Noname manuscript No.
(will be inserted by the editor)

Cell-vertex discretization of shallow water equations on mixed unstructured meshes

S. Danilov · A. Androsov

the date of receipt and acceptance should be inserted later

Abstract Finite-volume discretizations can be formulated on unstructured meshes composed of different polygons. A staggered cell-vertex finite-volume discretization of shallow water equations is analyzed on mixed meshes composed of triangles and quads. Although triangular meshes are most flexible geometrically, quads are more efficient numerically and do not support spurious inertial modes of triangular cell-vertex discretization. Mixed meshes composed of triangles and quads combine benefits of both. In particular, triangular transitional zones can be used to join quadrilateral meshes of differing resolution. Based on a set of examples involving shallow water equations it is shown that mixed meshes offer a viable approach provided some background biharmonic viscosity (or the biharmonic filter) is added to stabilize the triangular part of the mesh.

Keywords Unstructured mixed meshes · finite-volume discretization · shallow water equations · filter operators · numerical stability · tides

S. Danilov
Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research, Postfach 12-01-61, 27515 Bremerhaven, Germany, and A. M. Obukhov Institute of Atmospheric Physics RAS, Pyzhevsky per., 3, 119017 Moscow, Russia
Tel.: +4947148311764
E-mail: Sergey.Danilov@awi.de

A. Androsov
Alfred Wegener Institute Helmholtz Centre for Polar and Marine Research, Postfach 12-01-61, 27515 Bremerhaven, Germany
Tel.: +4947148312106
E-mail: Alexey.Androsov@awi.de

1 Introduction

Unstructured meshes are now common in coastal ocean modeling, and the number of models based on such meshes is continuously increasing. Models like FVCOM (Chen et al. (2003)), ADCIRC (Westerink et al. (1992)) or SELFE (Zhang and Baptista (2008)), to mention just few, are widely used by many research groups for various practical tasks. Indeed, when it comes to resolving the details of coasts or estuaries, there is no viable alternative in a general case (although structured curvilinear meshes can provide accurate and more numerically efficient solutions in particular cases). Most unstructured-mesh models are formulated on triangular meshes simply because triangles offer more geometrical flexibility than other polygons. For example, quads have to be substantially deformed in places where the resolution is varied. Quads are on the other hand more economical computationally, since they involve fewer edges. Many finite-volume codes can easily be generalized to mixed meshes, in particular, to the meshes made of quads and triangles, enabling using the best sides of both elements where appropriate. While the approach is common in computational fluid dynamics, its applications to oceanographic tasks are rare. Kernkamp et al. (2011) propose to use a C-grid-based approach on general mixed meshes and explain why it may be beneficial in coastal applications. In particular, on C-grids, quads are better suited to represent one-dimensional structures or can more easily be aligned with the flow or bathymetry where needed. However, C-grid meshes should be orthogonal, which makes them less general. Additionally, special measures are needed to represent the Coriolis operator in situations when its effects are not negligible. Relevant discussion of the reconstruction procedure needed to implement the operator can be found in Thuburn et al. (2009) who propose the approach that ensures the presence of a stationary geostrophic mode on an f-plane. The divergence noise may be another issue on triangular C-grids (see Le Roux et al. (2007)) in applications where baroclinicity is important.

This may partly serve as an argument in favor of exploring a cell-vertex finite-volume discretization, which keeps a full velocity vector at cell centroids and locates scalar quantities at the vertices. The presence of full velocity makes the treatment of the Coriolis term trivial, and the discretization does not require orthogonal meshes. Admittedly, C-grids are more accurate in simulating the propagation of the Poincaré waves, and would be a better choice in cases when the interest lies solely in computing barotropic flows. However, the triangular cell-vertex discretization works sufficiently well in practice, as demonstrated by numerous studies performed with FVCOM. It is not free of numerical modes, and we will address them below. We will deal here only with the shallow-water equations both for simplicity and because the vertical dependence does not interfere with the mixed character of the horizontal mesh.

The numerical modes of triangular cell-vertex discretization are one of main factors motivating our attention to mixed meshes. Indeed, this discretization (analogous to $P_0 - P_1$ finite-element discretization) has too many degrees of freedom in its velocity field because the number of cells is approximately twice that of vertices. The excessive size of velocity space results in spurious inertial modes (see, e. g., LeRoux (2012)). These modes are of zero divergence in a linearized, uniform-depth shallow water system. They do not project on the elevation and are eliminated by viscosity. In practice, however, nonlinearity, uneven bottom topography or forcing continuously generate them, and because of nonlinearity, the solution is aliased by noise unless dissipation is sufficient to remove it. Although the noise can be well handled (see, e. g., Danilov (2012)), it it always of concern. Quads, in contrast, implement the necessary balance between the vector (velocity) and scalar (elevation) degrees of freedom and are free of the noise discussed above (a caveat is that on structured meshes the cell-vertex discretization is similar to a B-grid and may be prone to the pressure modes, as documented elsewhere).

The other factor in favor of mixed meshes is the already mentioned fact that quads are more efficient computationally. Indeed, compared to triangular meshes with the same number of scalar degrees of freedom, they have twice fewer cells and 1.5 times fewer edges. This leads to shorter cycles and a reduced computational burden. Clearly, in a close proximity to coasts one may need the versatility of triangular meshes for geometrical reasons. Outside these areas the triangular part may only be needed to provide smooth transition between quadrilateral meshes of differing resolution. This is very similar to the idea of nesting, and is perhaps of more value for modeling the large-scale ocean circulation than for the applications on a coastal scale.

The point of concern is that the areas of triangles adjacent to quads are about 2 ($4/\sqrt{3}$ for equilateral triangles)) times smaller, which can be thought as leading to a local loss of approximation for flow features on a grid scale. However, this jump in resolution involves only the velocities, since the difference in the size of scalar control volumes is much smaller ($(4/3)^{1/2} \approx 1.15$ if triangles are equilateral). One may therefore argue that the factor of about 2 for velocity is rather the consequence of the increased velocity space than of the jump in actual resolution offered by the discretization, and extra degrees of freedom in velocities should effectively be filtered out. The consequences of an abrupt change in mesh geometry depend on how well the mesh is able to resolve the dynamics. For many wave applications the approach proposed here may work just fine. However, if nonlinear effects are significant, or, in other words, the grid-scale Reynolds number is sufficiently high, the transitional zones may be vulnerable to noise, especially on the triangular part. For this reason the reliability of mixed meshes depends on whether we are able to control the noise in the case it is generated. That said, it should be clear that the transition we are dealing with here is much more gradual than the jump in resolution in traditional nesting, and fields are always consistent across the coupling zone if dissipation operators are properly adjusted. This paper is therefore about noise on mixed meshes and on how to control it by relatively simple means. Our main goal is to show that it is possible through efficient viscosity operators.

The plan of the paper is as follows. The next section briefly describes the implementation, and focuses on the details of viscosity and filter operators needed to maintain stability of too large velocity space on triangles without excessive damping. The first-order upwind in momentum advection used in some coastal codes to ensure numerical stability is kept as an option, but we consider it overly dissipative. We discuss other implementations of momentum advection, including a variant with build-in filtering, which may be helpful in practical situations. Section 3 deals with numerical tests of wave and tide propagation both in idealized examples and in realistic cases, where the comparison against observations is carried out. Neither of the cases considered here is intended to demonstrate advantages of mixed meshes in terms of accuracy; they rather serve to illustrate that the approach works stably. Section 4 continues with tests, but concerned with the quasi-geostrophic part of dynamics, and the final section concludes. For completeness, Appendix proposes the derivation of energy and potential vorticity conserving scheme, which is based on the vector-invariant form of momentum advection.

2 Implementation, viscosity, filters and momentum advection

Here we deal with the shallow water equations

$$\partial_t \mathbf{u} + f\mathbf{k}\times\mathbf{u} + (\mathbf{u}\cdot\nabla)\mathbf{u} + g\nabla\eta = \mathbf{F} + (D_d + D_v)\mathbf{u}, \quad (1)$$

$$\partial_t \eta + \nabla\cdot(H\mathbf{u}) = 0. \quad (2)$$

The notation is standard: $\mathbf{u} = (u, v)$ is the velocity, f the Coriolis parameter, $\mathbf{k}$ the unit vertical vector, g the acceleration due to gravity, η the elevation, $H = h + \eta$ the full fluid layer thickness and h the unperturbed thickness, which may vary horizontally due to uneven bottom. The operators D_d and D_v introduce the bottom drag and viscosity respectively, and $\mathbf{F}$ is the external force (due to wind). The bottom drag is written as $HD_d\mathbf{u} = -C_d|\mathbf{u}|\mathbf{u}$, where C_d is the drag coefficient. The viscosity operator can be both harmonic and biharmonic, and we will also consider the option when it is replaced by a numerical filter. Equations (1) and(2) can be combined to provide the flux form for momentum advection

$$\partial_t\mathbf{U} + f\mathbf{k}\times\mathbf{U} + \nabla\cdot(\mathbf{u}\mathbf{U}) + gH\nabla\eta = H\mathbf{F} + H(D_d + D_v)\mathbf{u}, \quad (3)$$

where $\mathbf{U} = \mathbf{u}H$.

We briefly describe the implementation in plane geometry with coordinates (x, y). Spherical geometry is introduced by changing to a spherical coordinate system with poles over the land. The cosine of latitude is estimated in that case at cell centroids, where the velocity is located. It is used to define the advecting velocity (see, e. g., Szmelter and Smolarkiewicz (2010)) in computations of divergence, and to compute the gradients of scalar quantities on cells. Metric differentiation terms only appear in the computations of velocity gradients, and are estimated on cells if needed. They are omitted in all examples below because of their smallness.

As is common with finite-volume method, the governing equations are integrated over control volumes and the divergence terms, by virtue of the Gauss theorem, are reduced to the sums of respective fluxes through the boundaries of control volumes. The scalar control volumes are formed by connecting the cell centroids with the centers of edges, which gives the so-called median-dual control volumes, as schematically shown in Fig. 1. The vector control volumes are the mesh cells themselves. The basic structure to describe the mesh is the array of edges given by their vertices v_1 and v_2, and the array of two pointers c_1 and c_2 to the cells on the left and on the right of the edge. Using this language, there is no difference between triangles and quads in the cycles which assemble fluxes. Quads and triangles are described as four indices to vertices forming them; in the case of triangles the fourth index equals the first one. The treatment of triangles and quads differs slightly in what concerns computations of gradients, which will be detailed below. We will use a symbolic notation: $e(c)$ for the list of edges forming cell c, $e(v)$ for the list of edges connected to vertex v, $v(c)$ for the list of vertices defining cell c and so on. We will use $\overline{v}(c)$ to denote non-repeating vertices of triangles ($v(c)$ lists the first vertex twice). As concerns the time stepping implementation, we use the methods described in Maßmann et al. (2010). To facilitate reading we concentrate on the AB3-AM4 (Adams–Bashforth Adams–Multon) time stepping as recommended in Shchepetkin and McWilliams (2005).

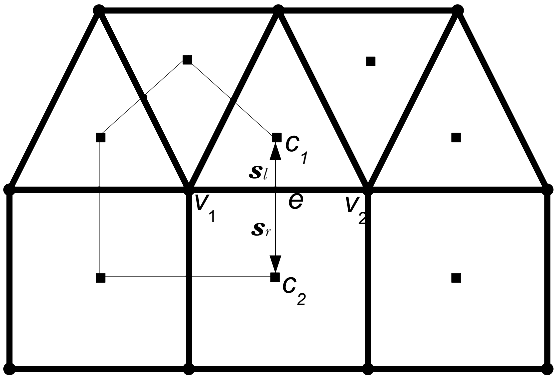

Fig. 1 Schematic of mesh structure. The velocities are located at centroids (the squares) and the elevation at vertices (the large circles). A scalar control volume associated with vertex v_1 is formed by connecting the neighboring centroids to edge centers (the small circles). The lines passing through two neighboring centroids (e. g., c_1 and c_2) are broken in a general case at the edge centers. Their fragments are described by the left and right vectors directed to centroids ($\mathbf{s}_l$ and $\mathbf{s}_r$ for edge e). Edge e is defined by its two vertices v_1 and v_2 and is considered to be directed to the second vertex. It is also characterized by two cells c_1 and c_2 to the left and to the right respectively.

2.1 AB3-AM4 time discretization

First, we advance the elevation

$$\eta^{n+1} - \eta^n = -\tau\nabla \cdot (H\mathbf{u})^{AB3}. \quad (4)$$

Next the momentum equation is updated

$$\mathbf{u}^{n+1}H^{n+1} - \mathbf{u}^n H^n + \tau C_d|\mathbf{u}^n|\mathbf{u}^{n+1} = -\tau(gH\nabla\eta)^{AM4} + \tau(-Hf\mathbf{k}\times\mathbf{u} - \nabla\cdot(\mathbf{U}\mathbf{u}) + HD_v\mathbf{u} + H\mathbf{F})^{AB3}. \quad (5)$$

Here τ is the time step, AB3 stand for

$$\eta^{AB3} = (3/2+\beta)\eta^n - (1/2+2\beta)\eta^{n-1} + \beta\eta^{n-2}, \quad (6)$$

in the case of elevation, and the same rule applies for the velocity. The elevation gradient in the momentum equation is estimated as

$$\eta^{AM4} = \delta\eta^{n+1} + (1-\delta-\gamma-\varepsilon))\eta^n + \gamma\eta^{n-1} + \varepsilon\eta^{n-2}. \quad (7)$$

In the standard AB3, $\beta = 5/12$, but the choice $\beta = 0.281105$ warrants better stability; the other parameters are $\gamma = 0.088$, $\delta = 0.614$, and $\varepsilon = 0.013$ (Shchepetkin and McWilliams (2005)).

This implementation requires additional storage to keep three time slices of fields, and one more set to keep the AB3 interpolated values.

We also consider a simplified approach which is based on (1) instead of (3). In this case

$$\mathbf{u}^{n+1} - \mathbf{u}^n + \tau C_d|\mathbf{u}^n|\mathbf{u}^{n+1}/H^{n+1} = -\tau(g\nabla\eta)^{AM4} + \tau(-f\mathbf{k}\times\mathbf{u} - (\mathbf{u}\cdot\nabla)\mathbf{u} + D_v\mathbf{u} + \mathbf{F})^{AB3}, \quad (8)$$

which is more efficient numerically in cases it is relevant.

2.2 Divergence and gradients

The divergence operator on scalar control volumes is computed as

$$\int_v \nabla\cdot(H\mathbf{u})dS = \sum_{e=e(v)} [(H\mathbf{u}\cdot\mathbf{n}l)_l + (H\mathbf{u}\cdot\mathbf{n}l)_r]_e,$$

where the cycle is over edges containing vertex v, the indices l and r imply that the estimates are made on the left and right segments of the control volume boundary attached to the center of edge e (connecting the center of edge e to c_1 and c_2 in Fig. 1), $\mathbf{n}$ is the outer normal and l the length of the segment, and H is the estimate of fluid thickness on segments. Introducing vectors $\mathbf{s}_l$ and $\mathbf{s}_r$ connecting the mid-point of edge e with the cells on the left and on the right, we get $(\mathbf{n}l)_l = \mathbf{k}\times\mathbf{s}_l$ and similarly, but with the minus sign for the right cell ($\mathbf{k}$ is the unit vertical vector). The velocities are taken on their locations, and the only remaining issue is the estimate of thickness. The simplest estimate is the central one, $H = H_e = \sum_{v=v(e)} H_v/2$. It, however, does not ensure that the operator of divergence is the negative adjoint of the gradient operator in the energy norm. To achieve this, one needs to use the mean cell values, $H = H_c = \sum_{v=\overline{v}(c)} H_v/3$ on triangles and $H = H_c = \sum_{v=v(c)} H_v w_{cv}$ on quads, where $w_{cv} = S_{cv}/S_c$ is the area weight, with S_c the cell area and S_{cv} the part of it in the scalar control volume around vertex v. In practice, however, both implementations lead to very similar results despite the theoretical energy inconsistency associated to the former. Note that, as is common in ocean modeling, there is no upwinding here: the thickness equation originates from the continuity equation, so no dissipation is relevant. However, this is overridden by the upwind estimate if wetting and drying is simulated, which is implemented following Stelling and Duinmeijer (2003).

Gradients of scalar quantities are needed on cells, and are computed as

$$\int_c \nabla\eta dS = \sum_{e=e(c)} (\mathbf{n}l\eta)_e,$$

where summation is over the edges of cell c, the normal and length are related to the edges, and η is estimated as the mean over edge vertices. The gradients of velocities are also needed on cells, they enter computations of the viscosity term, but also the computations of the momentum advection if the form (8) or the upwind version of (5) are used. In this case the least squares fit is used for a linear velocity reconstruction based on the neighboring values. The neighbor cells are those sharing edges with the given cell. Their list will be denoted by $n(c)$ (it contains empty places at the lateral boundary). If the horizontal velocities at these cells are $\mathbf{u}_n$, the derivatives of u component of velocity are found by solving

$$\mathcal{L} = \sum_{n=n(c)} (u_c - u_n - (\alpha_x,\alpha_y)\cdot\mathbf{r}_{cn})^2 = \min.$$

Here $\mathbf{r}_{cn} = (x_{cn}, y_{cn})$ is the vector connecting the center of c to that of its neighbor n. One differentiates $\mathcal{L}$ by α_x and α_y and on equating these derivatives to zero obtains a system of two linear equations on α_x and α_y. Their solution can be reformulated in terms of two sets of coefficients a_{cnx} and a_{cny} which give the components of the gradient after multiplication with the respective velocity differences:

$$a_{cnx} = (x_{cn}Y^2 - y_{cn}XY)/d,$$
$$a_{cny} = (y_{cn}X^2 - x_{cn}XY)/d,$$
$$d = X^2Y^2 - (XY)^2.$$

Here $X^2 = \sum_{n=n(c)} x_{cn}^2$, $Y^2 = \sum_{n=n(c)} y_{cn}^2$ and $XY = \sum_{n=n(c)} x_{cn}y_{cn}$. The coefficients are computed at the initialization phase and stored for each cell. The derivatives of v component of velocity are computed analogously. On the boundaries we apply the concept of ghost cells (mirror reflections with respect to boundary edge) and compute the velocity on these cells either as $\mathbf{u}_n = -\mathbf{u}_c$ for the no-slip case, or as $\mathbf{u}_n = -\mathbf{u}_c + 2(\mathbf{u}_c \cdot \mathbf{n}_{nc})\mathbf{n}_{nc}$ in the free-slip case. Here n is the index of the ghost cell, and $\mathbf{n}_{nc}$ is the vector of unit normal to the edge at the boundary of cell c. Note that this alone is insufficient to implement the free-slip condition and one has to explicitly eliminate the tangent component of viscous stress flux at the boundary to warrant that this condition is observed on general meshes.

To treat trangles and quads similarly, the gradient arrays operate with four coefficients, the fourth being zero for triangles, and the fourth neighbor being the cell itself. This increases, but only slightly, the number of floating point operations on the triangular part of the mesh compared to a purely triangular case. Notice also that to store the gradient coefficients one needs more space than for all the fields involved in the AB3-AM4 time stepping. It will be different in the 3D case because the gradient coefficients will be the same for all layers.

2.3 Implementation of momentum advection

In the case the form (8) is used, the momentum advection term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ is easily computed after the velocity gradients are known on each cell. This form is more efficient computationally than two other forms to be discussed below, and is preferred in cases it is sufficient. Sometimes one needs to introduce either upwinding or smoothing, and in these cases the other, flux form is more convenient. Variant A provides the upwind implementation. In this case

$$\int_c \nabla\cdot(\mathbf{u}H\mathbf{u})dS = \sum_{e=e(c)} (\mathbf{u}\cdot\mathbf{n}lH\mathbf{u})_e,$$

where the summation is over the cell edges. For edge e, linear velocity reconstructions (known after the gradients are computed) on the cells on its both sides are estimated at the edge center. One of the cells is c, and let n be its neighbor across e. The respective velocity estimates will be denoted as $\mathbf{u}_{ce}$ and $\mathbf{u}_{ne}$. One takes $\mathbf{u}\cdot\mathbf{n} = \mathbf{n}\cdot(\mathbf{u}_{ne}+\mathbf{u}_{ne})/2$, and $2\mathbf{u} = \mathbf{u}_{ce}(1+\text{sign}(\mathbf{u}\cdot\mathbf{n}))+\mathbf{u}_{ne}(1-\text{sign}(\mathbf{u}\cdot\mathbf{n}))$. This corresponds to gentle upwinding. The other form $2\mathbf{u} = \mathbf{u}_c(1+\text{sign}(\mathbf{u}\cdot\mathbf{n}))+\mathbf{u}_n(1-\text{sign}(\mathbf{u}\cdot\mathbf{n}))$ is the standard first-order upwind and proves to be too dissipative in many practical cases. Yet it may be needed if one deals with bores.

The other form (variant B) is adapted from Danilov (2012) and provides additional spatial filtering. In this case one first estimates the momentum flux term on scalar control volumes:

$$\int_v \nabla\cdot(\mathbf{u}H\mathbf{u})dS = \sum_{e=e(v)} ((\mathbf{u}\cdot\mathbf{n}lH\mathbf{u})_l + (\mathbf{u}\cdot\mathbf{n}lH\mathbf{u})_r)_e,$$

and the notation follows that for the divergence. No velocity reconstruction is involved. These estimates are then averaged to the centers of cells. In both variants the fluid thickness is estimated at the cell centers.

We note that these variants do not exhaust all possibilities. One more useful form is obtained by rewriting the momentum advection in (8) in the vector-invariant form. It is addressed in the Appendix where it is shown to lead to an energy and potential vorticity conserving implementation. Its other advantage is natural smoothing because it operates with vorticity and kinetic energy defined at scalar points, and in this way removes excessively small scales that are present in the variant A. This form is rather similar to that of Ringler and Randall (2002), with an obvious difference stemming from the median-dual control volumes in our case instead of the Voronoi cells. Simulations reported below rely on (8) and the momentum advection in the form $(\mathbf{u}\cdot\nabla)\mathbf{u}$.

2.4 Viscosity, biharmonic viscosity and filtering

The major drawback of cell-vertex discretization is the excessively large size of the velocity space on a triangular mesh. Keeping the ensuing velocity noise under control requires special attention to the implementation of viscosity operator. Consider the harmonic viscosity operator $D_v\mathbf{u} = \nabla\cdot(A\nabla\mathbf{u})$, where A is the viscosity coefficient. We use this form in (8), and update it to $HD_v\mathbf{u} = \nabla\cdot(Ah\nabla\mathbf{u})$ in (5). Although it does not provide a correct (invariant under rotations) expression for a variable viscosity A, it works well in practice and we prefer it for the reasons to be clarified further. Its computation follows the rule

$$\int_c \nabla\cdot(A\nabla\mathbf{u})dS = \sum_{e=e(c)} (Al\mathbf{n}\cdot\nabla\mathbf{u})_e.$$

The estimate of velocity gradient on edge e is symmetrized, following the standard practice, over the values on the neighboring cells. The consequence of this symmetrization is that on regular (formed of equilateral triangles or rectangular quads) meshes the information from the nearest neighbors will be lost. The implication is that any singularity in velocity between the nearest cells will not be penalized. Although unfavorable for both quads and triangles, it has far reaching implications for the latter: it cannot efficiently remove the velocity noise at the smallest scale. This fact is well known and the modification needed to increase the coupling is to write the identity

$$\mathbf{n} = \mathbf{r}_{cn}/|\mathbf{r}_{cn}| + (\mathbf{n} - \mathbf{r}_{cn}/|\mathbf{r}_{cn}|),$$

for the normal vector, with $\mathbf{r}_{cn}$ the vector connecting the centroids of cell c and neighboring cell n, and replace

$$\mathbf{n} \cdot \nabla \mathbf{u}$$

with

$$(\mathbf{u}_n - \mathbf{u}_c)/|\mathbf{r}_{cn}| + (\mathbf{n} - \mathbf{r}_{cn}/|\mathbf{r}_{cn}|) \cdot \nabla \mathbf{u}.$$

The first term here introduces coupling between the neighbors. In it easy to see that on rectangular quads (equilateral triangles) the second term will disappear, and one will arrive at the standard five-(four-) point estimate of the Laplacian operator for constant A. However, on general meshes the second term cannot be discarded, so that the full expression has to be estimated. This makes the computation of viscous term rather expensive. The computation of true viscous operator would be even more expensive, which is precisely the reason for keeping the reduced form $\nabla \cdot (A\nabla \mathbf{u})$. Note that this difficulty can partly be alleviated by storing the coefficients of 'corrected' harmonic operators (and indices of cells involved) for each cell. We did not explore this possibility here leaving it to future 3D implementations.

The modified viscosity performs well, but may lead to too high dissipation if applied to fully eliminate the velocity noise (the code might remain stable for much lower values of A). In these cases the biharmonic viscosity does a much better job. We implement it by computing the Laplacian first, multiplying it with the (negative) biharmonic viscosity coefficient (appropriately scaled with the cell size) and computing the Laplacian once again. It works very efficiently, yet the drawback is that it is even more expensive, generally taking as much CPU time as the rest of the code. Recognizing this fact we looked for perhaps suboptimal, but cheaper solutions. This is done by introducing a weak filtering implemented in a time-stepping mode. The simplest filter is implemented by adding to the rhs of the momentum equation the term (we will consider (8) for simplicity)

$$\mathbf{F}_c = -(1/\tau_f) \sum_{n(c)} (\mathbf{u}_n - \mathbf{u}_c),$$

with τ_f a time scale to be selected experimentally. Two remarks are due here. First, on ideal (rectangles or equilateral triangles) meshes the expression above provides the discretization of the Laplacian operator. Indeed, it is easy to see that $\mathbf{F}_c \approx (a^2/3\tau_f)(\partial_{xx} + \partial_{yy})$ on triangles and 3 times that on quads, so that τ_f of about 1 day corresponds to viscosity of about 10^3 m^2/s on a mesh with a side $a = 10$ km. On irregular meshes this approximation fails, and little can be said about the type of the equivalent operator at the grid scale. Ringler and Randall (2002), discussing the ZM discretization which is analogous to the cell-vertex one analyzed here, suggest to use the Laplacian operator computed on the stencil of the nearest neighbors. It is possible by computing the vector Laplacian as $\nabla\nabla \cdot - \nabla \times \nabla \times$, but this does not ensure local approximation of the Laplacian unless the mesh is regular. In principle, one can implement it as the filter by introducing coefficients with each contribution in the expression for $\mathbf{F}_c$ above, which correspond to those of the nearest-neighbor Laplacian. This way of implementing the filter would be more mathematically consistent. We follow here the simplest approach and will explore the implications of the other one in a future work. Second, same as with viscosity, one may introduce a 'biharmonic' filter by performing the same operations as in $\mathbf{F}_c$ (but without multiplication with τ_f^{-1}) on the field of $-\mathbf{F}_c$ and adding the result to the rhs. We have found that this option is much more efficient than just $\mathbf{F}_c$ in removing noise in the velocity field, and is much cheaper computationally than biharmonic viscosity. In many cases it is sufficient, but care should be taken in the vicinity of boundaries. Currently, we do the cycle over the internal edges only, which is roughly equivalent to the no-slip boundary condition. In principle, on edges forming the boundary, computations can be carried out with the same concept of ghost cells, yet it is not necessarily stable. Another issue is the factor 1/3 in the Taylor expansion above for triangles. Since triangles need more, and not less, damping some scaling of $1/\tau_f$ is needed on triangles.

To conclude this subsection we note that viscosities and inverse time factors in filters must be scaled with the geometrical size of mesh cells in a general case. Scalings are task-dependent.

3 Numerical simulations

We will consider several cases which altogether give some idea of the versatility of mixed meshes. We begin with a simple case of a high-amplitude wave propagating in a narrow channel on a mesh composed of triangles and quads, to illustrate in an elementary way the issue of noise control. We continue with more practical examples dealing with tides in the North Sea on different meshes, and in the Strait of Messina on a curvilinear mesh. Here the intention is to show that the code easily handles meshes of different type. We also consider a wind driven flow in a rectangular box as an example of balanced flows. The barotropic shallow water dynamics stay very far from 3D dynamics in a stratified fluid, so our goal here is once again only to show on a qualitative level that solutions in transitional zones remain smooth despite the turbulent character of dynamics.

3.1 Wave in a channel and computational efficiency

Consider a channel mesh made of quads with a side of 0.02 degree. The channel occupies a belt from 0 to 2 degrees and is 20 degrees long. The cosine of latitude is set to one, but the small Coriolis parameter is taken into account. The quad mesh is then made triangular between 0 and 4 degrees in zonal direction by splitting quads into triangles in a regular way. The area of scalar control volumes does not change on passing from triangles to quads, the vector control volumes on triangles are smaller by a factor of 2. The initial perturbation is a single period of a sine wave of amplitude 10 m located on the triangular part of the mesh between 0 and 3 degree, with initial velocities corresponding to the eastward propagation. The unperturbed fluid depth is 500 m. Figure 2 shows the zonal velocity (in m/s) in the main wave (top panel) and the zonal velocity patterns behind the wave (three rows of panels) at $t = 10^4$ s when the wave is entirely on the quadrilateral part of the mesh. The colorbar in row panels is adjusted to visualize the small-amplitude part of the signal. The wave steepens as it travels east, leaving behind a small-amplitude (spurious) wave train because of numerical dispersion, and a perturbation of Kelvin wave type linked to the Coriolis term. The position and shape of the main wave are predicted accurately, but our focus here is on the noise. Viscosity and filters are off in the case shown in the first row, and there is velocity noise on triangles (with a 'triangular checkerboard' pattern), which makes visible the boundary between triangular and quadrilateral cells at x=4 degrees. The biharmonic filter removes the noise (the second row), and simultaneously smoothes the part of the signal associated to dispersive errors. Finally the third row of panels corresponds to simulations performed with the same biharmonic filter on a mesh obtained by randomly displacing the vertices of the previous mesh by up to 35% of the cell size. In this case the cell areas strongly vary, and small-amplitude noise would be seen in the wave wake in both the velocity and elevation if simulations were run without the filter. However, it is eliminated if the filter is on, as demonstrated here. There are some differences in the patterns shown in the middle and bottom panels despite the use of the same filter, which can be attributed to the difference in the meshes. The amplitude of the main wave is hardly affected in the three cases shown, only the fourth significant digit is modified. We conclude that the biharmonic filter is generally necessary, and is efficient in keeping the noise under control.

The computational efficiency is examined by comparing the CPU time needed to simulate 16 hours of wave propagation on a single core in the channel on the mesh made of only triangles, only quads and a mixed mesh used in the experiments above. The cell numbers are related as 200000 to 100000 to 119104. We have found that the total CPU times are related approximately as 1.68 (triangles/quads) and 1.45 (triangles/mixed). The velocity part takes most of the time step (up to 95% depending on options). The operations on cells, which include computing the velocity and elevation gradients, Coriolis and bottom friction, are twice less expensive on quadrilateral meshes. Computations of thickness transport, viscosity (harmonic, biharmonic or filters) and the momentum transport for (3) are carried out in a cycle over edges which is 1.5 times shorter for quads. The most expensive computational procedure is the biharmonic viscosity which takes about 50% of total CPU time if used. Since the quadrilateral part is less susceptible to the noise, biharmonic viscosity is not necessarily needed there, which may lead to further improvements. Note that in a code designed for purely triangular meshes computations of gradients will involve fewert multiplications than on quadrilateral meshes (3:4), yet with twice as many triangles they still require 1.5 times more operations. In summary, a factor of 1.5 in speed-up is expected, and it is obtained.

3.2 M_2 tide simulations for the North Sea

Here we present the results of simulations of the M_2 tide in the North Sea. It is a well explored domain characterized by complex geometry and high barotropic tidal activity (see, for example, Defant (1961); Davis et al. (1985); Maßmann et al. (2010)). Besides, a large amount of observations is available for this region helping to validate the model. Tidal dynamics in the North Sea is defined by two waves coming from the Strait of Dover and from the north-west of the domain. The M_2 wave plays the main role in the formation of barotropic currents. The southwards propagating wave travels as a Kelvin wave along the east coast of the United Kingdom and then turns anticlockwise continuing along the Dutch, German and Danish coasts where its loses its energy so that only a small part of this wave enters the Baltic Sea or continues to the Norwegian coast. Computations have been performed on three meshes covering the North and Baltic Seas. The first one (not shown) is composed of triangles (89859 vertices and 168117 cells)

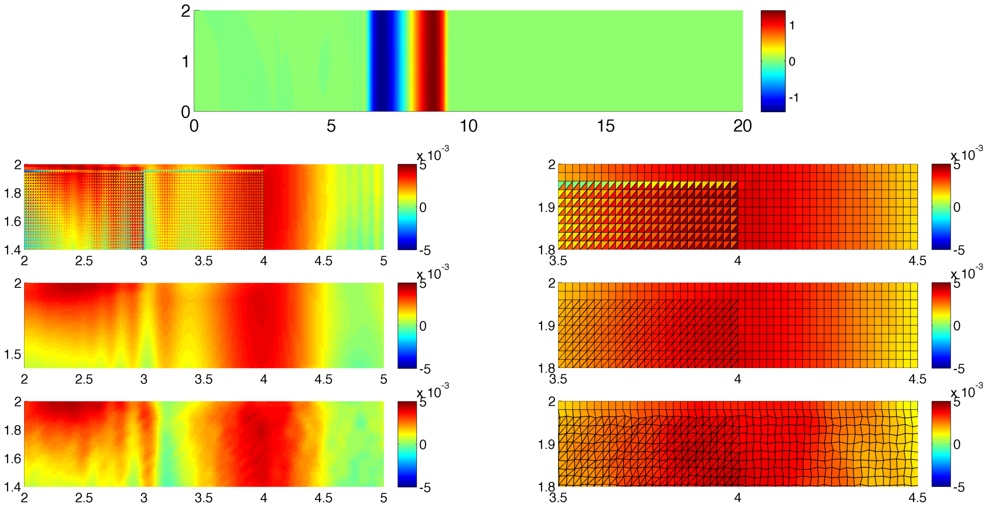

Fig. 2 The zonal velocity (m/s) of the main wave (top) and zoomed-in patterns behind it at $t = 10^4$ s. The left and right column differ in the size of subdomain shown; note the difference in colorbar (in m/s) between the top panel and the rest. The first row: no filtering and viscosity; the second row: with the biharmonic filter added; the third row: same as the second, but on a mesh with randomly shifted vertices (the mesh fragment is displayed in the right column). The meshes are triangular between 0 and 4 degrees. Noise is seen on triangles in the absence of small-scale dissipation, but small background dissipation controls it even if the cells are strongly perturbed. Since the filter effect depends on the mesh, the middle and bottom patterns slightly differ.

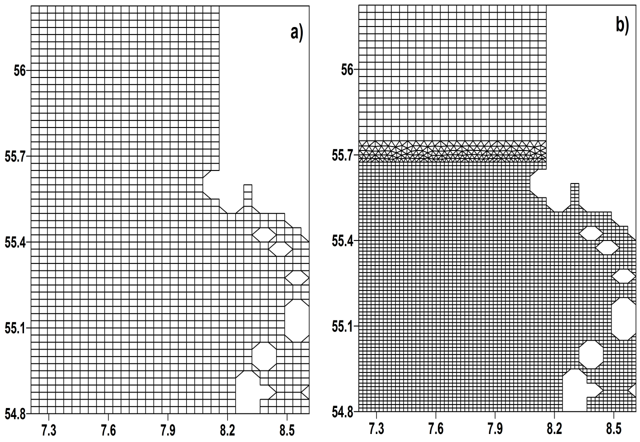

Fig. 3 Fragments of two quadrilteral meshes (the second (left) and third (right)) used in numerical simulations.

with a size varying from $\triangle_{min} \simeq 800$ m to $\triangle_{min} \simeq 5000$ m. Two other meshes are derived from a regular mesh covering the area. The second mesh is made of quads, with a small number of triangles added at the coast (153339 vertices, 2252 triangles and 148355 quads) with the cell size of 2500 m. The third mesh is a mixed one. It is obtained from the second mesh by nesting into it an area refined by a factor of two through a layer of triangles (180055 vertices, 5089 triangles and 173345 quads), as illustrated in Fig. 3. The third grid illustrates the idea of nesting. Our intention here is only to show that in all cases the algorithm works stably and accurately. The actual need in nesting of that kind is perhaps more of interest for simulations of full baroclinic dynamics.

The meshes contain two open boundaries. One is located near the shelf break at the northern end of the North Sea between Scotland and Norway and the other is in the western part of the British Channel. The model is forced by the elevation prescribed at the open boundaries. The M_2 tidal data at the northern open boundary are extracted from the TPXO6.2 model (Egbert et al. (1994)). The elevation at the open boundary in the English Channel is taken from the station located in the vicinity. The bathymetry from EMODnet morphometry dataset is employed (http://www.emodnet-hydrography.eu). The tidal map of M_2 presented in Fig. 4 was obtained on the triangular mesh, and the simulations on the quadrilateral and mixed meshes give, as expected, very close results, without any tendency to instability or noise on the mixed mesh. The amphidromic system in the North Sea is formed by the two Kelvin waves which produce three nodal lines: on the exit from the English Channel, near the northern coast of Germany, and near the southern coast of Norway. The neighborhoods of the nodal lines are characterized by a rapid phase change. We compare our model simulation results against observations at 106 stations collected and analyzed by Andersen (2008). The stations are mainly located near the coast or gathered in some area of the North Sea as shown in Fig. 4. Figure 5 shows the amplitudes and phases computed on different

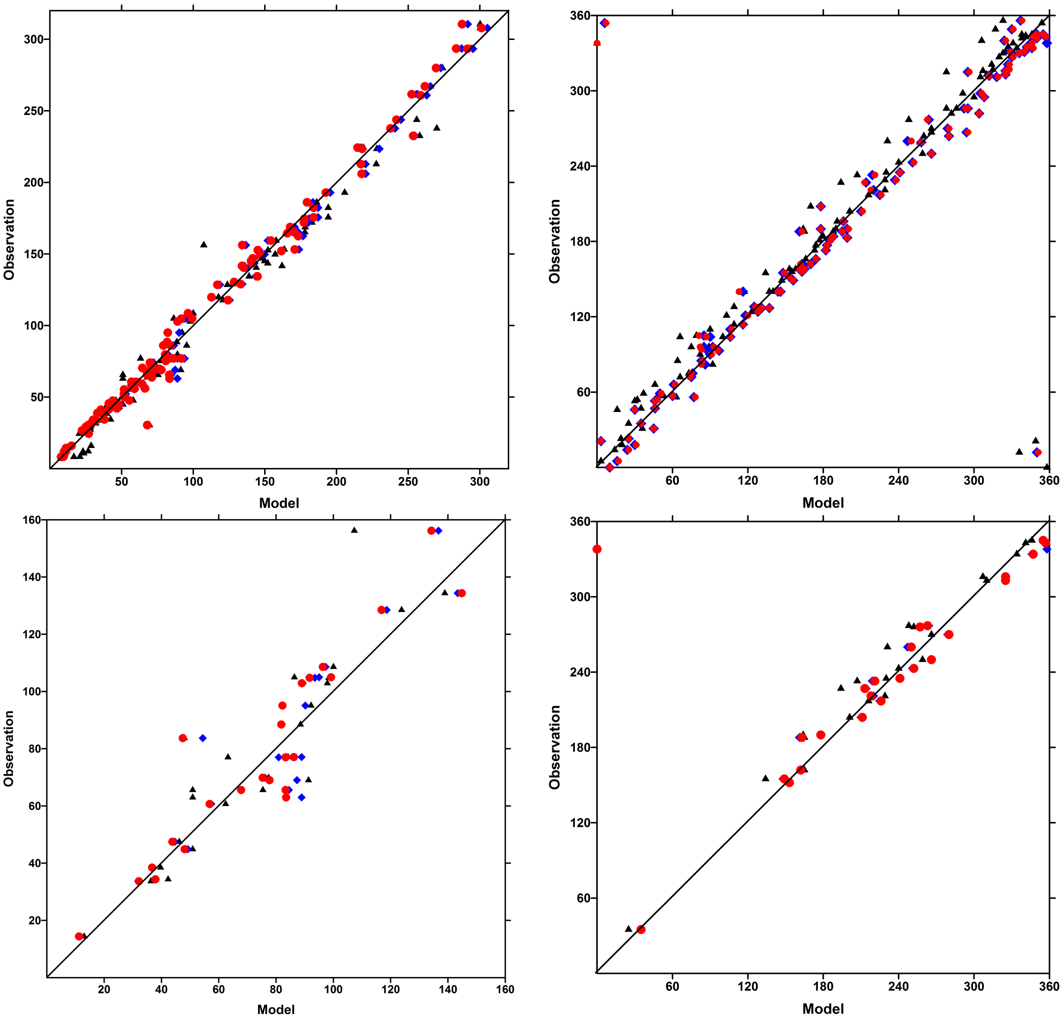

Fig. 5 Comparison of observed and simulated amplitude and phase for the entire domain (upper panels) and its fragment (dashed in Fig. 4) for the amplitude (left, cm) and and phase (right, degrees). Black triangles, blue rhombi and red circles correspond to the triangular, quadrilateral and mixed meshes respectively.

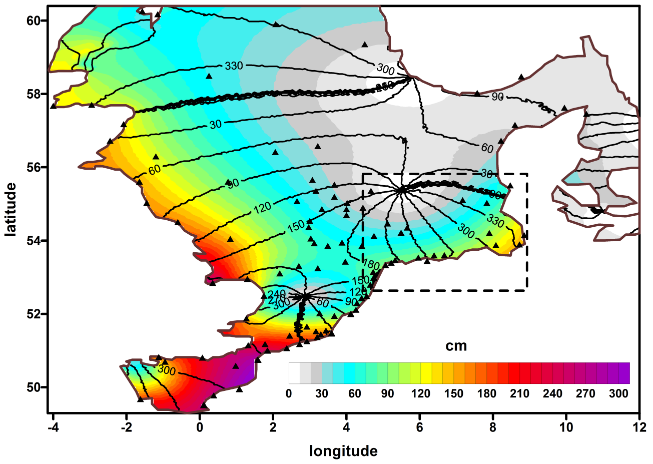

Fig. 4 Tidal map of the M_2 wave (only the North Sea part of the mesh is shown): the amplitude in cm and phase (solid line) in degrees. The triangles indicate the station locations. A dashed rectangle shows the domain where the mesh is refined; the transition zone is over its periphery.

meshes against the observed ones for the entire domain and for the part of the North Sea where the resolution is refined on the mixed mesh (see Fig. 4). Some errors seen in the results can be attributed to the uncertainty of bottom topography and the fact that the bottom friction coefficient was taken constant over the whole domain. The resolution of all three meshes is sufficient so the error is only marginally related to them. The accuracy of calculations is frequently expressed through the vector error given by the formula:

$$\mu = \frac{1}{N}\sum_{n=1}^{N}((A_* \cos\varphi_* - A\cos\varphi)^2 + (A_* \sin\varphi_* - A\sin\varphi)^2)_n^{1/2},$$

where A_*, φ_* and A, φ are the observed and computed amplitudes and phases, respectively at N stations. The total vector error on Fig. 6 does not exceed 25 cm for all 106 stations for the triangular mesh and makes approx-

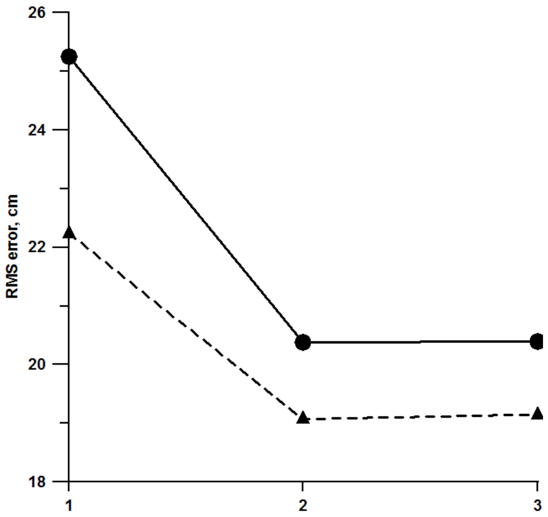

Fig. 6 RMS error on the triangular (1), quadrilateral (2) and mixed (3) meshes. The solid line corresponds to all 106 stations and the dashed one to 24 stations in the area dashed in Fig. 4.

imately 20 cm for the rectangular and mixed meshes. Similar results hold also for the stations located inside the dashed region in Fig. 4. Somewhat larger RMS error on the triangular mesh can be attributed to the need to suppress the noise as well as to the smaller number of its scalar degrees of freedom.

3.3 Barotropic tidal dynamics in the Messina Strait

The Strait of Messina separates Calabria and Sicily and links the Ionian and Tyrrhenian seas. It is characterized by a complex geometry: over a distance of about 25 km its depth varies from 1200 m to 70 m in the region of the narrow part (Fig.7). We simulate it on a quadrilateral curvilinear mesh in order to demonstrate that the approach is capable of handling such meshes too. Although such a mesh is suboptimal in our case because of its high curvature in the narrowest place of the mesh, the simulated tidal map agrees favorably with observations. Primary role in the formation of the barotropic tides is played by the M_2–wave (up to 12 cm). In the narrowest part of the Strait where the amphidrome is located, the phase of tidal velocity changes by 180^o across a 3 km distance and the elevation gradient has a minimal value of 1.7 cm/km. The tidal elevations are out-of-phase, and tidal velocities may reach 2 m/s.

The curvilinear mesh is generated using the elliptic method by Thompson (1982), with orthogonalization at the boundary (Fig.7). It contains 6171 vertices and 6000 cells with the horizontal size varying approximately between 30 m and 350 m. The elevation at the open boundaries is taken as $\zeta = \sum A\cos(2\pi t/T - \psi)$, where A, ψ,and T are the amplitude, phase, and period of the M_2 wave respectively. Computations begin from the state of rest and are continued until the periodic regime is established.

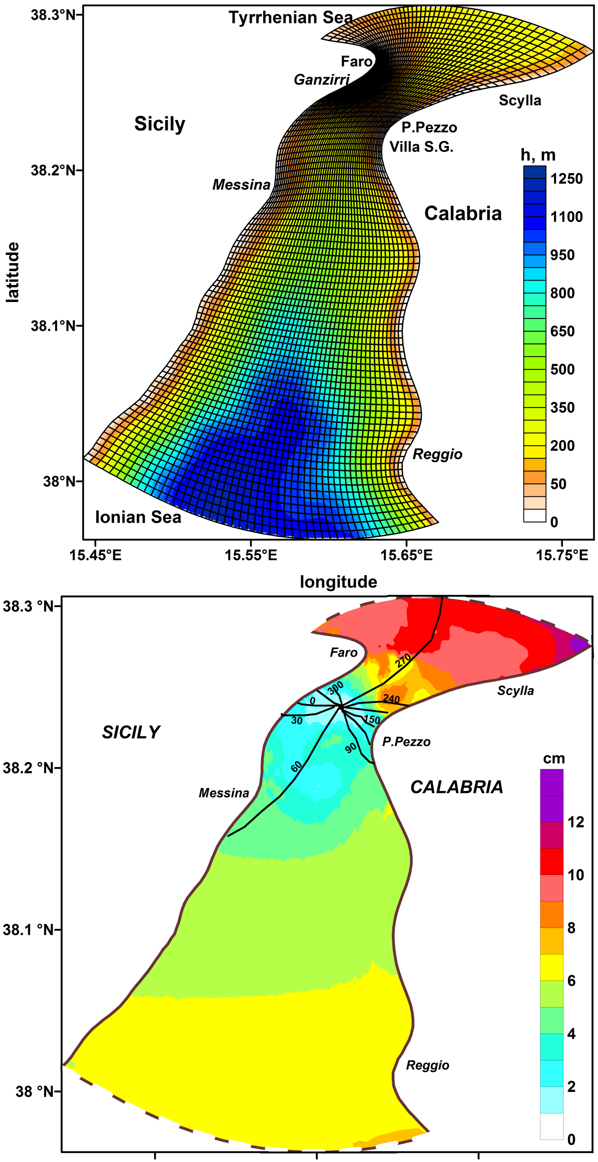

Fig. 7 Mesh with bathymetry (upper panel) and the tidal map of M_2 wave (bottom panel), amplitude in cm (color), phase in degrees (solid line).

Figure 7 presents the computed tidal map for M_2 component. Its amphidromic system with cyclonic rotation is usually interpreted in terms of Kelvin waves with opposite direction of propagation which results in an amphidrome in the middle of the channel.

The computed amplitudes and phases are compared to the available observational data in Fig. 8. The largest discrepancy is observed for the Punto Pezzo and Faro

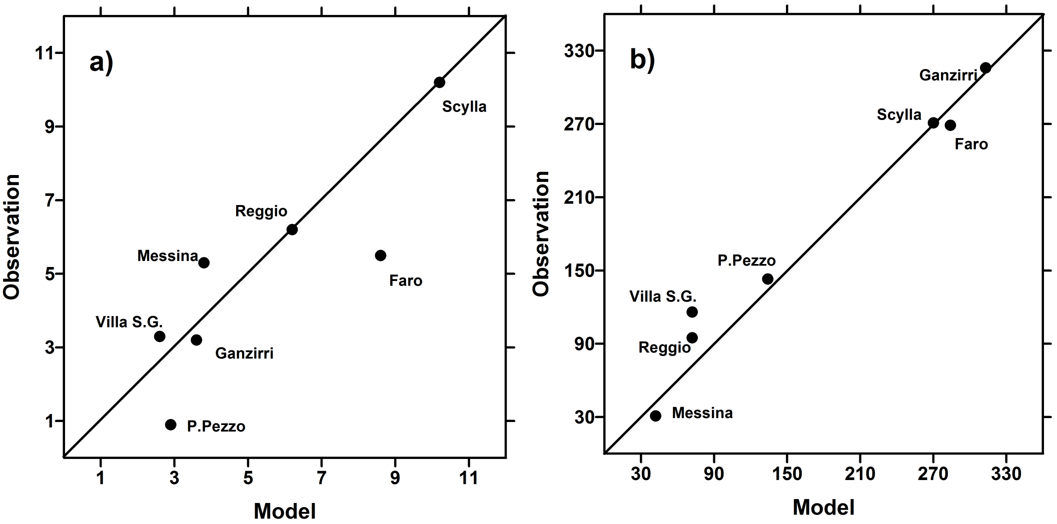

Fig. 8 Computed versus observed amplitude (left, cm) and phase (right, degrees).

points located on both sides of the amphidrome in the region. The RMS errors for all points during the tidal cycle about 1.8 cm. Very similar results were obtained by Androsov et al. (2002) with the finite-difference boundary-fitted coordinate model on the same mesh.

3.4 Wind-driven flows

Here we consider a circulation excited by a zonal wind with the stress $\boldsymbol{\tau} = (-\tau_0, 0)\cos(2\pi y/L_y)$ in a rectangular basin of approximately 30 by 30 degrees centered around a latitude of 35° N. We use the amplitude of 0.1 N/m^2, for which the circulation is non-stationary. The resolution varies from 0.07° along the boundaries (in a layer of approximately 0.5°) and in the central part of the domain (between 38.5° and 41.5° N and west of 20° E) to 0.28° in the rest of domain, as shown in Fig. 9. The transition between the coarse and fine part is through a narrow triangular layer less than one degree in width. More than 80% of the degrees of freedom are located on the fine quadrilateral mesh, so that the fact that the time step is defined by the fine mesh leads to only small loss in efficiency. The fluid layer thickness is 500 m and the viscosity varies linearly from 50 m^2/s at the western boundary to 10 m^2/s in the 2 degree zone adjacent to the western boundary, staying constant in the rest of domain. The flow is further stabilized with biharmonic viscosity. Its amplitude on the fine mesh is selected so as to keep simulations stable, and is scaled with the cell area to power 3/2 on the remaining part of the mesh. The width of the Munk boundary layer in the case of non-slip boundary condition (the distance to zero crossing in the meridional velocity) is less than 50 km, but it is still resolved on the fine mesh. The viscosity increase toward the western boundary helped to achieve this. Keeping the viscosity on the high level (50 m^2/s) everywhere in the domain suppresses the instability and results in a rather slow and smooth dynamics. In contrast, a well-developed turbulent double-gyre flow evolves in these simulations for small viscosity.

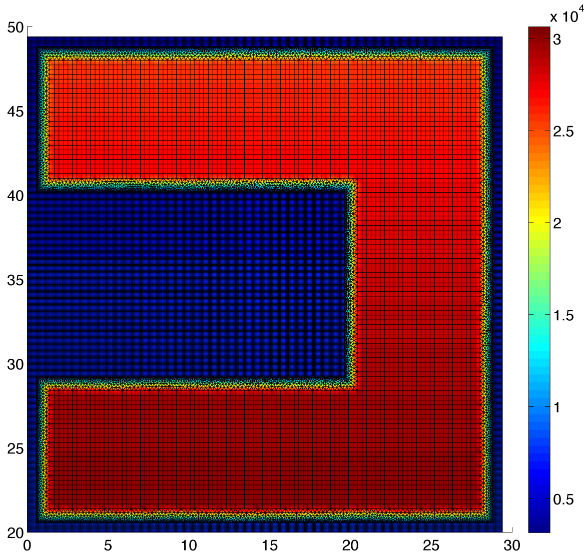

Fig. 9 Mixed mesh used for double-gyre simulations. Color corresponds to square root of the scalar volume area (in m).

Although there is no indication of noise in the fields of surface height and velocity, potential problems with the mixed meshes in flows with high grid-scale Reynolds numbers are revealed by patterns of relative vorticity. Figure 10 presents relief plots of relative vorticity in simulations run with the background biharmonic viscosity (a), and in simulations where it is increased by a factor of 3 over the transitional triangular zone (b). Although the plots correspond to the same time moment, the patterns differ because of the turbulent character of the flow. The flow is turbulent only in the fine-resolution area, and the instability is strongly damped over the coarse mesh because of the scaling of biharmonic viscosity. In panel (a) the zone of triangular mesh is visible through the small-scale (grid) noise along the periphery of the central area and also along the boundaries of the domain. Even more importantly, once generated, this noise propagates into the domain, as is well seen north of 40° N close to the western boundary in panel a). The noise is largely of geometrical origin and is due to both the lack of the order on the triangular part of the mesh (which reduces the accuracy in approximating the operators) and the change in the cell size (so that the features propagating from the fine mesh cannot be resolved any longer). This noise has to be removed. In panel b), where the biharmonic viscosity is increased three-fold (with respect to background) over the triangles, some noise is still left, but it is greatly reduced, in particular along the periphery of the central fine-resolved domain. Notice that the places where noise is still seen are associated with vorticity patches hitting the transitional zone. Noise suppression can be tuned further, and perhaps the Leith viscosity parameterization ($A_h = C|\nabla\omega|S^{3/2}$ or $A_{bh} = -C|\nabla\omega|S^{5/2}$, where ω is the relative vorticity, for harmonic and biharmonic cases respectively, with C some constant) is the most suitable candidate to handle the issue. In the same vein, it should be clear that the biharmonic viscosity coefficient or filter inverse time should be increased over some vicinity of the transitional zone on the fine mesh side to smooth features approaching the transitional part.

The implication is that fine tuning of (biharmonic) viscosity may be required over the transitional part of the mesh in a general case. It comes at no surprise, and in the case considered here the issue of noise is emphasized because the grid-scale Reynolds number is large (typical velocity is about 1m/s). It however signals that practical implementations would benefit from the availability of efficient viscosity parameterization and scaling. We also note that marginally resolved boundary layers may prove vulnerable to the inclusion of sharp transitional zones, which is the reason why the fine-resolution mesh follows the boundaries in the example here. In a companion experiment on a modified mesh where the transitional zone cuts through the western boundary layer (not shown), we were able to control the noise only for free-slip boundary conditions. Clearly the mesh must resolve the features of the flow, and large triangles cannot ensure it in the western boundary layer. Once again, although it is foreseeable, it means additional tuning in practice.

4 Conclusions

We have described the implementation of the cell-vertex discretization of shallow-water equations that works on mixed meshes composed of triangles and quads. The key element ensuring seamless performance of cell-vertex discretization on mixed meshes is dissipation which suppresses small scale noise that would otherwise develop on the triangular (transitional) part of these meshes either because of too large velocity space or because of sharp change in resolution. We have shown how to implement biharmonic viscosity and biharmonic filter which are sufficient in many cases to maintain stability, and can be augmented to the Smagorinsky or Leith type of parameterizations.

Although transitional zones between triangles and quads cause no problems in the test cases described here, the stability depends on the presence of a tuned (background) biharmonic filter or viscosity in a general case. Additionally, sharp transitional zones should be avoided in places where the flow is not resolved. We have not encountered manifestations of pressure modes on the quadrilateral part of the mesh. They may still be an issue in 3D on z-coordinate meshes where they will be triggered by steps in bottom topography. The shallow-water framework is by far insufficient to explore these and other issues of relevance for large-scale flows, and the analysis here needs to be augmented in future. Nevertheless, the consideration here hints at certain interesting possibilities for ocean modeling.

Indeed, codes formulated on unstructured meshes are as a rule slower per degree of freedom than their structured-mesh counterparts. Quads are more economical in this respect as they introduce fewer edges and cells, and lead to shorter cycles. Moreover, there is a hypothetical possibility to combine structured (on the refined patch) and unstructured representations, as in most cases in large-scale applications one is willing to use setups where the refined part is responsible for the large fraction of degrees of freedom while occupying a compact domain. The functionality of mixed meshes comes almost at no price and the approach may still be used on purely triangular meshes.

It is also clear that the discretization discussed here is not the only possibility, and that C-grids could perhaps be a better choice if they would demonstrate a stable performance in geostrophically dominated flows, which remains to be seen.

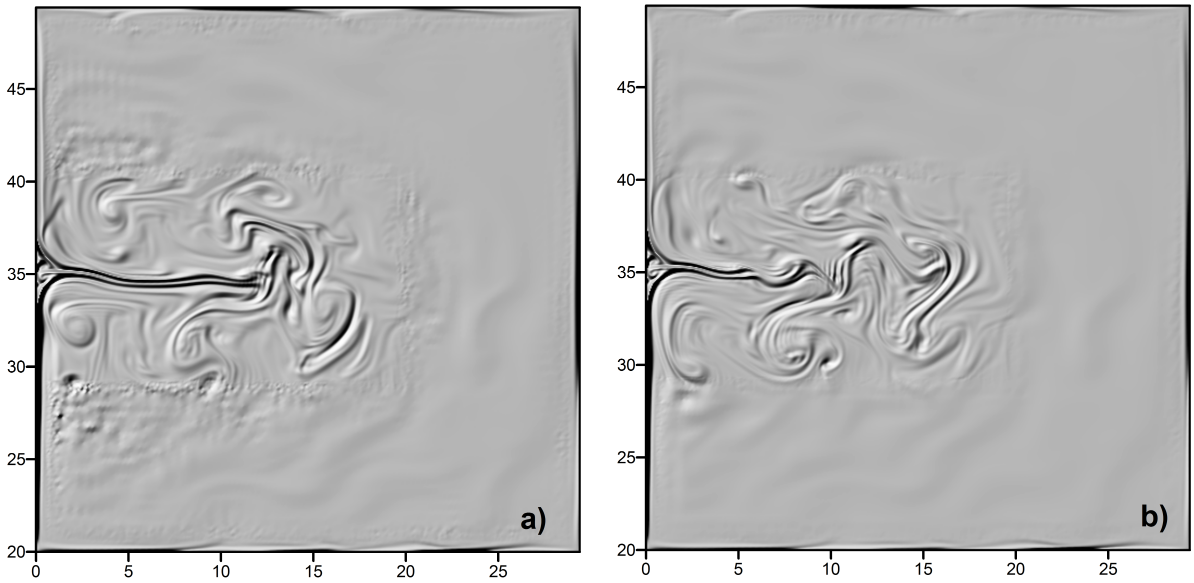

Fig. 10 Snaphots of relative vorticity (relief) in double-gyre flow in a turbulent regime. In b) the biharmonic viscosity is increased by a factor 3 on triangles compared to a), which proved sufficient to suppress most of noise in the transitional zone.

An interesting aspect, which deserves a separate study, is the size of transitional zones. Too narrow zones may require additional dissipation for their stability, so broader zones can be more robust. The use of refinement for eddying flows is another issue of interest. Because eddies interact with the mean flow, locally resolving them may have a global impact. We think, however, that these and similar questions should be addressed with more realistic 3D setups.

5 Appendix

Here we briefly consider the energy and potential vorticity (PV) conservation by the cell-vertex discretization. We are interested only in the spatial part (sometimes term semi-conservation is used to reflect it). The conservation of both is easily achieved because the discretization is mimetic. In contrast, enstrophy conservation cannot be easily maintained because of the averaging of relative vorticity, as detailed below.

A key element in making the code energy and PV conserving lies in writing the momentum equation in the vector-invariant form

$$\partial_t \mathbf{u} + (f+\omega)\mathbf{k}\times\mathbf{u} + \nabla B = \mathbf{F} + (D_d + D_v)\mathbf{u}, \qquad (9)$$

where $\omega = \mathbf{k}\cdot\mathrm{curl}\mathbf{u}$ is the relative vorticity, and $B = \mathbf{u}^2/2 + g\eta$ is the analog of the Bernoulli function. For the discretization in hand the relative vorticity is defined at the scalar points (vertices) and is computed similar to the divergence, but with the outer normals replaced by the tangent vectors

$$\int_v \omega dS = \sum_{e=e(v)} [(H\mathbf{u}\cdot\mathbf{t}l)_l + (H\mathbf{u}\cdot\mathbf{t}l)_r]_e,$$

where the tangent vectors are directed so that they make a left circle around vertex v.

For convenience, we introduce the matrix notation for the operators of divergence, gradient and $\mathbf{k}\cdot$ curl as D_{vc}, $\mathbf{G}_{cv}$ and C_{vc} respectively. Here the indices v and c denote vertices and cells respectively. These operators, as defined above, satisfy the following identities. First, the divergence and gradient are adjoint in the sense

$$(S\mathbf{a})_c\cdot\mathbf{G}_{cv}b_v = -(Sb)_v D_{vc}\mathbf{a}_c,$$

where summation is implied over the repeating indices, $\mathbf{a}$ is a vector field defined at cell centers and b is a scalar field defined at vertices, and S is the area of the respective control volume. Second,

$$C_{vc}\mathbf{G}_{cj}b_j = 0,$$

which corresponds to $\nabla\times\nabla b = 0$ in the continuous case. Finally, a trivial consequence of curl and divergence being both defined at vertices is

$$C_{vc}(\mathbf{k}\times\mathbf{a}_c) = D_{vc}\mathbf{a}_c.$$

These identities are what makes the cell-vertex discretization mimetic. Many discretizations are not able to maintain them. For example, non-staggered vertex-vertex discretization fails to maintain the orthogonality of curl and gradient operators.

To prove energy conservation (we omit the dissipative terms here — they correspond to the sink of kinetic energy) one first sums the discretized (9) multiplied with $S_c H_c \mathbf{u}_c$ and the discretized (2) multiplied with $S_v(\mathbf{u}^2/2)_v$ (note that the kinetic energy is estimated at vertices because (2) is written at vertices) to obtain the kinetic energy budget

$$(SH\mathbf{u})_c \cdot \partial \mathbf{u}_c + (S\mathbf{u}^2/2)_v \partial_t H_v = -(SH\mathbf{u})_c \cdot \mathbf{G}_{cv}(\mathbf{u}^2/2)_v - (S\mathbf{u}^2/2)_v D_{vc}(\mathbf{u}H)_c - g(SH\mathbf{u})_c \cdot \mathbf{G}_{ci}\eta_v. \quad (10)$$

In this equation, the two terms on the left hand side make the time derivative of the total kinetic energy. Note that they involve summations over the vector (first) and scalar (second) control volumes. For the global budget the fact that they come from different spaces is immaterial, yet characterizing the kinetic energy budget on the local level would require to specify what is H_c and what is $\mathbf{u}_v^2$. A consistent way of doing it is to introduce $(SH)_c = S_{cv} H_v$, where S_{cv} is the area of cell c associated to vertex v (a part of cell cut by the median-dual control volume), and likewise $(\mathbf{u}^2 S)_v = \mathbf{u}_c^2 S_{cv}$. Here, as above, summation is implied over the repeating indices. This allows one, by rearrangement, to interpreted the terms in (10) as related to cells (or to S_{cv}).

The first two terms on the right hand side of (10) correspond to the kinetic energy redistribution and disappear by virtue of the adjoint character of gradient and divergence operators. The remaining term describes the transfer of kinetic energy from the potential energy. Its counterpart is the exchange term in the potential energy budget which is obtained by multiplying the discretized (2) with $gH_v S_v$ and summing over v

$$gH_v S_v \partial_t \eta_v = -gH_v S_v D_{vc}(\mathbf{u}H)_c. \quad (11)$$

The adjoint character of the gradient and divergence operators warrants here that the sum of exchange terms (kinetic plus potential) is identically zero. This proves that energy conservation is maintained by the spatial discretization considered here.

If the curl operator C_{vc} is applied to the discretized (9), the result is the PV balance

$$\partial_t(qH)_v + D_{vc}(qH\mathbf{u})_c = 0,$$

where we have introduced the PV q by $(qH) = \omega + f$. Here we used the mimetic properties of the curl operator listed above. It will be observed for any estimate of q at cell centers. However, since H_c is already defined, one can formally write $q_c = (\omega_c + f_c)/H_c$. There is certain freedom in selecting ω_c then. On C-grids similar freedom can be used to warrant enstrophy conservation. Here the need to specify the cell value instead of the edge one precludes achieving that. However, the cell values can be selected in an upwind biased way, which should lead to enstrophy dissipation, if needed.

References

Andersen OB (2008). Personal communication

Androsov AA, Kagan BA, Romanenkov DA, Voltzinger NE (2002) Numerical modelling of barotropic tidal dynamics in the strait of Messina. Adv in Water Res 25:401–415

Chen C, Liu H, Beardsley RC (2003) An unstructured grid, finite-volume, three-dimensional, primitive equations ocean model: Applications to coastal ocean and estuaries. J Atmos Ocean Tech 20:159–186

Danilov S (2012) Two finite-volume unstructured mesh models for large-scale ocean modeling. Ocean Modell 47:14–25

Davies AM, Sauvel J, Evans J (1985) Computing near coastal tidal dynamics from observations and a numerical model. Continental Shelf Research 4(3):341–366

Defant A (1961) Physical oceanography, vol II. Pergamon Press, New York.

Egbert G, Bennett A, Foreman M (1994) TOPEX/Poseidon tides estimated using a global inverse model. J of Geoph Res 99(C12):24821–24852

Kernkamp HWJ, Van Dam A, Stelling GS, de Coede ED (2011) Efficient scheme for the shallow water equations on unstructured grids with application to the continental shelf. Ocean Dyn 61:1175–1188

Le Roux DY, Rostand V, Pouliot B (2007) Analysis of numerically induced oscillations in 2D finite-element shallow-water models. Part I: Inertia-gravity waves. SIAM J Sci Comput 29:331–360

Le Roux DY (2012) Spurious inertial oscillations in shallow-water models. J Comput Phys 231:7959-7987

Maßmann S, Androsov A, Danilov S (2010) Intercomparison between finite element and finite volume approaches to model North Sea tides. Cont Shelf Res 30:680-691

Ringler TD, Randall DA (2002) The ZM grid: an alternative to the Z grid. Mon Wea Rev 130:1411–1422

Shchepetkin AF, McWilliams JC (2005) The Regional Oceanic Modeling System (ROMS): a split-explicit, free-surface, topography-following-coordinate oceanic model. Ocean Modell 9:347-404

Stelling GS, Duinmeijer SPA (2003) A staggered conservative scheme for every Froude number in rapidly varied shallow water flows. Internat J for Numer Methods in Fluids 43(12):1329–1354

Szmelter J, Smolarkiewicz P (2010) An edge-based unstructured mesh discretization in geospherical framework. J Comput Phys 229:4980-4995

Thompson JF (Ed) (1982) Numerical Grid Generation. Appl Math and Comput 10–11: p.910

Thuburn J, Ringler TD, Skamarock WC, Klemp JB (2009) Numerical representation of geostrophic modes on arbitrarily structured C-grids. J Comput Phys 228:8321–8335

Zhang Y, Baptista AM (2008) SELFE: A semi-implicit Eulerian-Lagrangian finite-element model for cross-scale ocean circulation. Ocean Modelling 21:71–96

Westerink JJ, Luettich RA, Blain CA, Scheffner NW (1992) ADCIRC: An Advanced Three-Dimensional Circulation Model for Shelves, Coasts and Estuaries; Report 2: Users Manual for ADCIRC-2DDI. Contractors Report to the US Army Corps of Engineers. Washington DC