty Holleman[1,*,†], Oliver Fringer[2] and Mark Stacey[1]

[1] *Civil and Environmental Engineering, University of California, Berkeley, CA, USA*
[2] *Environmental Fluid Mechanics Laboratory, Stanford University, Stanford, CA, USA*

SUMMARY

The benefits of unstructured grids in hydrodynamic models are well understood but in many cases lead to greater numerical diffusion compared with methods available on structured grids. The flexible nature of unstructured grids, however, allows for the orientation of the grid to align locally with the dominant flow direction and thus decrease numerical diffusion. We investigate the relationship between grid alignment and diffusive errors in the context of scalar transport in a triangular, unstructured, 3-D hydrodynamic code. Analytical results are presented for the 2-D anisotropic numerical diffusion tensor and verified against idealized simulations. Results from two physically realistic estuarine simulations, differing only in grid alignment, show significant changes in gradients of salinity. Changes in scalar gradients are reflective of reduced numerical diffusion interacting with the complex 3-D structure of the transporting flow. We also describe a method for utilizing flow fields from an unaligned grid to generate a flow-aligned grid with minimal supervision. Copyright © 2013 John Wiley & Sons, Ltd.

Received 29 August 2011; Revised 15 November 2012; Accepted 1 January 2013

KEY WORDS: unstructured grid; numerical diffusion; grid generation; grid alignment

1. INTRODUCTION

Unstructured grid models are used in a wide variety of computational fluid dynamics applications, including extensive use in hydrodynamics [1–3]. Unstructured models allow seamless transitions from large scales to small and efficient adaptation to complex domain geometry. Along with these advantages come challenges related to grid generation, higher-order methods, and error analysis. In particular, many higher-order advection schemes are not directly applicable to unstructured grids, leaving these models to use diffusive low-order advection schemes. Characterizing this numerical diffusion and finding ways to reduce its impact on global model error are important facets of utilizing unstructured models.

A standard mathematical tool for evaluating discretization errors is the modified equation analysis [4, 5], which can be applied to structured grid methods with relative ease. This analysis is still possible in the case of unstructured grids, but irregular stencils and coupling between coordinate dimensions complicate matters. Special care must be taken to account for heterogeneities in the grid and the fact that neighboring cells may not be characterized by the same modified equation and may not be independently consistent with the original PDE being modeled. For example, Bouche [6] investigated the upwind finite-volume method applied to the scalar advection equation and observed an apparent loss of consistency when analyzing the method on unstructured grids. Even discretizations on structured but nonuniform grids can lead to inconsistency, such as

*Correspondence to: Rusty Holleman, Civil and Environmental Engineering, University of California, Berkeley, CA, USA.
†E-mail: holleman@berkeley.edu

the nonuniform curvilinear meshes studied by Turkel [7]. The author observed that irregular meshes led to inconsistent discretizations if the spacing was not quasi-uniform and the staggering placed the dependent variable at cell centers. In the specific case of finite-volume, first-order upwind on triangular unstructured grids, the focus of this manuscript, these heterogeneities arise on a uniform triangular grid because of asymmetries between two classes of cells, distinguished by the number of inflow faces.

In practice, first-order accurate advection schemes are often too diffusive to accurately predict scalar gradients [8]. Although there do exist higher-order methods that are commonly implemented on unstructured grids, such as total variation diminishing and flux-corrected transport, these methods are generally formulated as corrections or extensions to an underlying low-order method [9]. Monotonic higher-order schemes are necessarily nonlinear, as Godunov's theorem states that a linear monotone numerical scheme is at most first-order accurate [10]. This nonlinearity makes a generalized error analysis difficult, and for that reason, we choose to investigate the diffusive behavior of the underlying low-order method, for which a complete analysis is possible in the case of a uniform mesh.

The bulk of analysis for numerical methods is focused on the following: (i) consistency (with particular attention paid to the order of the error terms) and (ii) stability. Together, consistency and stability imply convergence by way of the Lax equivalence theorem [11], and for some applications, it is sufficient to identify only the order of the error terms and the presence of stability, without regard to the coefficients on the error terms. Several aspects of hydrodynamic applications, however, call for a more quantitative analysis of numerical diffusion. Firstly, hydrodynamic applications often have well-defined principal velocity directions, whether due to river flows, prevailing winds, or the major axis of the tidal ellipse. This attribute of marine flows motivates the desire to understand not just the worst-case behavior of an advection operator but also the expected behavior given a specific velocity distribution. Additionally, hydrodynamic applications often need to distinguish between lateral, or across-flow, diffusion and longitudinal, or streamwise, diffusion. Secondly, oceanographers are often interested in quantifying physical mixing because of processes ranging from small-scale turbulent diffusion to diffusive processes arising from the evolution of the structure of the large-scale flow. Nonphysical diffusive errors are not just a frustration but significantly raise the noise floor when the ultimate goal is to quantify diffusion arising from physical processes.

Both physical and numerical diffusions can interact with spatial variations in the velocity field to modify large-scale mixing and transport. For instance, shear in the velocity field will cause the spreading of a scalar distribution as fluid parcels sample different portions of the velocity profile. Diffusion perpendicular to the flow will cause a single parcel to sample a broader swath of the velocity profile. In the limit of infinite lateral diffusion, all parcels will sample all parts of the velocity profile equally, such that the mean transport of each parcel is identical and the distribution of parcels in the direction of the straining will remain constant. This interaction of shear and lateral diffusion where longitudinal diffusion is inversely proportional to lateral diffusion (i.e., shear dispersion [12]) is an example of the potentially hidden effects of numerical diffusion and motivates the need to differentiate lateral and longitudinal numerical diffusions.

There have been many methods proposed for reducing numerical diffusion in unstructured finite-volume schemes while retaining monotonicity. Total variation diminishing methods for 1-D problems [13] are naturally extensible to multidimensional structured methods by way of operator splitting and have more recently been extended to unstructured multidimensional problems (e.g., [14, 15]). Another approach, the reservoir method [16], decreases numerical diffusion by time-shifting fluxes to achieve an effective local Courant number near unity. Grid alignment has long been known to affect the accuracy of results, which has resulted in a number of approaches for adapting a grid to flow features. Adapting a grid to the flow may occur before the simulation, such as in applications focused on resolving boundary layers that use a static grid, which envelopes boundaries with layers of highly anisotropic, boundary-oriented cells (such as the advancing layers method discussed in [17]), and the methods described in the current manuscript. Many adaptive methods rely on runtime error estimates to target areas for refinement, such as the methods described in [18] applied to quadtree and octree meshes. Adaptive methods for fully unstructured grids such as those of [19] are able to apply anisotropic stretching and refining of grid elements to better resolve oriented

DOI: 10.1002/fld

features of a flow. Another perspective on adaptive grid methods is the Vofire method of [20] in which triangular cells are subdivided parallel to the local velocity direction at each step, decreasing the cross-flow component of numerical diffusion. These runtime adaptive methods require significant modifications to or complete re-implementation of the numerical core of a simulation code. Additionally, because of the nonlinear character of most of the schemes, numerical diffusion is both difficult to predict *a priori* and may be either positive or negative (such as the oversharpening of gradients seen with the superbee limiter [15]).

For an important class of finite-volume C-grid hydrodynamic solvers (e.g., [1, 2]), the computational grid must be orthogonal, meaning that the line joining adjacent cell centers must be perpendicular to their common edge, which is equivalent to requiring that cell centers are the circumcenters of the cells. Additionally, they require that the cell center falls within the cell itself, a condition that is violated for triangular cells that are not acute. This constraint both limits the utility of anisotropic mesh approaches and makes regridding generally too expensive to be performed dynamically.

Section 2 presents the details of the modified equation analysis for the linear advection equation on an idealized triangular grid. Using a first-order upwind, finite-volume discretization of the scalar advection equation, we first analyze numerical diffusion in each of the two classes of grid cells independently and demonstrate that such an independent analysis is insufficient to capture the coupling of errors between the two classes of cells. Consequently, the analysis is extended to consider the aggregate errors due to the interaction of the two classes of cells, leading to analytical expressions for the components of the 2-D anisotropic numerical diffusion tensor. Section 3 introduces a series of idealized scalar advection test cases from which we calculate empirical numerical diffusion coefficients. These empirical estimates are shown to verify the analytical expressions derived in Section 2. Section 4 presents an algorithm for generating flow-aligned grids and a simple test case applying this algorithm and evaluating the effects on numerical diffusion in the test case. Section 5 describes a physically realistic pair of simulations of San Francisco Bay, where the first case uses an unaligned grid and the second employs a grid with cells aligned to the principal velocity directions. Comparisons of salinity model data with observations show significant improvements in salinity gradients when using the aligned grid, consistent with decreased nonphysical lateral diffusion. Although the first-order upwind advection scheme utilized in these simulations is not state of the art, it is effective at highlighting the role of grid alignment in numerical diffusion and the effect of anisotropic numerical diffusion on scalar distributions, both of which are present in higher-order advection schemes.

2. ANALYTICAL GRID DIFFUSION

As a point of reference for the modified equation analysis [5], we briefly consider the modified equivalent PDE for 1-D, first-order, forward-in-time, backward-in-space scalar advection. This scheme can be written as

$$A_j^{n+1} = CA_{j-1}^n + (1-C)A_j^n \tag{1}$$

where A_j^n is the scalar concentration in cell j and time step n and $C = U\Delta t/\Delta x$ denotes a volumetric Courant number, equivalent to the fraction of each cell's mass displaced by advective fluxes during each time step. A Taylor series expansion of all scalar values about time n and location j leads to the modified equivalent equation

$$A_t + UA_x = \frac{\Delta x^2}{2\Delta t}C(1-C)A_{xx} + O(\Delta x^2, \Delta t^2) \tag{2}$$

where subscripts denote the corresponding partial derivative and the coefficient on A_{xx} is interpreted as numerical diffusion. The stability of this scheme can be related to the sign of the diffusive term, leading to the conclusion that this method is stable for $0 \leqslant C \leqslant 1$.

In the context of 2-D structured grids, this analysis may be applied directly, but a difficulty arises in applying a modified equation analysis to an unstructured triangular grid, even when the flow is

DOI: 10.1002/fld

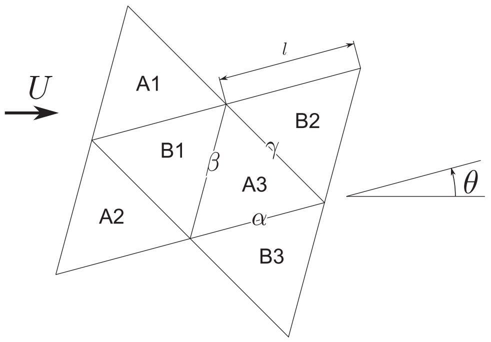

Figure 1. Schematic for derivation of recurrence relation.

uniform and the grid regular. The key hurdle to direct application of modified equation analysis is the fact that for a uniform and constant flow, half of the cells have one inflow face and two outflow faces (type A cells in Figure 1), whereas the other half have two inflow faces and one outflow face (type B cells). The direct, decoupled analysis is illustrated in the following section in which we derive the truncation error independently for type A and B cells. The result is unrealistic but motivates an extension of the analysis that considers the coupling of type A and B cells.

In the following derivation, following the schematic in Figure 1, the scalar concentration at time step n is denoted A_i^n for type A cells and B_i^n for type B cells, to reinforce the notion that the two scalar fields are not interchangeable. Scalar values are cell averages and, for the purpose of gradients, are located at the circumcenter of each cell. The orientation of the grid relative to the horizontal is given by $0 \leqslant \theta \leqslant \pi/3$, and the three types of edges are denoted by α, β, and γ. The free-stream velocity U is taken parallel to the x-axis, and the components of the velocity normal to the faces are given by $U_\alpha = U \sin\theta$, $U_\beta = U \sin(\theta + \pi/3)$, and $U_\gamma = U \sin(\theta + 2\pi/3)$. Because the grid is congruent over rotations of $n\pi/3$, the given range of θ is sufficient to cover all possible orientations of the grid relative to the flow. The edge length is l, and the cell area is a, which for the case of equilateral triangles is $a = (\sqrt{3}/4)l^2$.

2.1. Independent analysis of A and B cells

We discretize the scalar advection equation for an axis-aligned flow, $A_t + UA_x = 0$, for a type A cell using a first-order upwind, finite-volume discretization:

$$a\left(A_3^{n+1} - A_3^n\right) = l\Delta t\left(U_\beta B_1^n - U_\alpha A_3^n - U_\gamma A_3^n\right) \tag{3}$$

Defining a finite-volume analog to the Courant number for each face $C_\xi = U_\xi l\Delta t/a$ and noting that, from continuity, $C_\beta = C_\alpha + C_\gamma$, we rewrite (3) as

$$A_3^{n+1} = \left(1 - C_\beta\right) A_3^n + C_\beta B_1^n \tag{4}$$

Applying Taylor series expansions in time and space, centered around A_3^n, we find the modified equation for a type A cell:

$$\begin{aligned} A_t &+ \frac{4}{3}\cos^2\left(\theta - \frac{\pi}{6}\right) UA_x + \frac{4}{3}\cos\left(\theta - \frac{\pi}{6}\right)\sin\left(\theta - \frac{\pi}{6}\right) UA_y \\ &= \left[\frac{2}{3\sqrt{3}}\cos^3\left(\theta - \frac{\pi}{6}\right) Ul - \frac{1}{2}\Delta t U^2\right] A_{xx} \\ &\quad + \frac{2Ul}{3\sqrt{3}}\cos\left(\theta - \frac{\pi}{6}\right)\sin^2\left(\theta - \frac{\pi}{6}\right) A_{yy} \\ &\quad + \frac{4Ul}{3\sqrt{3}}\cos^2\left(\theta - \frac{\pi}{6}\right)\sin\left(\theta - \frac{\pi}{6}\right) A_{xy} + O\left(\Delta t^2, l^2\right) \end{aligned} \tag{5}$$

DOI: 10.1002/fld

The analysis of type B cells follows in the same manner, with the resulting modified equation:

$$\begin{aligned}
B_t &+ \frac{4}{3}\left[\sin^2\left(\theta+\frac{2\pi}{3}\right)+\sin^2\theta\right]UB_x+\frac{2}{3}\left[\sin\left(2\theta+\frac{\pi}{3}\right)-\sin\left(2\theta\right)\right]UB_y \\
&= \left[\frac{2l}{3\sqrt{3}}\left(\sin^3\left(\theta+\frac{2\pi}{3}\right)+\sin^3\theta\right)-\frac{1}{2}U\Delta t\right]UB_{xx} \\
&+ \frac{2l}{3\sqrt{3}}\left(\cos\left(\theta+\frac{\pi}{6}\right)\sin^2\left(\theta+\frac{\pi}{6}\right)+\cos^2\theta\sin\theta\right)UB_{yy} \\
&+ \frac{4Ul}{3\sqrt{3}}\left[\cos^2\left(\theta+\frac{\pi}{6}\right)\sin\left(\theta+\frac{\pi}{6}\right)-\cos\theta\sin^2\theta\right]B_{xy}+O\left(\Delta t^2, l^2\right) \qquad (6)
\end{aligned}$$

With the coefficients on A_x and B_x denoted as the effective advective speeds U_A and U_B and the coefficients of A_{xx}, A_{yy}, and A_{xy} as numerical diffusivities K_A^x, K_A^y, and K_A^{cross} (and similarly for B cells), the modified equations for each type of cell have an apparent loss of consistency, due to $U_A \neq U_B \neq U$. Similar observations of loss of consistency have been made for discretizations on irregular structured grids (e.g., [7]). However, in this regular, unstructured case, the expected advective speed U can be recovered by assuming a simple average of the coefficients in (5) and (6), which also allows the cross-stream advection coefficients on A_y and B_y to cancel. Similarly, averaging the diffusion coefficients and simplifying give

$$K_{\text{mean}}^x = \left[\frac{1}{3\sqrt{3}}\sin^3\theta+\frac{1}{4}\cos\theta\right]Ul-\frac{1}{2}U^2\Delta t \qquad (7)$$

$$K_{\text{mean}}^y = \left(\frac{1}{3\sqrt{3}}\cos^2\theta\sin\theta+\frac{1}{12}\cos\theta\right)Ul \qquad (8)$$

$$K_{\text{mean}}^{\text{cross}} = \left(\frac{1}{6}-\frac{2}{3\sqrt{3}}\cos\theta\sin\theta\right)\sin\theta Ul \qquad (9)$$

Whereas the cross-stream advection coefficients cancel, coefficients on $\partial/\partial x^2$ and $\partial/\partial y^2$ in the Taylor series expansion are always positive and do not cancel, such that the resulting diffusion coefficients will overpredict both lateral and longitudinal diffusions. Note that the cross term only provides information on the orientation of diffusion; the magnitude of the total diffusion depends only on K^x and K^y.

This result is intuitively evident from examining a strip of triangular cells, parallel to the flow, such as in Figure 2. Cell centers of the two types of cells are offset from each other in the cross-flow

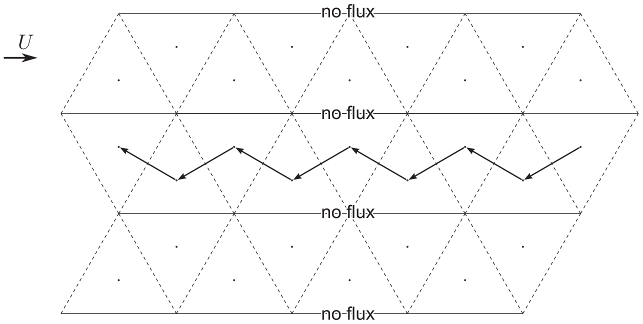

Figure 2. Schematic showing vanishing of lateral diffusion for $\theta = 0$. Solid lines denote zero-flux faces of the cells, and dashed edges have nonzero flux. The arrows along the center strip show the vector $(\Delta x, \Delta y)$ for which the Taylor series expansion is evaluated for each cell's upwind neighbor. The nonzero y component of the vectors leads to an overpredicted K^y unless the two types of cells are treated simultaneously.

direction, such that the Taylor series expansion in the lateral direction includes positive coefficients on A_{yy} and B_{yy}. The positive coefficients lead directly to a nonzero K^y in the averaged modified equation. Geometrically, though, any diffusive spreading is limited to this single-width triangle strip, as the flow-aligned faces have zero flux. The source of this discrepancy is that a Taylor series expansion about one type of cell ignores the biases in the other type of cell.

2.2. *Combined analysis*

One method for amending the analysis is to expand the values B_i^n for type B cells in terms of values in type A cells, using a recurrence relation rather than a Taylor series expansion. For the schematic in Figure 1, the update equation for cell B_1 at time step n, analogous to (4) but for a type B cell, is

$$B_1^n = \left(1 - C_\alpha - C_\gamma\right) B_1^{n-1} + C_\alpha A_1^{n-1} + C_\gamma A_2^{n-1} \tag{10}$$

For convenience, we define $\lambda \equiv 1 - C_\alpha - C_\gamma$. Successive application of the recurrence relation replaces B_1^{n-j} with an expression containing B_1^{n-j-1} and A_i^{n-j-1}, such that after k expansions, we obtain

$$B_1^n = \lambda^k B_1^{n-k} + C_\alpha \sum_{j=1}^{k} A_1^{n-j} \lambda^{j-1} + C_\gamma \sum_{j=1}^{k} A_2^{n-j} \lambda^{j-1} \tag{11}$$

The upwinded values from A_i at past time steps can be expanded as a Taylor series in time around step n, yielding A_i^{n-j} in terms of A_i^n and its time derivatives. Because the purpose of the analysis is to quantify numerical diffusion, we have limited the expansion to the second derivative. Assuming an n large enough that B_1^n is fully decoupled from the initial condition B_1^0, that is, $n\Delta t \gg l/U$ and $\lambda^n \approx 0$, we allow k to go to infinity resulting in

$$\begin{aligned} B_1^n = \sum_{j=0}^{\infty} \lambda^j \Big[& C_\alpha A_1^n + C_\gamma A_2^n - (j+1)\,\Delta t \left(C_\alpha A_{1,t}^n + C_\gamma A_{2,t}^n\right) \\ & + \frac{(j+1)^2\,\Delta t^2}{2} \left(C_\alpha A_{1,tt}^n + C_\gamma A_{2,tt}^n\right) + O\left(\Delta t^3\right) \Big] \end{aligned} \tag{12}$$

for which the infinite sums can be replaced with closed expressions:

$$\begin{aligned} B_1^n = & \frac{1}{1-\lambda}\left[C_\alpha A_1^n + C_\gamma A_2^n\right] - \frac{\Delta t}{(1-\lambda)^2}\left[C_\alpha A_{1,t}^n + C_\gamma A_{2,t}^n\right] \\ & + \frac{(1+\lambda)\,\Delta t^2}{2\,(1-\lambda)^3}\left[C_\alpha A_{1,tt}^n + C_\gamma A_{2,tt}^n\right] + O\left(l^3\right) \end{aligned} \tag{13}$$

This expression for B_1^n is now in terms of values at time step n evaluated only at the centers of type A cells. For the purposes of the modified equation analysis, A_i^n and corresponding derivatives are then expressed as Taylor series expansions around A_3^n and its derivatives. After substitution, we obtain the modified equation (now dropping the subscript on A_3, as all values have been expanded about this point)

$$A_t + U_{\text{eff}} A_x = \left[2K^x\right] A_{xx} + \left[2K^y\right] A_{yy} + \left[2K^{\text{cross}}\right] A_{xy} + O\left(\Delta t^2, l^2\right) \tag{14}$$

The simplified expressions (see Appendix A for details of the simplification) for the anisotropic numerical diffusion coefficients are

$$2K^x = \left[\frac{2\sqrt{3}}{3} \left(\cos^2\theta \cos\left(\theta + \frac{\pi}{6}\right) + \sin\theta \cos^2\left(\theta - \frac{\pi}{3}\right) \right) - \frac{\sqrt{3}}{4\sin\left(\theta + \frac{\pi}{3}\right)} \right] Ul - U^2 \Delta t \tag{15}$$

$$2K^y = \frac{\sin(3\theta)}{2\sqrt{3}} Ul \tag{16}$$

$$2K^{\text{cross}} = \left[\frac{4}{\sqrt{3}} \sin\theta \sin\left(\theta - \frac{\pi}{6}\right) \sin\left(\theta - \frac{\pi}{3}\right) \right] Ul \tag{17}$$

$$U_{\text{eff}} = U$$

The factor of 2 arises from the fact that the analysis includes truncation errors both for the update of a B cell from its upwind A neighbors and for an A cell updated from its upwind B neighbor. The effective mean numerical diffusion coefficients are consequently one-half the coefficients that appear in the modified equation.

For the special case of a flow-aligned grid, $\theta = 0$, and the longitudinal numerical diffusion simplifies to

$$K^x = \frac{1}{4} Ul - \frac{1}{2} U^2 \Delta t \tag{18}$$

Employing the fact that the longitudinal spacing of cell centers is $\Delta x = l/2$ when $\theta = 0$ and the Courant number $C = U\Delta t/\Delta x$, we can further simplify the longitudinal diffusion to $K_x = (1/2)\left(U\Delta x - U^2\Delta t\right) = (\Delta x^2/2\Delta t)C\,(1 - C)$. Thus, in the case of an exactly aligned flow, the unstructured method reduces to the same behavior as a 1-D structured grid. Likewise, the lateral diffusion coefficient goes to zero for an aligned grid, consistent with the schematic of Figure 2.

Because the update equation in the aligned grid case is identical to that for a regular 1-D discretization, the same conditions apply for stability, namely that $0 \leqslant C \leqslant 1$. For arbitrary θ, the condition for monotonicity, which is a sufficient condition for stability [9], is

$$0 \leqslant C_\beta = \frac{4U\Delta t \sin\left(\theta + \frac{\pi}{3}\right)}{\sqrt{3}\, l} \leqslant 1 \tag{19}$$

The worst-case scenario, which occurs at $\theta = \pi/6$, requires $0 \leqslant U\Delta t/l \leqslant \sqrt{3}/4$. Stability may also be investigated in terms of the diffusion coefficients. Because stability is related to the fastest-growing Fourier mode, which may be anisotropic and not necessarily aligned with the axes, the 2-D case requires consideration of the signs of the eigenvalues of the diffusion tensor

$$\underline{\underline{K}} = \begin{bmatrix} K^x & \frac{1}{2}K^{\text{cross}} \\ \frac{1}{2}K^{\text{cross}} & K^y \end{bmatrix} \tag{20}$$

rather than the total diffusion $K^x + K^y$ or the separate axis-aligned diffusion coefficients K^x and K^y. The condition that the smallest eigenvalue of $\underline{\underline{K}}$ be nonnegative can be shown to be equivalent to the monotonicity condition (19), although we omit the lengthy and generally unenlightening algebra from this manuscript.

For comparison, we include in the following the diffusion coefficients for the same first-order upwind advection scheme on a regular Cartesian grid. The discretization is

$$A^{n+1} = (1 - C_W - C_N)\, A^n + C_W A_W^n + C_N A_N^n \tag{21}$$

where subscripts W and N denote the respective quantities for the upwind cells to the 'west' (negative x-direction when $\theta = 0$) and 'north' (positive y when $\theta = 0$) directions. The Courant numbers are defined as

$$C_N = \sin\theta \frac{U\Delta t}{l} \tag{22}$$

$$C_W = \cos\theta \frac{U\Delta t}{l} \tag{23}$$

where θ takes on the same sense as in the triangular case, namely a counterclockwise rotation of the grid relative to a $+x$-directed mean flow. The modified equation analysis follows the same conventions as for the triangular grid, although with only a single type of cell, the derivation is straightforward and as such has been omitted for brevity. The anisotropic numerical diffusion coefficients for a Cartesian grid then read

$$K^x = \frac{1}{2}\left[\sin^3\theta + \cos^3\theta\right] Ul - \frac{1}{2}U^2\Delta t \tag{24}$$

$$K^y = \frac{1}{2}\left[\cos\theta\sin^2\theta + \cos^2\theta\sin\theta\right] Ul \tag{25}$$

$$K^{\text{cross}} = \left[\cos^2\theta\sin\theta - \cos\theta\sin^2\theta\right] Ul \tag{26}$$

For the Cartesian case, the edge length l is equal to the cell-center spacing Δx, and θ is constrained to the interval $[0, \pi/2]$. As expected, the flow-aligned ($\theta = 0$) behavior is identical to the 1-D case and the 2-D flow-aligned triangular case. The condition for monotonicity is $0 \leqslant 1 - C_W - C_N \leqslant 1$, which for arbitrary θ leads to the condition

$$0 \leqslant \frac{U\Delta t}{l} \leqslant \frac{1}{\sqrt{2}\sin\left(\theta + \frac{\pi}{4}\right)} \tag{27}$$

The worst-case scenario for monotonicity in the Cartesian case occurs at $\theta = \pi/4$, for which $U\Delta t/l \leqslant 1/\sqrt{2}$. For equivalent edge lengths l, the Cartesian grid permits a longer time step by a factor of $2\sqrt{2}/\sqrt{3} \approx 1.63$. For equivalent cell areas (i.e., equivalent spatial resolution), the Cartesian grid still permits a longer time step, although by only a factor of $\sqrt{2}/\sqrt[4]{3} \approx 1.075$.

3. IDEALIZED SIMULATIONS

To corroborate the modified equation analysis of Section 2, we calculate empirical estimates of numerical diffusion from a series of idealized scalar transport test cases simulating uniform flow in a flat-bottomed channel. Given the uniform nature of the test domain and our desire to quantify the full 2-D diffusion tensor, we employ the method of moments [21], estimating diffusion coefficients from the time rate of spreading of a Gaussian scalar plume. The domain, shown in Figure 3, extends $L = 50$ km in the x-dimension and $W = 20$ km in the y-dimension, with a constant depth of 10 m. In all cases, the northern and southern boundaries are closed, and a uniform velocity $u = 0.5$ m s^{-1} is imposed throughout the domain. Simulations are carried out using the discretization given by (4) and (10).

The simulations sample a portion of the parameter space relevant to physical simulations of estuarine domains, where the parameters are l, the side length of the cells ranging from 75 to 500 m; θ, the smallest positive counterclockwise angle between grid edges; the x-axis, ranging from 0° to 60°; and the time step. The grid is created by taking a tessellation of equilateral triangles, rotating by θ, and removing all triangles that fall outside the rectangular boundary (leaving the grid with two or four 'ragged' edges). The time step ranges from 40 s for the highest-resolution runs to 120 s for the low-resolution runs. The corresponding range of the volumetric Courant number is from 0.80 to 0.24.

DOI: 10.1002/fld

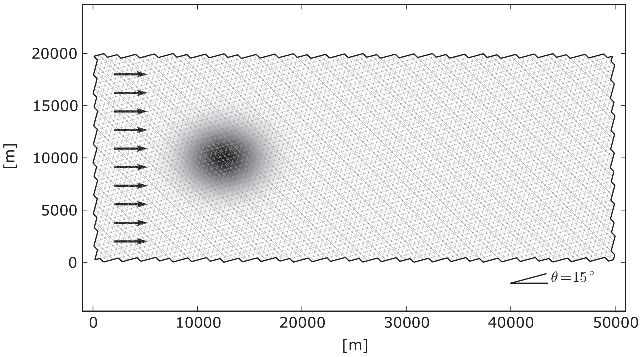

Figure 3. Idealized domain with initial condition for passive scalar. $l = 500\text{m}$, $\theta = 15^\circ$.

The passive scalar field is initialized with a Gaussian plume, centered at $(L/4, W/2)$ with $\sigma_x = 0.05L$ and $\sigma_y = 0.1W$ as shown in Figure 3. These dimensions ensure that the plume is sufficiently free from wall effects as it advects through the domain, while keeping computational costs reasonable. An additional constraint on the size of the plume is that it should cover a sufficient number of cells to reasonably approximate a Gaussian distribution.

The simulations run until the domain-integrated scalar mass falls to 99% of its original value, at which point the trailing 25% of the simulation period is additionally discarded to avoid outflow boundary effects, leaving the period $t \in [t_0, t_f]$. The tracer field is output every 20 min, and each snapshot of the tracer distribution is processed to find the centroid, the second central moments in x and y, and the spatial covariance. The centroid of the scalar plume is computed at each output time step according to

$$\mu = \frac{\sum_i \mathbf{r}_i V_i A_i}{\sum_i V_i A_i} \tag{28}$$

where $\mathbf{r}_i = (x_i, y_i)$ denotes the location of the ith cell center, V_i its volume, and A_i its tracer concentration. The second central moments and spatial covariance are then computed as

$$\sigma_x^2 = \frac{\sum_i (x_i - \mu_x)^2 V_i A_i}{\sum_i V_i A_i} \tag{29}$$

$$\sigma_y^2 = \frac{\sum_i \left(y_i - \mu_y\right)^2 V_i A_i}{\sum_i V_i A_i} \tag{30}$$

$$\text{cov}_{xy} = \frac{\sum_i (x_i - \mu_x)\left(y_i - \mu_y\right) V_i A_i}{\sum_i V_i A_i} \tag{31}$$

We estimate the time-averaged diffusion coefficients K^x, K^y, and K^{cross} as one-half the average rate of growth for the second central moments and covariance [21],

$$K^x = \frac{1}{2}\overline{\left(\frac{\mathrm{d}\sigma_x^2}{\mathrm{d}t}\right)} = \frac{1}{2}\left(\frac{\sigma_x^2\big|_{t_f} - \sigma_x^2\big|_{t_0}}{t_f - t_0}\right) \tag{32}$$

$$K^y = \frac{1}{2}\overline{\left(\frac{\mathrm{d}\sigma_y^2}{\mathrm{d}t}\right)} = \frac{1}{2}\left(\frac{\sigma_y^2\big|_{t_f} - \sigma_y^2\big|_{t_0}}{t_f - t_0}\right) \tag{33}$$

$$K^{\text{cross}} = \overline{\left(\frac{\mathrm{dcov}_{xy}}{\mathrm{d}t}\right)} = \left(\frac{\text{cov}_{xy}\big|_{t_f} - \text{cov}_{xy}\big|_{t_0}}{t_f - t_0}\right) \tag{34}$$

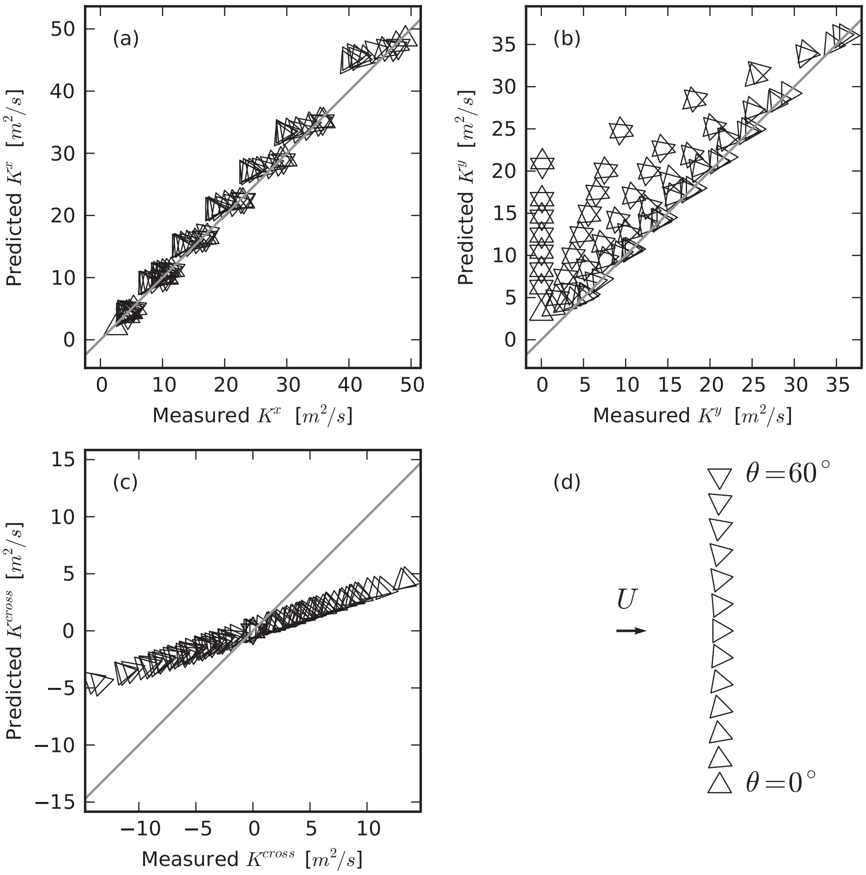

Figure 4. Measured versus predicted numerical diffusion, using the average of the independent modified equations. (a) Longitudinal diffusion, (b) lateral diffusion, and (c) cross-diffusion. Grid alignment in (a)–(c) is shown by the orientation of the triangular markers, relative to a left-to-right flow as illustrated in (d).

where t_0 corresponds to the time when the initial Gaussian plume is introduced and t_f the cutoff time as described before. Correlation between predicted and measured numerical diffusions, using the results of the independent modified equation analysis (Section 2.1), is shown in Figure 4. Both K^x and K^y are consistently overpredicted, and for the case of K^y with $\theta = 0$, it is apparent that measured diffusion is nearly 0 despite the prediction being only mildly attenuated relative to the unaligned case. As shown in Figure 5, the analytical predictions for numerical diffusion using the combined approach of Section 2.2 correlate very well with measured diffusion from simulations, with slightly more noise in the correlation for K^{cross} than K^x or K^y.

4. AUTOMATED GRID ALIGNMENT IN AN IDEALIZED FLOW

The analytically derived diffusion coefficients (15)–(17) quantify how much the orientation of grid cells affects grid diffusion, confirming that cells aligned with the velocity field lead to zero lateral diffusion. This motivates the desire to create computational grids locally aligned with the dominant flow, a common but labor-intensive practice. In this section, we describe an algorithm for automating this process, demonstrated on an idealized, rigid-body rotation flow field, and evaluate the efficacy of aligning the grid in terms of the measured and predicted numerical diffusion coefficients. The steps to creating a flow-aligned grid are as follows: (i) run a reference simulation on an unaligned grid for a representative period; (ii) extract principal velocity directions from the reference simulation; (iii) construct principal streamlines by integrating trajectories in the principal

DOI: 10.1002/fld

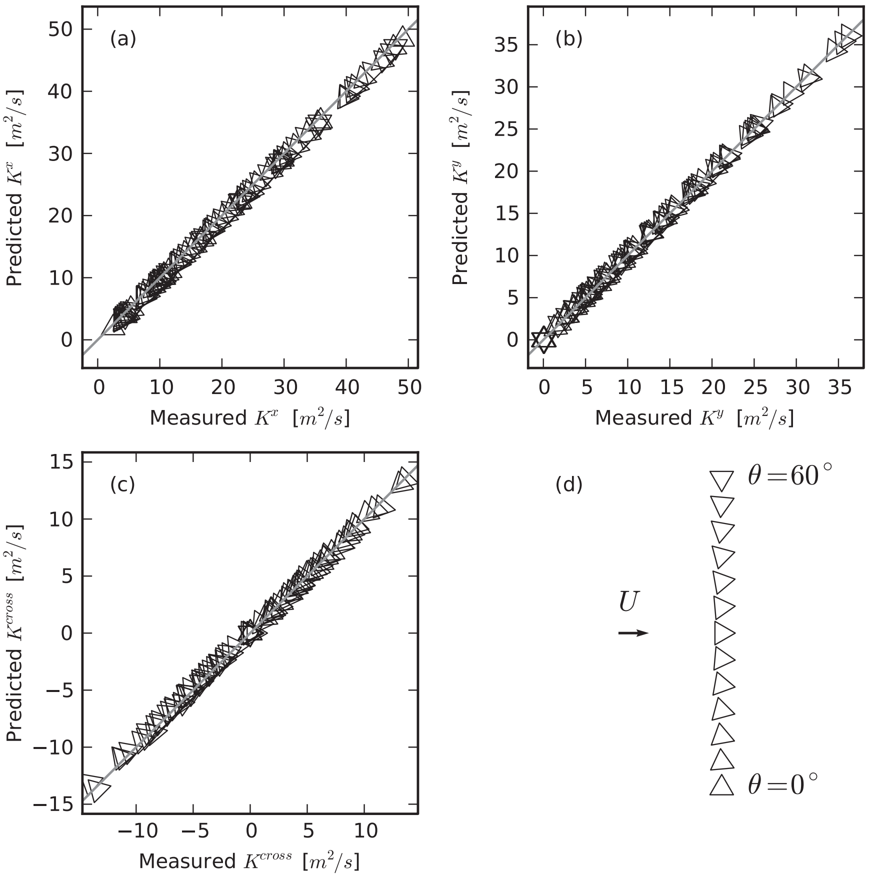

Figure 5. Measured versus predicted numerical diffusion using a recurrence relation to unify the two types of cells. (a) Longitudinal diffusion, (b) lateral diffusion, and (c) cross-diffusion. Grid alignment in (a)–(c) is shown by the orientation of the triangular markers, relative to a left-to-right flow as illustrated in (d).

velocity field; and (iv) generate a new grid in which the principal streamlines are embedded in the grid as cell edges.

The flow for the idealized flow test case is a steady rigid-body rotation,

$$u = \frac{2\pi}{T} y \tag{35}$$

$$v = -\frac{2\pi}{T} x \tag{36}$$

such that a full revolution is achieved over a period T. The domain is the region $[-L, L]^2$, chosen to be square such that the boundaries do not coincide with streamlines.

The unaligned grid is denoted G, with a velocity field $u_i^m(t_n)$, where $i \in \{1, 2\}$ indicates components of the 2-D depth-averaged velocity field, m an index unique to each cell, and t_n the time at step n of the simulation. For the steady rigid-body rotation case, u_i^m is clearly independent of t_n, but in anticipation of the tidal case described in Section 5, we retain the time variability during the present description of the algorithm. The upper-right quadrant of the unaligned grid G for the test case is shown in Figure 6, with $L = 1000$ m and a nominal edge length l of 20 m. In the more general case, the nominal edge length, for both the unaligned and aligned grids, is potentially variable in space, prescribed by the target edge length $l = l(x, y)$.

The reference simulation on the unaligned grid is run for a period long enough to obtain accurate principal velocities from the flow, such as a spring–neap cycle for a tidal flow. In the steady test

DOI: 10.1002/fld

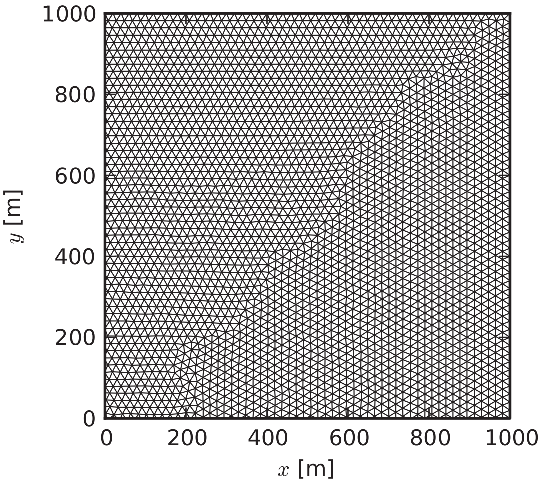

Figure 6. Unaligned grid for rigid-body rotation test case, upper-right quadrant.

case, the reference simulation is run for 12 h after which we extract the principal velocities from the simulation results. For the purposes of the principal velocities, the velocity field is treated as constant within each cell. For each cell m, we compute the velocity correlation matrix for N time steps of the unaligned simulation as

$$C_{ij}^m = \overline{u_i^m u_j^m} = \frac{1}{N}\sum_{n=0}^{N} u_i^m(t_n)\, u_j^m(t_n) \tag{37}$$

as well as the eigenvalues λ_1^m and λ_2^m of C^m, with $\lambda_1^m \geqslant \lambda_2^m$. The principal velocity direction $\hat{u}_i^m$ is then defined as the eigenvector corresponding to λ_1^m, and the ratio $e^m \equiv \lambda_2^m/\lambda_1^m$ is a measure of how well defined the principal velocity direction is. A direct analog to a tidal ellipse can be defined with the orientation of the major axis parallel to the principal velocity direction $\hat{u}_i^m$, and the lengths of the major and minor axes are defined to be λ_1^m and λ_2^m, respectively, although the principal velocity captures variability at all frequencies, whereas tidal ellipses are specific to each tidal constituent. For the steady flow of the test case, the principal velocity directions coincide with the steady velocity field itself, within a constant factor (as the principal directions are eigenvectors).

A collection of flow-aligned curves, termed *principal streamlines*, are created by integrating trajectories in the principal velocity field. These contours are similar to streamlines but differ in that they come from an integration of the principal velocity directions rather than the instantaneous velocity field. Two aspects of the creation of principal streamlines for grid alignment present a challenge relative to calculating streamlines in an instantaneous velocity field. The first issue is that because $\hat{u}_i$ is defined as an eigenvector, it is only determined within a constant factor. We treat this issue by assuming that when a principal streamline crosses from cell m to cell p, $\hat{u}_i^p$ is scaled by a factor $\alpha = \pm 1$ such that $\hat{u}_i^m \alpha \hat{u}_i^p > 0$ (utilizing the Einstein summation notation). For a well-resolved velocity field, this heuristic has been robust, and in all cases where $\hat{u}_i^m \hat{u}_i^p \approx 0$, we found that $e^m \approx 1$, indicating that the principal velocity direction in cell m was poorly defined and the local orientation of the grid irrelevant. The second issue is that only a limited number of curves can be embedded in the aligned grid. The collection of principal streamlines should be dense enough to force most cells to be aligned but not so dense as to overconstrain the grid generation process, which would lead to poor-quality cells. We define the lower limit of the proximity between shorelines and principal streamlines as a constant factor ϵ_l times the local target edge length l. We have found that a reasonable trade-off between grid quality and alignment is achieved with $\epsilon_l = 3$, meaning that there is space for three to four cells between any two principal streamlines or shorelines.

DOI: 10.1002/fld

To evenly distribute principal streamlines throughout the domain while maintaining this lower bound on proximity between curves, we have implemented the algorithm described in the following. Here, T is the constrained Delaunay triangulation of the shoreline, incrementally updated with principal streamlines as they are calculated. All edges, from both the shoreline and the principal streamlines, are sampled such that adjacent vertices are separated by at most the local grid scale l. We define $m(\mathbf{x})$ to be the grid cell in G containing point $\mathbf{x}$, $\mathbf{c}(t)$ to be the circumcenter of the Delaunay triangle t, and $r(t)$ the circumradius of the Delaunay triangle t. T' is defined as the set of triangles $t \in T$ for which $m(\mathbf{c}(t)) \neq \emptyset$ and $\mathrm{e}^{m(\mathbf{c}(t))} < \epsilon_e$. These are the triangles of T that fall inside the computational domain and for which the circumcenter lies in a region where the principal velocity direction is well defined. We choose $\epsilon_e = 0.2$ from the observed distributions of e^m. Each principal streamline is traced according to the following algorithm:

```
1. Find the triangle $t \in T'$ with the greatest circumradius relative to the local scale, $r(t)/l(\mathbf{c}(t))$.
2. If $r(t) < \epsilon_l l(\mathbf{c}(t))$, then terminate.
3. Add the point $\mathbf{c}(t)$ to the triangulation $T$.
4. For each $\alpha_0$ in $\{+1, -1\}$,
   (a) Initialize $i \leftarrow 0$, $\mathbf{x}_0 \leftarrow \mathbf{c}(t)$, $\alpha \leftarrow \alpha_0$.
   (b) Loop
       i. Calculate the minimum $\delta$ such that $m\left(\mathbf{x}_i + \delta\alpha\hat{\mathbf{u}}^{m(\mathbf{x}_i)}\right) \neq m(\mathbf{x}_i)$, that is, take a step in
          the principal velocity direction to reach another grid cell.
       ii. $\mathbf{x}_{i+1} \leftarrow \mathbf{x}_i + \delta\alpha\hat{\mathbf{u}}^{m(\mathbf{x}_i)}$.
       iii. If the distance between $\mathbf{x}_{i+1}$ and any constrained edge of $T$ is less than $\epsilon_l l$, then end
            loop.
       iv. Insert the point $\mathbf{x}_{i+1}$ into $T$, with a constrained edge $\overline{\mathbf{x}_i\mathbf{x}_{i+1}}$.
       v. If $\hat{\mathbf{u}}^{m(x_i)} \cdot \hat{\mathbf{u}}^{m(x_{i+1})} < 0$, then $\alpha \leftarrow -\alpha$.
       vi. $i \leftarrow i + 1$.
```

This process is repeated until the termination condition in step 2 is achieved. The set of all edges added in step 4(b)iv form the principal streamlines, which will be used as constraints in the grid generation step. In the rigid-body rotation test case, the principal streamline tracing generates a set of concentric rings as shown in Figure 7, with a gap of approximately $3l$ between each ring and a similarly sized gap in each ring. Finally, the set of scale-compatible principal streamlines are included as additional constraints for the generation of an aligned grid, following the method described in Appendix B. The upper-right quadrant of the aligned grid for the rigid-body rotation test case is shown in Figure 8.

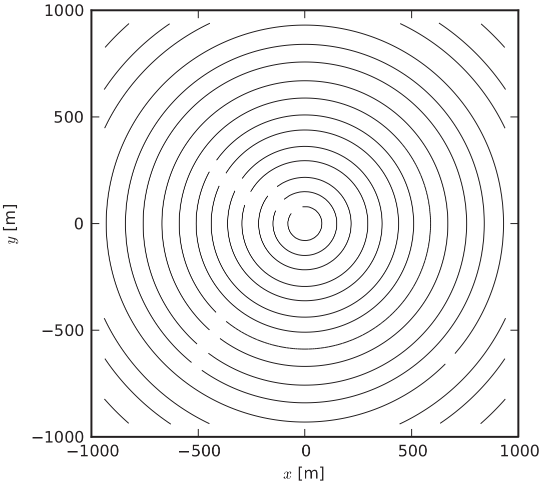

Figure 7. Principal streamlines extracted from the unaligned simulation.

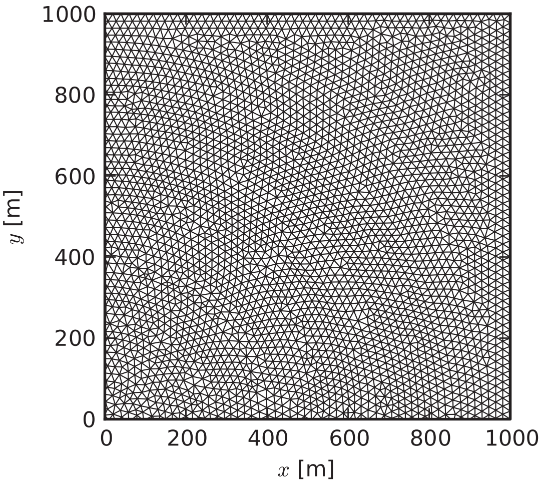

Figure 8. Upper-right quadrant of the aligned grid for the rigid-body rotation test case.

To quantify the effects on diffusion of orienting the grid with the principal streamlines, we advect a passive tracer field in both the unaligned and aligned domains, initialized with a Gaussian concentration distribution centered at $(L/2, 0)$ with standard deviation $\sigma = L/10$, as shown in Figure 9. The rigid-body rotation flow advects this tracer distribution for one full revolution period T. Comparing the second central moments of the initial and final concentration distributions, we can estimate the grid diffusion during one revolution. Although the plume spans some finite range of angles relative to the center of the domain, for the sake of simplicity, we analyze the Cartesian moments σ_x^2 and σ_y^2, approximating the radial diffusion coefficient

$$K^{\mathrm{r}} \approx \frac{\sigma_x^2(T) - \sigma_x^2(0)}{T} \tag{38}$$

and tangential diffusion coefficient

$$K^{\theta} \approx \frac{\sigma_y^2(T) - \sigma_y^2(0)}{T} \tag{39}$$

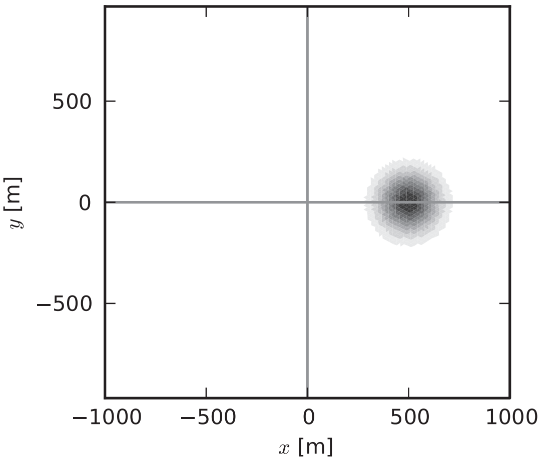

Figure 9. Initial scalar field on the unaligned grid. Grayscale graduations indicate 0.1 contours, with a maximum concentration of 1.0

We find $K^{\mathrm{r}} = 0.081$ m^2 s^{-1} and $K^{\theta} = 0.145$ m^2 s^{-1} in the case of the unaligned grid and $K^{\mathrm{r}} = 0.020$ m^2 s^{-1} and $K^{\theta} = 0.134$ m^2 s^{-1} for the aligned grid. In the evaluation of the analytical diffusion coefficients at the center of the release, a perfectly aligned grid should have zero radial diffusion and $K^{\theta} = 0.162$ m^2 s^{-1}. A 'perfectly' unaligned grid, taking $\theta = \pi/6$, leads to $K^{\mathrm{r}} = 0.105$ m^2 s^{-1} and $K^{\theta} = 0.137$ m^{2}s^{-1}. The along-flow diffusion coefficient K^{θ} is comparable in both cases and falls within the expected range based on the analytical coefficients. Although we would expect that along-flow tangential diffusion would be slightly greater on the aligned grid, the results show a slightly attenuated tangential diffusion. The major difference is of course in the radial, or across-flow, diffusion. Here, the observed diffusion on the unaligned grid is 77% of the predicted maximum, whereas the aligned grid sees only 19% of the predicted maximum.

5. APPLICATION TO A PHYSICAL SYSTEM

To further explore the effects of grid alignment and grid diffusion, we describe in this section a physically realistic tidal simulation of San Francisco Bay, with and without grid alignment, and a comparison of the resulting salinity fields. We use the SUNTANS [2] code in hydrostatic mode to solve the 3-D Reynolds-averaged primitive equations with the Boussinesq approximation in a rotating frame of reference. In accordance with the hydrostatic approximation, the vertical velocity field w is solved by continuity. We assume that horizontal numerical diffusion is in the order of or larger than the physical horizontal eddy viscosity, and the latter is omitted from the equations of motion. The equations of motion are then

$$\frac{\partial u}{\partial t} + \nabla \cdot (\mathbf{u}u) - fv = -\frac{1}{\rho_0}\frac{\partial p}{\partial x} + \frac{\partial}{\partial z}\left(\nu_V \frac{\partial u}{\partial z}\right) \tag{40}$$

$$\frac{\partial v}{\partial t} + \nabla \cdot (\mathbf{u}v) + fu = -\frac{1}{\rho_0}\frac{\partial p}{\partial y} + \frac{\partial}{\partial z}\left(\nu_V \frac{\partial v}{\partial z}\right) \tag{41}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{42}$$

where u is eastward velocity, v northward velocity, $f = 2\omega \sin\phi$ the Coriolis parameter with angular velocity of the Earth ω and latitude ϕ, ν_{V} the vertical eddy viscosity derived from a two-equation turbulence model (Mellor–Yamada 2.5 [22]), and ρ_0 a constant reference density ($\rho_0 = 1000$ kg m^{-3}). The hydrostatic pressure p is defined by $\partial p/\partial z = -(\rho_0 + \rho)\, g$, where $\rho = \beta s$ is the density anomaly due to salinity s and coefficient of contractivity β. These equations are discretized on a prismatic finite-volume grid comprising unstructured triangles in the horizontal and structured z-levels in the vertical.

Two nearly identical simulations of San Francisco Bay are carried out, differing only in the alignment of grid cells. Model outputs from both simulations are compared with observations of depth-averaged salinity in the study area, South San Francisco Bay. The model domain, shown in Figure 10, extends to the south through South San Francisco Bay as far as Calaveras Point, beyond which the basin devolves into a network of tidal sloughs. In the northern end of the bay, the domain includes San Pablo Bay, Suisun Bay, and a pair of false deltas each with a width of 1 km. The false deltas are surrogates for the Sacramento and San Joaquin Rivers and have hypsometric curves matched to accurate delta bathymetry, evenly split between the two channels. The ocean boundary is an arc of 100-km radius, centered on the Golden Gate. The spatial resolution is nominally 5 km in the coastal ocean, gradually decreasing to 250 m in South San Francisco Bay, leading to approximately 26,000 2-D cells and 430,000 3-D cells. The vertical dimension is discretized with 40 z-levels, with cell thicknesses ranging from 0.5 m at the free surface to 10 m at a depth of 100 m. Depths below 100 m are lumped into a single layer. The time step is 60 s, and the scalar transport routines utilize substepping such that the overall hydrodynamic time step is limited only by the propagation speed of wetting and drying fronts and internal waves.

The ocean boundary is forced with zero velocity and a free-surface height set by observed tides from Point Reyes. So that a phase and amplitude agreement at the Golden Gate can be reached,

DOI: 10.1002/fld

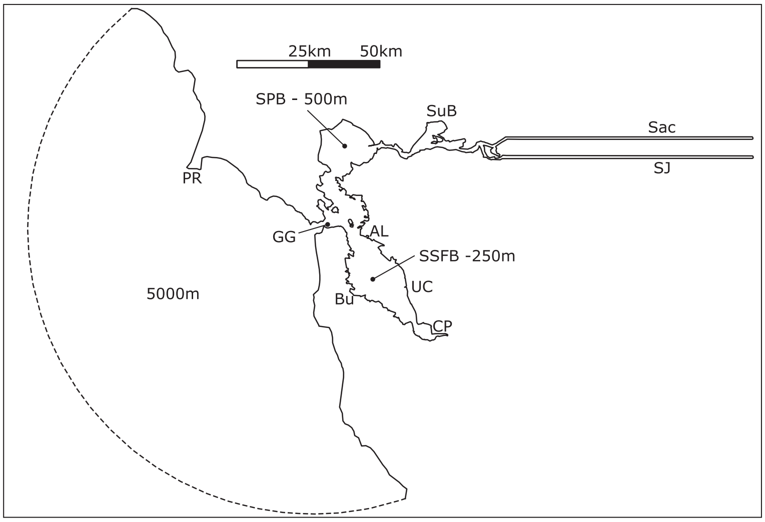

Figure 10. Computational domain for the San Francisco Bay model runs. Nominal grid edge length is 5000 m in the coastal ocean, 500 m in the North San Francisco Bay, and 250 m in South San Francisco Bay (SSFB). For computational efficiency and boundary condition simplicity, the Sacramento (Sac)–San Joaquin (SJ) delta has been replaced by a pair of long dissipative channels. The dashed line represents the tidal ocean boundary condition. AL, Alameda; Bu, Burlingame; CP, Calaveras Point; GG, Golden Gate; PR, Point Reyes; SPB, San Pablo Bay; SuB, Suisun Bay; UC, Union City.

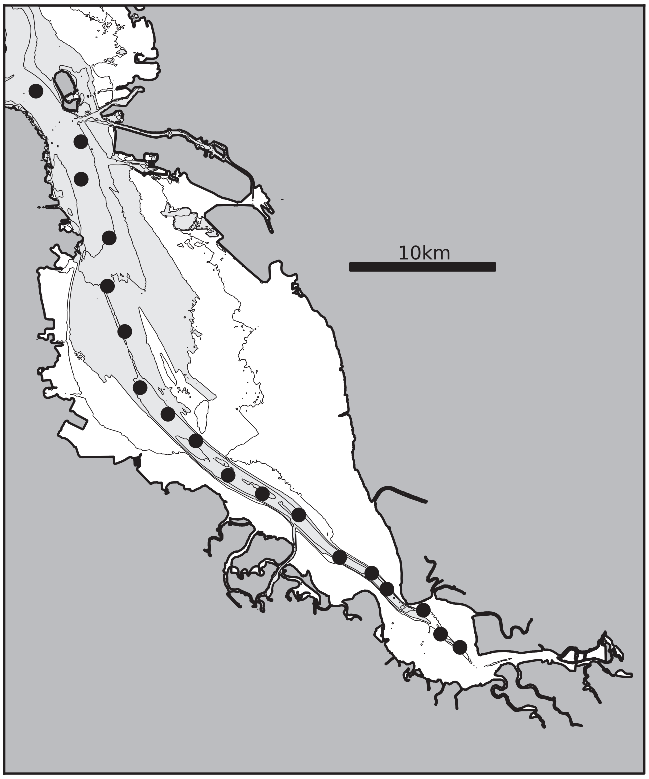

Figure 11. South San Francisco Bay bathymetry used in the physically realistic model runs. Contours at −15, −10, −5 (shaded), and −2.5 m (NAVD88). Transect sampling locations are shown as black dots, with the start of the transect (0 km) at the southern end.

DOI: 10.1002/fld

the ocean tides are advanced in time by 17.7 min and attenuated by a factor of 0.97. Winds in the model were spatially uniform, as measured at Alameda. Wind stress is calculated using the drag law of Large and Pond [23]. Freshwater inputs are assembled from a wide range of data sources, including delta flows from the delta Dayflow [24] measurements, US Geological Survey gaged streams, and ungaged streams correlated to nearby gages based on drainage area. Discharges from wastewater treatment plants are also included, based on daily data for the larger sources and monthly data otherwise. Evaporation and precipitation are spatially uniform, with evaporation based on a monthly climatology at Burlingame and hourly precipitation from a gage in Union City. To avoid introducing a bias toward the validation of one grid over the other, we avoid any calibration beyond the already mentioned phase and amplitude tuning at the Golden Gate. The simulations are initialized with a quiescent velocity field and a salinity distribution extrapolated from salinity observations [25] along the thalweg of the bay. Following a spin-up period starting on August 1, 2008, and running for 90 days, the analysis period is October 30, 2008, through January 22, 2010.

Although the model domain includes the majority of the San Francisco Bay system, the study area for the simulations is South San Francisco Bay (Figure 11), a 45-km-long basin forming one of two branches in the larger San Francisco Bay system. Most of this embayment exhibits a channel–shoal morphology, with channel depths of 15–20 m and shoal depths of 0–4 m. The pronounced channel–shoal geometry leads to significant lateral shear and lateral salinity gradients.

In the present case of San Francisco Bay, a tidally dominated estuarine and coastal ocean domain, the depth-averaged principal velocities are essentially stationary over timescales greater than the spring–neap cycle, such that the representative period for the reference simulation should span at least one spring–neap tidal cycle. We use model output from the unaligned simulations over the first 90 days of the analysis period to establish a representative principal velocity field. Figure 12 shows the principal velocity ellipses for a portion of the domain, resampled to a coarse regular grid for clarity as well as normalized such that the length of the major axis is constant. Principal streamlines, consistent with the grid resolution function $l(x, y)$, are depicted in Figure 13. The flow-aligned grid, with identical resolution and shoreline data, is then constructed such that the flow-aligned contours are embedded in the grid as edges of cells (shown in Figure 14). Appendix B gives further details

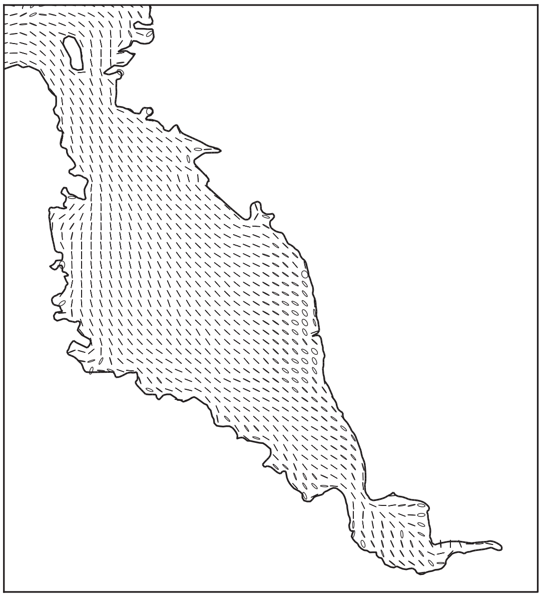

Figure 12. Principal velocity ellipses in the vicinity of the study area, derived from model output from the unaligned grid. Ellipses are calculated for all cells but are shown interpolated onto a coarse regular grid and normalized by major axis magnitude for clarity.

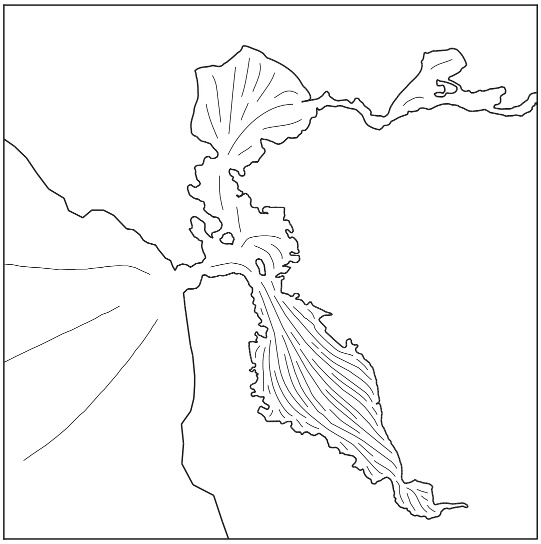

Figure 13. Flow-aligned contours, traced from the principal velocities, ready for input to the regridding process. The separation between adjacent contours scales with the grid resolution, leading to sparser contours in the north and west.

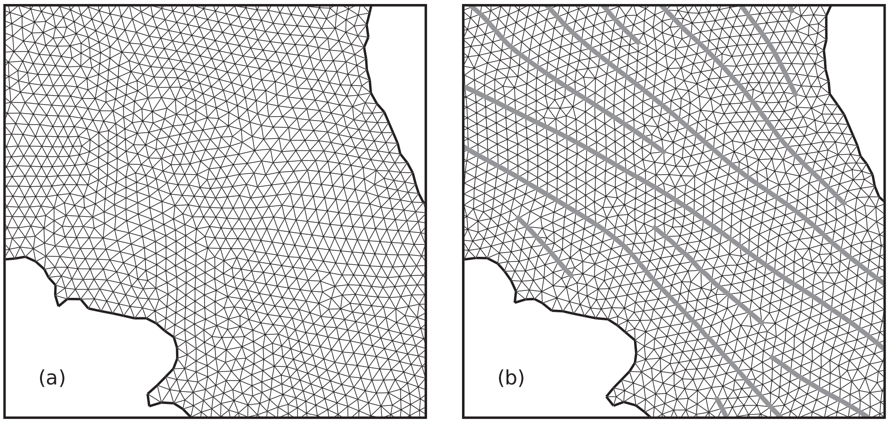

Figure 14. Details of the computational grids, toward the southern end of the study area. (a) The unaligned grid. (b) The aligned grid with flow-aligned contours superimposed.

on the mesh generation algorithm. Figure 15 compares distributions of cell-center spacing and cell skewness between the original unaligned grid and the aligned grid. Cell skewness is defined as $1 - A_{\mathrm{eq}}/A$, where A is the area of the cell and A_{eq} is the area of an equilateral triangle with the same circumcircle, such that 0.0 indicates a perfectly equilateral cell and 1.0 a zero-area sliver. These distributions reflect the entire domain, but because the majority of the cells are in the high-resolution South San Francisco Bay area, it is this area with a nominal 250-m resolution that dominates the histograms. Cell-center spacings are similar between the two grids, but skewness shows an increase of moderately low-quality cells in the aligned grid, most likely due to the greater number of constraints on cell placement.

To quantify the effects of grid alignment in the San Francisco Bay case, we evaluate (15)–(17) to obtain K^x, K^y, and K^{cross} for each grid cell, depth-averaged and averaged over the first 90 days of the analysis period. Whereas the method of Burchard and Rennau [26] would likely give a more accurate measure of the total numerical mixing for a specific scalar field, our approach directly evaluates the numerical diffusion coefficients rather than calculating decay of scalar variance on a per-scalar basis (although a diffusion coefficient may be obtained by normalizing by the gradient of

DOI: 10.1002/fld

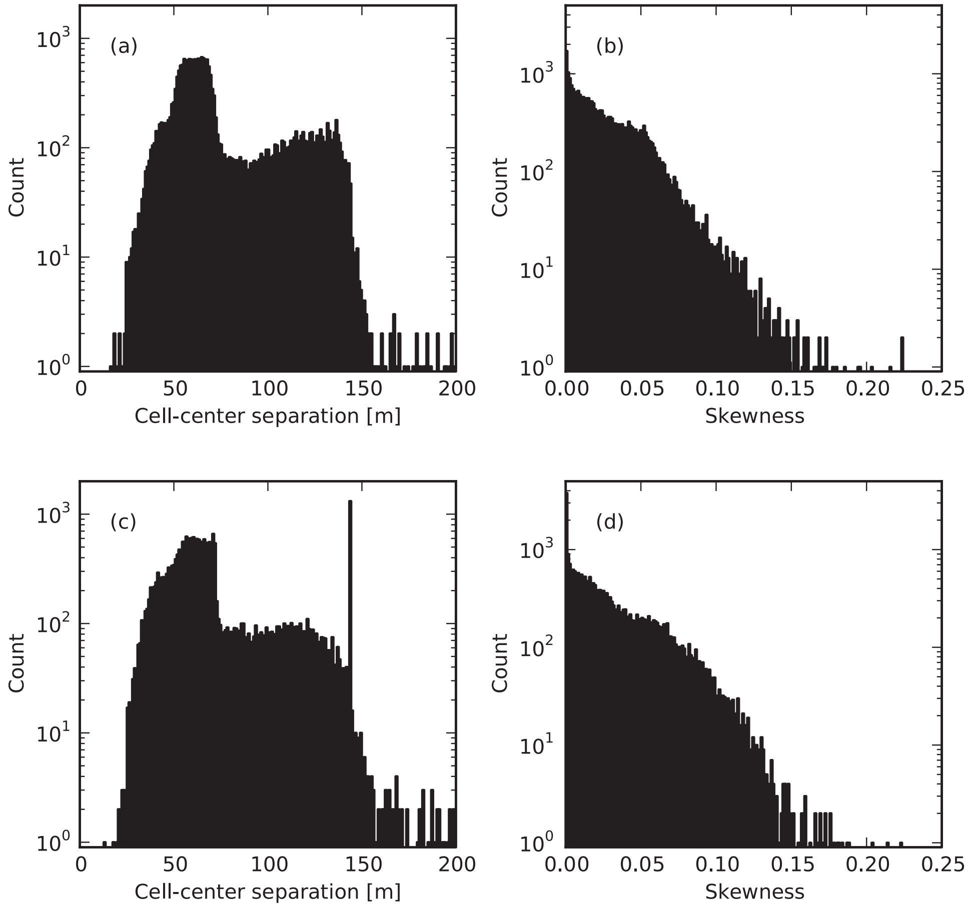

Figure 15. (a) Distribution of cell-center spacing for the unaligned grid. (b) Distribution of skewness for the unaligned grid. (c) Distribution of cell-center spacing for the aligned grid. (d) Distribution of skewness for the aligned grid.

the scalar concentration). Our method also retains the distinction between lateral and longitudinal diffusions, which is helpful in interpreting the effects of numerical mixing on the synoptic evolution of the scalar field.

The alignment angle θ was taken from the cell edge most closely aligned with the instantaneous velocity vector, modulo $\pi/3$. Because the expressions of Section 2 give the diffusion tensor in a coordinate system parallel to the instantaneous velocity, it is necessary to rotate each instantaneous diffusion tensor into a common coordinate system before averaging over time and depth. The principal velocity direction of each cell is used as the common coordinate system, such that the resulting diffusion tensor gives K^x as the longitudinal diffusion, parallel to principal flow, and K^y as the lateral diffusion, perpendicular to principal flow. Figure 16 shows the resulting spatial distributions of longitudinal K^x and lateral K^y. Volume-averaged and time-averaged diffusion coefficients for the study area, roughly defined as the southern two-thirds of the region shown in Figure 16, over days 91–180, are $K^x = 12.9$ m^2 s^{-1}, $K^y = 6.5$ m^2 s^{-1}, and $K^{\text{cross}} = 0.25$ m^2 s^{-1} for the unaligned grid. For the aligned grid, the same calculation gives $K^x = 12.8$ m^2 s^{-1}, $K^y = 3.2$ m^2 s^{-1}, and $K^{\text{cross}} = 0.1$ m^2 s^{-1}, showing a significant decrease in lateral numerical diffusion whereas the longitudinal component is essentially unchanged.

5.1. Validation of unaligned and aligned model output

Although the focus of our comparisons is the difference between the two models rather than specific validation against observations, we include a comparison of both models with observed salinity in

DOI: 10.1002/fld

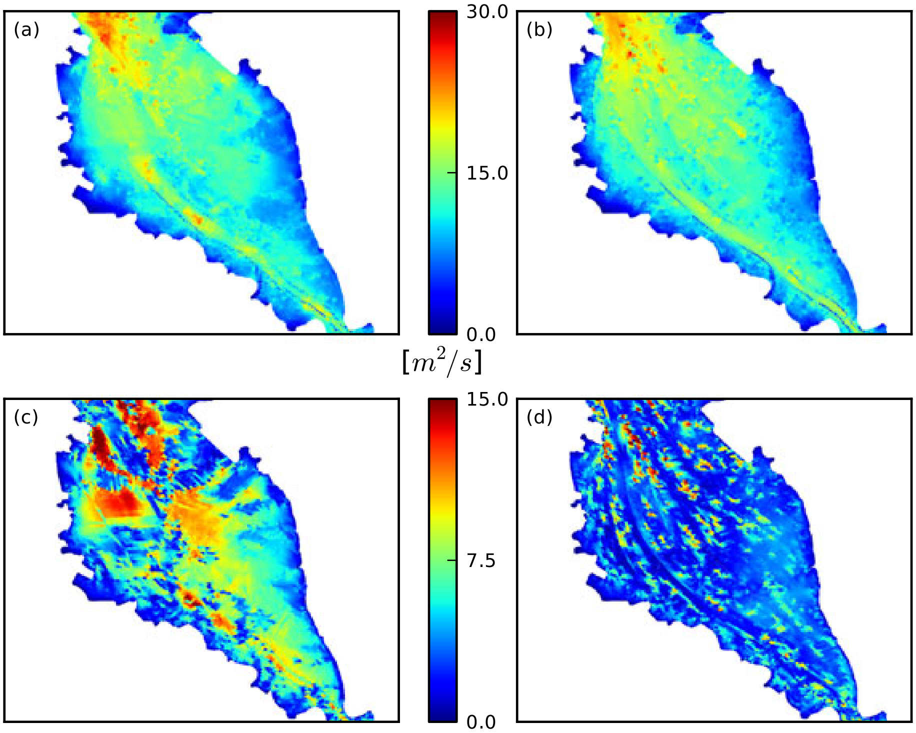

Figure 16. Distribution of depth-averaged and time-averaged longitudinal and lateral diffusions in square meters per second. (a) K^x for unaligned grid. (b) K^x for aligned grid, showing minimal decrease in numerical diffusion relative to the unaligned grid. (c) K^y for unaligned grid. (d) K^y for aligned grid, showing a significant decrease in numerical diffusion.

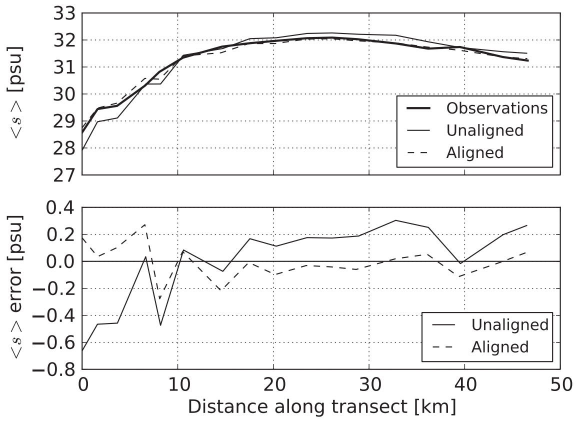

Figure 17. Comparison of unaligned and aligned model outputs and observations of salinity along a longitudinal transect in South San Francisco Bay (shown in Figure 11), from November 5, 2008. (upper) Depth-averaged salinity and (lower) absolute error of model output compared with observations.

the South San Francisco Bay. A full treatment of model predictive skill would necessarily include evaluations of forcing data, the turbulence closure and bathymetry, and is beyond the scope of this manuscript. Model data comparisons of the longitudinal salinity field are shown in Figures 17 and 18. Each set of salinity observations was collected by the US Geological Survey Polaris [25] over the course of a particular day, starting at the southern end of San Francisco Bay. Model output has been interpolated in time to match the respective time of each observation along the transect.

DOI: 10.1002/fld

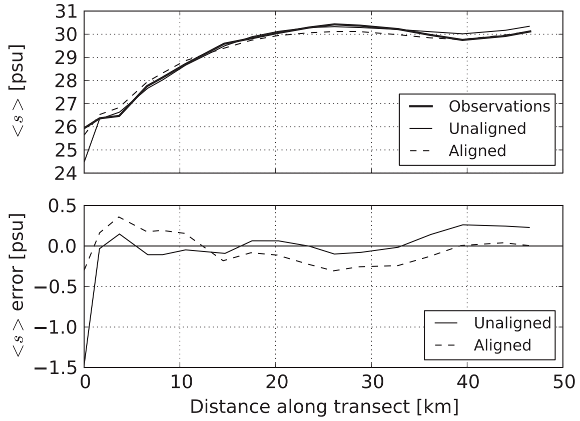

Figure 18. Comparison of unaligned and aligned model outputs and observations of salinity along a longitudinal transect in South San Francisco Bay (shown in Figure 11), from January 1, 2009. (upper) Depth-averaged salinity and (lower) absolute error of model output compared with observations.

Both the model and observations resolve the water column at approximately 1-m resolution but have been depth-averaged for simplicity.

Considering only the portion of the transects that fall within the high-resolution portion of the grid (the region depicted in Figure 11), we calculate bias (mean error), the Pearson product–moment correlation coefficient, the index of agreement [27], and the root mean square error in the longitudinal salinity gradient over all transect observations available for the 450-day analysis period. These results and definitions of the respective metrics are summarized in Table I. Although the unaligned grid has a marginally smaller bias, both the unaligned and aligned simulations underpredict salinity, and the difference in bias is small relative to the standard deviation of the errors. The aligned grid, however, has better correlation with observations and a slightly better prediction of longitudinal gradients.

Table I. Error statistics for comparison with longitudinal salinity transects.

	Definition	Unaligned	Aligned
Mean error (psu)	$\left\langle s_i^n - \hat{s}_i^n \right\rangle$	−0.36 ± 0.69	−0.42 ± 0.49
Correlation coefficient (r)	$\mathrm{cov}(s, \hat{s})/\sigma_s \sigma_{\hat{s}}$	0.945	0.958
Index of agreement	$1 - \left(\sum \left(\widehat{s_i^n} - s_i^n \right)^2 / \sum \left(\left| \hat{s}_i^n - \langle s \rangle \right| + \left| s_i^n - \langle s \rangle \right| \right)^2 \right)$	0.966	0.971
RMS gradient error (psu)	$\left\langle \left[\left(s_i^n - \hat{s}_i^n \right) - \left(s_{i-1}^n - \hat{s}_{i-1}^n \right) \right]^2 \right\rangle^{1/2}$	0.29	0.22

RMS, root mean square.

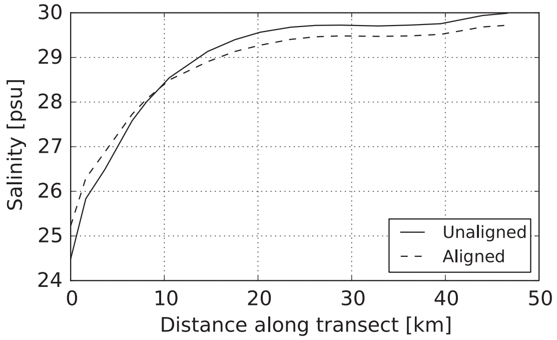

Figure 19. Time-averaged and depth-averaged salinity for the unaligned and aligned grids, along the thalweg of South San Francisco Bay. Decreased longitudinal gradients in the aligned grid model are consistent with an increase in longitudinal diffusion.

DOI: 10.1002/fld

The observed transects, such as those shown in Figures 17 and 18, are available only once a month and are timed to the phase of the tide. As such, there is the possibility of a sampling bias relative to the true mean salinity distribution. Figure 19 shows predicted long-term average salinity distributions sampled every 20 min. The difference in longitudinal salinity gradients between the two simulations is persistent and significant, although of similar order to the error relative to observations.

6. DISCUSSION

A priori estimates of the diffusion tensor have been both verified for idealized simulations and estimated for physically realistic simulations. A distinct advantage to estimation of the diffusion tensor compared with other error estimates is that it communicates the error back to the modeler in the same language as the original governing PDE. In this way, the effects of numerical diffusion and how it compares and interacts with physical dispersion processes are clearer.

Perhaps the greatest advantage of using grid alignment to combat numerical diffusion is that it is code agnostic. Any simulation code that uses a prismatic triangular grid can potentially benefit from an aligned grid, although of course the specifics of the numerical diffusion will vary by choice of numerical scheme. The fact that grid generation in a nonadaptive simulation is decoupled from the computation itself allows for greater reuse of the grid generation methods across simulation frameworks. Of course, a key step in the application of these methods to realistic simulations is the creation of a high-quality grid that enforces cell alignment. Although the method detailed in Appendix B is able to produce reasonably aligned grids, there are always trade-offs. In particular, we found that the price of enforcing cell alignment was a slight decrease in cell quality in terms of cell-center spacing, which in turn affects the maximum scalar advection time step. The 90-day simulations on the unaligned grid average 20.5-h wall time, compared with 23.1 h for the aligned grid (with both simulations using 16 cores of a high-performance computing cluster). Still, given the linear dependence of K_{lat} on the grid size l, a simple reduction of grid scale to achieve the same 50% reduction in the lateral numerical diffusion would require four times more cells and potentially twice as many time steps. Although switching to higher-order advection schemes is almost certainly a more efficient way to achieve the same decrease in numerical diffusion, the two approaches are complementary, and the results of using a flow-aligned grid in conjunction with a high-resolution advection scheme can be expected to perform better than either method alone.

The primary observation from comparing the salinity gradients predicted in the unaligned and aligned simulations is an effective *increase* in longitudinal diffusion in the aligned grid simulations. Although estimates of the longitudinal numerical diffusion are essentially equal between the two grids, the interaction of lateral shear and lateral diffusion leads to an effective increase in the longitudinal diffusion, that is, the shear flow dispersion scaling of [28], $K_{\text{lon}} \sim U^2h^2/K_{\text{lat}}$. In physical systems, this mechanism is often responsible for the bulk of longitudinal dispersion, and in comparing the two simulations, we see this scaling at work, translating the decreased lateral dispersion of the aligned grid into a more efficient longitudinal shear dispersion process. Enhanced longitudinal diffusion is evidenced by the decreased salinity gradients in Figure 19, most notably around 10 km along the transect, and in fact, the salinity gradients in the aligned case more closely track the observations. We note, however, that comparisons between either simulation and the observations should be viewed through a wide-angle lens, as the quality of the validation is affected by a broad set of factors, ranging from the quality of available data for freshwater flows to the treatment of unstable stratification by the turbulence closure. It is the comparison between the two simulations, in particular, the difference in mean gradients shown in Figure 19, that best demonstrates the effect of grid alignment.

We have considered only velocity fields where the principal velocities are statistically stationary, a valid assumption for strongly tidal embayments, but there is a clear extension to nonstationary flows with an adaptive regridding process. Compared with related forms of adaptive mesh refinement, such as the algorithm proposed by Marcuzzi *et al.* [29] in which directionality in the error field adaptively controls anisotropy in a finite element mesh, the numerical schemes with which we are concerned place much tighter constraints on the geometry of the grid. These constraints translate to more computationally intensive grid generation procedures; although our current implementation would be inefficient for frequent, global regridding steps, localized regridding in areas with nonstationary principal velocities would be feasible even with the current implementation. In applications such as persistent, meandering ocean currents or morphological hydrodynamic models, the regridding costs could be amortized over relatively long computational periods.

7. CONCLUSIONS

Numerical diffusion for first-order upwind advection on a regular triangular grid has been analytically derived and shown to agree with estimates from idealized simulations. Two distinct classes of cells, when analyzed independently, lead to two distinct modified equations. The sum of these equations recovers consistency but overestimates numerical diffusion. An analysis that takes into account the coupling of the two cell types results in a numerical diffusion tensor that correctly quantifies the known behavior that flow-aligned grids eliminate lateral numerical diffusion.

In the context of a physically realistic simulation, we have presented a largely unsupervised process for creating a flow-aligned computational grid and compared model results from an unaligned grid and an aligned grid. The mean predicted lateral numerical diffusion within the study area, South San Francisco Bay, was reduced by a factor of 2. Longitudinal salinity gradients were smaller on the aligned grid, consistent with an increase in longitudinal diffusion. Considering the interaction of lateral shear and diffusion, this increase in longitudinal diffusion is in turn consistent with a decrease in lateral diffusion. Model observation comparisons confirm that the salinity gradient is better predicted by the aligned grid model.

APPENDIX A: CALCULATION OF DIFFUSION COEFFICIENTS

Starting from the closed-form recurrence expression for B_1^n,

$$\begin{aligned} B_1^n = \frac{1}{1-\lambda}\left[C_\alpha A_1^n + C_\gamma A_2^n\right] - \frac{\Delta t}{(1-\lambda)^2}\left[C_\alpha A_{1,t}^n + C_\gamma A_{2,t}^n\right] \\ + \frac{(1+\lambda)\,\Delta t^2}{2\,(1-\lambda)^3}\left[C_\alpha A_{1,tt}^n + C_\gamma A_{2,tt}^n\right] + O(l^3) \end{aligned} \tag{A.1}$$

And the update equation for A_3^{n+1},

$$A_3^{n+1} = \left(1 - C_\beta\right) A_3^n + C_\beta B_1^n \tag{A.2}$$

Expressions for the volumetric Courant numbers are

$$C_\alpha = \frac{4U\Delta t \sin\theta}{\sqrt{3}l} \tag{A.3}$$

$$C_\beta = \frac{4U\Delta t \sin\left(\theta + \frac{\pi}{3}\right)}{\sqrt{3}l} \tag{A.4}$$

$$C_\gamma = \frac{4U\Delta t \sin\left(\theta + \frac{2\pi}{3}\right)}{\sqrt{3}l} \tag{A.5}$$

Taylor expansions of A_1, A_2, and their respective derivatives, where a lack of subscript on A implies evaluation at A_3, are

$$A_i^n = A^n + \delta_{xi} A_x + \delta_{yi} A_y + \frac{\delta_{xi}^2}{2} A_{xx} + \frac{\delta_{yi}^2}{2} A_{yy} + \delta_{xi}\delta_{yi} A_{xy} + O\left(l^3\right) \tag{A.6}$$

$$\left(A_i^n\right)_t = -U\left(A_x + \delta_{xi} A_{xx} + \delta_{yi} A_{xy}\right) + O\left(l^2\right) \tag{A.7}$$

$$\left(A_i^n\right)_{tt} = U^2 A_{xx} + O(l) \tag{A.8}$$

DOI: 10.1002/fld

Although the expansions for first and second time derivatives have been truncated at $O\left(l^2\right)$ and $O(l)$, the coefficients on these terms after substitution retain the overall $O\left(l^3\right)$ bound on the modified equation. Distances between cell centers are defined as

$$\delta_{x1} = -l\cos\left(\theta - \frac{\pi}{3}\right) \tag{A.9}$$

$$\delta_{y1} = -l\sin\left(\theta - \frac{\pi}{3}\right) \tag{A.10}$$

$$\delta_{x2} = -l\cos\theta \tag{A.11}$$

$$\delta_{y2} = -l\sin\theta \tag{A.12}$$

Substituting (A.3)–(A.12) into (A.1), we can write B_1^n in the form

$$B_1^n = DA + D^x A_x + D^{xx} A_{xx} + D^{xy} A_{xy} + D^{yy} A_{yy} + O\left(l^3\right) \tag{A.13}$$

where the coefficients are identified by superscripts and subscripts continue to denote partial derivatives. Expressions for each of the coefficients simplify to

$$D = 1 \tag{A.14}$$

$$D^x = -\frac{\sqrt{3}}{4\sin\left(\theta + \frac{\pi}{3}\right)} l \tag{A.15}$$

$$\begin{aligned} D^{xx} = {} & \frac{\sin\left(\theta + \frac{\pi}{3}\right)\left[\cos^2\theta\cos\left(\theta + \frac{\pi}{6}\right) + \sin\theta\cos^2\left(\theta - \frac{\pi}{3}\right)\right] - \frac{3}{8}}{2\sin^2\left(\theta + \frac{\pi}{3}\right)} l^2 \\ & - \frac{\sqrt{3}}{8\sin\left(\theta + \frac{\pi}{3}\right)} U l \Delta t \end{aligned} \tag{A.16}$$

$$D^{yy} = \frac{\sin(3\theta)}{8\sin\left(\theta + \frac{\pi}{3}\right)} l^2 \tag{A.17}$$

$$D^{xy} = \frac{\frac{\sqrt{3}}{4}\sin\theta - \cos\theta\sin^2\theta}{\sin\left(\theta + \frac{\pi}{3}\right)} l^2 \tag{A.18}$$

Now considering (4), which is the update equation for A_3, A_3^{n+1} is replaced by a Taylor series expansion in time and B_1^n replaced according to (A.13) to obtain

$$\begin{aligned} & A + \Delta t A_t + \frac{\Delta t^2}{2} A_{tt} + O\left(\Delta t^3\right) \\ & \quad = \left(1 - C_\beta\right) A + C_\beta\left(DA + D^x A_x + D^{xx} A_{xx} + D^{xy} A_{xy} + D^{yy} A_{yy} + O\left(l^3\right)\right) \end{aligned} \tag{A.19}$$

Canceling A and replacing second-order time derivatives with spatial derivatives and omitting all higher-order terms for brevity, we obtain the modified equation

$$A_t - \frac{C_\beta D_x}{\Delta t} A_x = \left[\frac{C_\beta D_{xx}}{\Delta t} - \frac{U^2 \Delta t}{2}\right] A_{xx} + \frac{C_\beta D_{xy}}{\Delta t} A_{xy} + \frac{C_\beta D_{yy}}{\Delta t} A_{yy} \tag{A.20}$$

Considering each term separately, the effective advective speed is

$$\begin{aligned} U_{\text{eff}} & = -\frac{C_\beta D_x}{\Delta t} \\ & = -\frac{4U\sin\left(\theta + \frac{\pi}{3}\right)}{\sqrt{3}l}\left[-\frac{\sqrt{3}l}{4\sin\left(\theta + \frac{\pi}{3}\right)}\right] \\ & = U \end{aligned} \tag{A.21}$$

The longitudinal diffusion is

$$\begin{aligned} 2K^x &= \frac{C_\beta D_{xx}}{\Delta t} - \frac{U^2 \Delta t}{2} \\ &= \left[\frac{2}{\sqrt{3}} \left(\cos^2\theta \cos\left(\theta + \frac{\pi}{6}\right) + \sin\theta \cos^2\left(\theta - \frac{\pi}{3}\right) \right) - \frac{\sqrt{3}}{4 \sin\left(\theta + \frac{\pi}{3}\right)} \right] Ul - U^2 \Delta t \end{aligned} \tag{A.22}$$

The lateral diffusion is

$$\begin{aligned} 2K^y &= \frac{C_\beta D_{yy}}{\Delta t} \\ &= \frac{Ul \sin(3\theta)}{2\sqrt{3}} \end{aligned} \tag{A.23}$$

And the cross-diffusion coefficient is

$$\begin{aligned} 2K^{\text{cross}} &= \frac{C_\beta D_{xy}}{\Delta t} \\ &= Ul \left[\frac{4}{\sqrt{3}} \sin\theta \sin\left(\theta - \frac{\pi}{6}\right) \sin\left(\theta - \frac{\pi}{3}\right) \right] \end{aligned} \tag{A.24}$$

APPENDIX B: GRID GENERATION

The San Francisco Bay computational grids were generated using a variant of the advancing front method [30]. Although many methods for generating triangular meshes guarantee a Delaunay triangulation, the discretization used in SUNTANS and other Arakawa C-grid models places an additional constraint on the interior angles of cells. These models calculate gradients of cell-centered quantities under the assumption that the line between adjacent cell centers intersects and is perpendicular to the common edge. The consequence of this requirement is that cell centers must be the circumcenters of the triangles. For the circumcenter to fall within the cell itself, the cell must be an acute triangle. The primary motivation for development of a triangular grid orthogonal mesh generator is to provide automated creation of triangular meshes with all cells obeying the acute-angle constraints.

Inputs and preprocessing

Inputs to the mesh generation process are the following: (i) a polygon (possibly with holes) representing the shoreline and computational boundaries; (ii) a set of points and associated length scales giving the *requested* grid scales; and optionally (iii) a collection of polylines in the interior of the domain for controlling cell alignment. The grid scale refers to the nominal edge length and is defined everywhere within the domain on the basis of a natural-neighbors interpolation from the set of input points. Before grid creation can begin, the shoreline must be made consistent with the requested scale. This step ensures that features in the input that are ‘narrower’ than the size of the grid cells to be created are removed from the input. Starting with a constrained Delaunay triangulation of the input shoreline (Figure B.1a), Steiner points are added until the circumcenters lie near the medial axis of the shoreline. Specifically, for each constrained edge in the triangulation, we find the adjacent triangle internal to the domain and compare the circumradius of this triangle to the point-line distance from its center (the Voronoi point) to the constrained edge. If this ratio is greater than a prescribed threshold (empirically chosen to be 1.2), the edge is subdivided with a Steiner point inserted at its midpoint (Figure B.1b). The process is repeated until no more edges require subdividing. This resampling ensures that the circumradius reflects the length scales between features and not the length scales between successive vertices on the shoreline. Any triangles with a circumradius smaller than half the requested grid scale are removed, and a new shoreline polygon is constructed from the union of the remaining triangles (Figure B.1c).

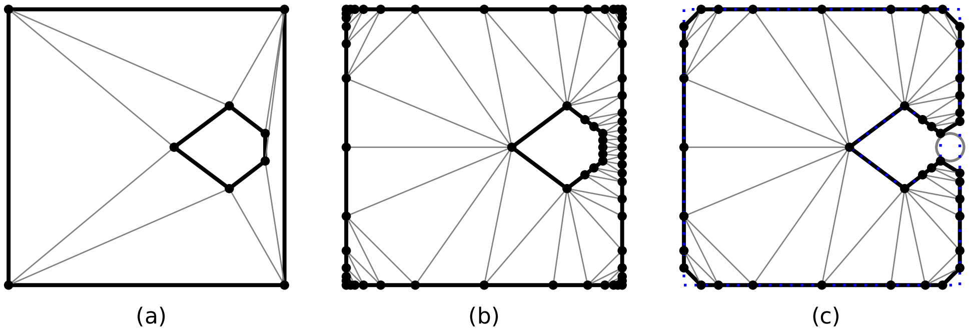

Figure B.1. Removal of features smaller than local grid scale. (a) Input shoreline and constrained Delaunay triangulation. (b) Constrained Delaunay triangulation after adding Steiner points. (c) Smoothed shoreline after removal of triangles with circumradius smaller than $l/2$ (illustrated by minimum circumcircle shown on the right).

Alignment of the grid is achieved by embedding a set of polylines in the interior of the domain. These lines are treated identically to shorelines, with the caveat that both sides of each line are internal to the domain and by the end of the grid generation each segment of the polyline will be adjacent to two cells.

Paving

In an advancing front method, the *front* is the boundary of the region in which no cells have been created, initially identical to the shoreline. Starting with the preprocessed shoreline polygon and any interior lines, the paving process iteratively applies the following steps, illustrated in Figure B.2:

1. Find the point along the front with the smallest internal angle, label it B, and keep a reference to the points A and C that are immediately clockwise and counterclockwise, respectively, from B along the front. This will be the site of the cell or cells created during this iteration.

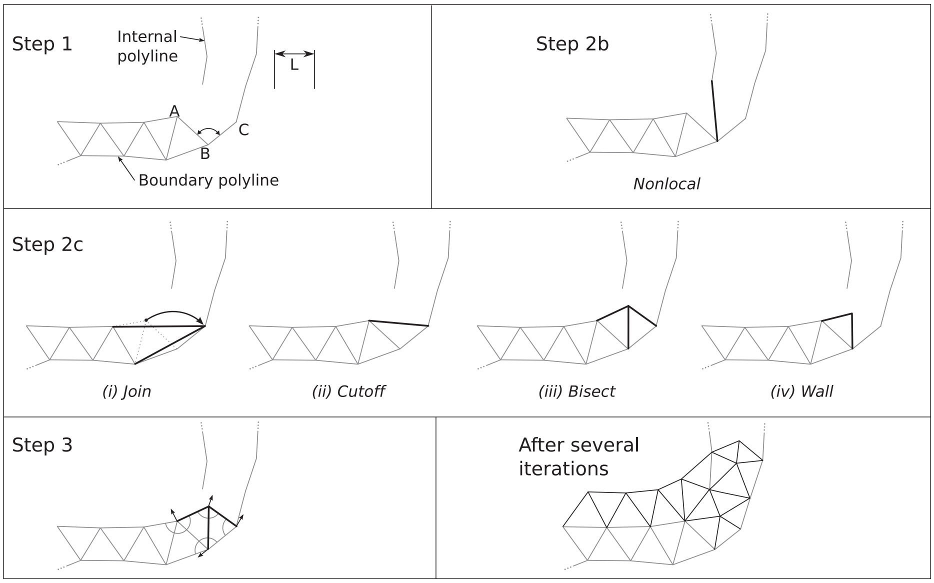

Figure B.2. One iteration of the paving algorithm. The node B is chosen on the basis of the smallest internal angle. The strategy for how to add edges and cells at B is chosen on the basis of the internal angle and whether the local scale is smaller than, equal to, or greater than the desired scale. After the topology is updated, node locations are refined by minimizing a cost function involving cell angles and edge lengths.

2. There are multiple choices or strategies for how new cells will be added at B. Topologically, these strategies correspond to how the degree of node B, the number of edges adjacent to B, is modified. For example, adding a line AC to create the cell ABC preserves the degree of B, whereas adding a new point D to form two new cells increases the degree by 1. In a perfectly regular triangular grid, all nodes have a degree of 6. Strategies that result in a greater degree will tend to create shorter edges, whereas the edge length can be increased by limiting the degree of B. In this way, the choice of strategy is the key mechanism by which a specific requested scale can be achieved. In many situations, though, local geometric constraints rule out some strategies, so it becomes necessary to construct a prioritized list of strategies based on the local scales and geometry. The steps in constructing the prioritized list are as follows:

   (a) Calculate a *scale factor* s as the ratio of the *current scale* (the average of edge lengths, $1/2\,(|AB| + |BC|)$) to the requested grid scale L.

   (b) Scan the vicinity around B for the possibility of nonlocal connections, that is, when the advancing front encounters an island, an opposing shoreline, or an internal polyline. If the nonlocal feature is within $1.25L$, the highest-priority strategy will be creating an edge from B to this feature. In the event that the nearest point on the nonlocal feature is not a preexisting vertex, a new vertex will be inserted in the feature at the point closest to B.

   (c) From both the interior angle $\theta = \angle ABC$ and the scale factor s, the remaining local strategies are prioritized. For example, if the angle were 60°, a *cutoff* will add the edge AC, making a roughly equilateral triangle ABC. The scale factor changes the thresholds between choosing one strategy over another. If the angle were instead 80° but the current scale were large compared with the requested scale, the *bisect* strategy would attempt to bisect the angle $\angle ABC$ by adding a new point D and creating two triangles ABD and BCD (and increasing the degree of B in an attempt to reduce the edge lengths). If the current scale were small compared with the requested scale, the cutoff strategy would be tried first, creating a single triangle ABC with longer edge lengths. Aside from nonlocal connections, the strategies (and the range of θ for which each is the highest-priority strategy) are as follows:

      i. Join: Merge AB and CB into a single edge [$\theta < 30°$].
      ii. Cutoff: Add the edge AC [$30° \leqslant \theta < 85° s^{-3/2}$].
      iii. Bisect: Add a new node D and edges AD, BD, and CD [$85° s^{-3/2} \leqslant \theta < 160° - 50°\,(s-1)$].
      iv. Wall: Add a new node D, but only create the edges AD and BD [$\theta \geqslant 160° - 50°\,(s-1)$].

3. For each attempted strategy, the mesh topology is updated, and a local optimization step adjusts vertex locations to restore the acute-angle constraints. The cost function minimized during this optimization is

$$C(p) = 10\left[\frac{\max_{i\in I}\left(|\phi_i - 60°|\right)}{\phi_{\max} - 60°}\right]^5 + \exp\left(\frac{\max_{i\in I}\left(\phi_i\right) - \phi_{\max}}{3°} - 1\right)\cdots$$
$$\cdots + 2\left(\left[\frac{\max_{e\in E}\left(l_{\mathrm{e}}\right)}{L}\right]^2 + \left[\frac{L}{\min_{e\in E}\left(l_{\mathrm{e}}\right)}\right]^2\right) \qquad \text{(B.1)}$$

where I denotes the set of internal angles ϕ_i of triangles that include the vertex p, $\phi_{\max}$ is the target maximum internal angle, chosen to be 85°, E the set of edges with p as an endpoint, and l_{e} the respective length of each of those edges. The first term favors equilateral triangles, the second term penalizes angles approaching $\phi_{\max}$, and the third term favors edge lengths close to L. If the optimization is not able to satisfy the angle constraints or the optimization causes a self-intersection, the mesh is reverted to its original state, and the next strategy is attempted.

Postprocessing

Although the paving process creates only cells that respect the interior angle criterion, poor-quality cells that only barely satisfy the angle requirements may still exist. Two postprocessing methods are applied to improve overall grid quality: relaxation and repaving. The relaxation steps apply the same optimization method used during paving to regions containing low-quality cells. By applying the optimization over a larger group of vertices, it is sometimes possible to nudge the vertices toward a better layout without modifying the grid topology. Where this approach is unable to improve the cells, the offending cell and its neighbors are removed, and the paving method is re-applied locally. Although repaving usually creates cells with better geometry, it may upset the alignment of cells that are not constrained by an adjacent shoreline or interior line. For the aligned grid created in this study, only the most borderline low-quality cells were selected for repaving in an effort to preserve as much of the alignment as possible.

Implementation and complexity

The current implementation of a triangular grid orthogonal mesh generator is primarily in Python, with broad utilization of the numerical libraries *numpy*, *CGAL*, and *scipy*. Although the asymptotic complexity is $O(n \log n)$, where n is the total number of cells, in practice, the optimization step is the most expensive, and for grids of at least up to 5×10^5 cells, the run time is very close to linear in n. On a standard 64-bit desktop computer, the speed ranges from approximately 40 cells/second in tightly constrained areas to 110 cells/second in broad, unconstrained areas.

REFERENCES

1. Casulli V, Walters R. An unstructured grid, three-dimensional model based on the shallow water equations. *International Journal for Numerical Methods in Fluids* 2000; **32**:331–348.
2. Fringer OB, Gerritsen M, Street R. An unstructured-grid, finite-volume, nonhydrostatic, parallel coastal ocean simulator. *Ocean Modelling* 2006; **14**:139–173. DOI: 10.1016/j.ocemod.2006.03.006.
3. Chen C, Beardsley R, Cowles G. An unstructured-grid, finite-volume coastal ocean model system. *Advances in Computational Oceanography* 2006; **19**(1):78–89.
4. Hirt C. Heuristic stability theory for finite-difference equations. *Journal of Computational Physics* 1968; **2**:339–355.
5. Warming R, Hyett B. The modified equation approach to the stability and accuracy analysis of finite-difference methods. *Journal of Computational Physics* 1974; **14**(2):159–179.
6. Bouche D, Ghidaglia JM, Pascal F. Error estimate and the geometric corrector for the upwind finite volume method applied to the linear advection equation. *SIAM Journal of Numerical Analysis* 2005; **43**(2):578–603. DOI: 10.1137/040605941.
7. Turkel E. Accuracy of schemes with nonuniform meshes for compressible fluid flows. *Applied Numerical Mathematics* 1986; **2**:529–550.
8. Chua VP, Fringer OB. Sensitivity analysis of three-dimensional salinity simulations in North San Francisco Bay using the unstructured-grid SUNTANS model. *Ocean Modelling* 2011; **39**:332–350. DOI: 10.1016/j.ocemod.2011.05.007.
9. Durran D. *Numerical Methods for Fluid Dynamics*, 2nd edn, Texts in Applied Mathematics. Springer: Heidelberg, 2010.
10. Godunov S. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik (N.S.)* 1959; **47(89)**(3):271–306.
11. LeVeque R. *Finite Difference Methods for Ordinary and Partial Differential Equations*. Society for Industrial and Applied Mathematics: Philadelphia, 2007.
12. Taylor GI. Dispersion of soluble matter in solvent flowing slowly through a tube. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 1953; **219**:186–203. DOI: 10.1098/rspa.1953.0139.
13. Sweby P. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal on Numerical Analysis* 1984; **21**(5):995–1011.
14. Namin M, Lin B, Falconer R. Modelling estuarine and coastal flows using an unstructured triangular finite volume method. *Advances in Water Resources* 2004; **27**(12):1179–1197.
15. Darwish M, Moukalled F. TVD schemes for unstructured grids. *International Journal of Heat and Mass Transfer* 2003; **46**:599–611.
16. Alouges F, Coq G, Lorin E. Two-dimensional extension of the reservoir technique for some linear advection systems. *Journal of Scientific Computing* 2007; **31**(3):419–458. DOI: 10.1007/s10915-006-9115-7.
17. Owen S. A survey of unstructured mesh generation technology. In *Proceedings of the 7th International Meshing Roundtable*, Albuquerque, NM, USA, 1998; 239–267.

18. Jasak H. Error analysis and estimation for the finite volume method with applications to fluid flows. *Ph.D. Thesis*, Imperial College of Science, Technology and Medicine, 1996.
19. Tam A, Ait-Ali-Yahia D, Robichaud M, Moore M, Kozel V, Habashi W. Anisotropic mesh adaptation for 3D flows on structured and unstructured grids. *Computer Methods in Applied Mechanics and Engineering* 2000; **189**:1205–1230.
20. Lagoutière F, Després B. Genuinely multi-dimensional non-dissipative finite-volume schemes for transport. *International Journal of Applied Mathematics and Computer Science* 2007; **17**(3):321–328. DOI: 10.2478/v10006-007-0026-z.
21. Aris R. On the dispersion of a solute in a fluid flowing through a tube. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 1956; **235**:67–77.
22. Mellor G, Yamada T. Development of a turbulence closure-model for geophysical fluid problems. *Reviews of Geophysics* 1982; **20**(4):851–875. DOI: 10.1029/RG020i004p00851.
23. Large W, Pond S. Open ocean momentum flux measurements in moderate to strong winds. *Journal of Physical Oceanography* 1981; **11**:324–336.
24. An estimate of daily average delta outflow. (Available from: http://www.water.ca.gov/dayflow/) [Accessed May 1, 2011].
25. USGS water quality monitoring. (Available from: http://sfbay.wr.usgs.gov/access/wqdata) [Accessed May 1, 2011].
26. Burchard H, Rennau H. Comparative quantification of physically and numerically induced mixing in ocean models. *Ocean Modelling* 2008; **20**(3):293–311. DOI: 10.1016/j.ocemod.2007.10.003.
27. Willmott CJ, Ackleson SG, Davis RE, Feddema JJ, Klink KM, Legates DR, O'Donnell J, Rowe CM. Statistics for the evaluation and comparison of models. *Journal of Geophysical Research* 1985; **90**(C5):8995–9005.
28. Fischer HB, List EJ, Koh RC, Imberger J, Brooks NH. *Mixing in Inland and Coastal Waters*. Academic Press, Inc.: New York, 1979.
29. Marcuzzi F, Cecchi M, Venturin M. An anisotropic unstructured triangular adaptive mesh algorithm based on error and error gradient information. *Mathematics and Computers in Simulation* 2008; **78**(5–6):645–652. DOI: 10.1016/j.matcom.2008.04.006.
30. George P, Seveno E. The advancing-front mesh generation method revisited. *International Journal for Numerical Methods in Engineering* 1994; **37**:3605–3619.