Semi-implicit, semi-Lagrangian modelling for environmental problems on staggered Cartesian grids with cut cells

G. Rosatti [a,*], D. Cesari [b], L. Bonaventura [a,c]

[a] Dipartimento di Ingegneria Civile ed Ambientale, Mesiano di Povo, 77, I-38050 (TN), Italy
[b] ARPA Emilia Romagna – SIM, Viale Silvani 6, 40122, Bologna, Italy
[c] Max Planck Institut für Meteorologie, Bundesstr. 53 20146 Hamburg, Germany

Abstract

In this paper, we propose an extension of semi-implicit models for the shallow water or Euler equations on staggered Cartesian meshes to handle the presence of cut cells at the boundaries of the computational domain. Semi-Lagrangian advection algorithms are also extended to staggered Cartesian meshes with cut boundary cells. A linear reconstruction algorithm and radial basis function interpolation are used, respectively, to compute approximate trajectories and to recover an accurate velocity field in cells intersected by the boundary. The effective accuracy of radial basis function interpolators for semi-Lagrangian algorithms is assessed by simple numerical experiments. Timestep control techniques in the computation of the semi-Lagrangian trajectories are also discussed. The accuracy of the resulting models for environmental flows is demonstrated by a number of test cases simulating nonlinear open channel flows and baroclinic flows over an isolated obstacle.
© 2004 Elsevier Inc. All rights reserved.

1. Introduction

The use of Cartesian grids composed of rectangular cells does not allow for optimal representation of complex geometries. Numerical methods employing either boundary fitted coordinate systems or unstructured grids (finite volumes, finite elements) are a well established way to overcome this problems. In some environmental applications, however, these approaches may introduce other difficulties. Boundary fitted coordinate systems lead to equations of motion with metric terms that become singular over steep

---

orography (see e.g. the difficulties reported in [30,33]). This can result in systematic errors or loss of efficiency for semi-implicit time discretizations, due to the fact that the condition number of the resulting matrices depends on the orography steepness. Furthermore, boundary fitted coordinates are not easily applicable in very high resolution simulations, e.g. for air or water quality modelling purposes, where even vertical obstacles (buildings, piers) need to be considered. The use of unstructured grids for the vertical discretization yields an accurate description of the boundary, but, whenever gravity plays a major role, it implies that the vertical discretization grid is not aligned with the direction of the main force acting on the flow. As a consequence, more complex discretization approaches are required and many important related physical processes (e.g., moist convection or suspended sediment transport) cannot be represented in a straightforward way. Finally, for long term simulations of river and coastal morphodynamics, in which complex domains evolve in time, the use of boundary fitted coordinates or unstructured grids could result in possibly severe loss of efficiency due to extensive remeshing.

Although these difficulties do not prevent the application of boundary fitted approaches, they motivate the investigation of discretization techniques which allow for an improved description of the boundaries while retaining the accuracy, efficiency and simplicity of the standard discretizations on Cartesian grid in absence of boundaries. These techniques can be loosely described as *cut cell* or *shaved cell* or *partial step* approaches. Various attempts in this direction are documented in the literature, see e.g. [1,4,7,9–13,24,22,25,37]. The cut cell approaches are nothing but more or less extensive applications of the finite volume discretization technique. In particular, the *partial step* approach (according to the classification of [1]) relies essentially on coupling the finite volume discretization of the divergence operator with standard centered finite difference approaches for the pressure gradients. It can be used to develop very efficient semi-implicit, semi-Lagrangian time discretizations (see e.g. [4,10]) for which the matrices to be inverted in the semi-implicit step have very simple structure and very good conditioning properties.

In this paper, we propose an extension of semi-implicit, semi-Lagrangian models for the shallow water or Euler equations on staggered Cartesian meshes to handle the presence of cut cells at the boundaries of the computational domain. In particular, we will show that an appropriate semi-Lagrangian discretization coupled to the *partial step* approach allows to achieve sufficient accuracy for some environmental applications, even at coarse resolutions, both for the vertical and the horizontal discretizations. The proposed extension of semi-Lagrangian algorithms relies essentially on interpolation algorithms for the velocity field in the boundary cells, which allow to improve significantly the accuracy of the computed fields with respect to those obtained with simple Cartesian grids without cut cells. It will also be shown that the results obtained compare well with those produced by models employing boundary fitted coordinates. Specific applications targeted for the methods developed here are river hydraulics and high resolution mesoscale modelling.

In Section 2, the proposed extension of semi-implicit models is motivated and described. In Section 3, the basic features of the semi-Lagrangian method will be reviewed. In Section 4, trajectory approximation techniques based on Euler or Runge–Kutta methods with substepping will be described. It will also be shown how to choose the timestep for the trajectory computations so as to avoid that the trajectories cross the boundary lines in either full or cut cells. In Section 5, a linear interpolation technique for the velocity field will be discussed, which allows to impose the free-slip boundary condition exactly along the boundaries of cut cells. In Section 6, results of numerical tests will be presented, in which the proposed algorithm was employed in the computation of the backward trajectories starting from a given stationary velocity field. In Section 7, interpolators based on radial basis functions will be briefly described, along with an evaluation of their effective accuracy in cases relevant for the targeted applications. It will be shown that these interpolation algorithms can be advantageous even for standard Cartesian grids. In Section 8, results obtained with full two-dimensional semi-Lagrangian models including nonlinear advection will be presented, which show how the combination of the linear reconstruction described in Section 5 and the radial basis function interpolators allows to achieve rather accurate results on a number of test cases. Some conclusions and plans for applications to three-dimensional models will be presented in Section 9.

2. Semi-implicit modelling of environmental flows on staggered Cartesian grids with cut cells

The two-dimensional shallow water equations will be used as a model problem to introduce the key features of the proposed discretization approach. In particular, consider

$$
\begin{aligned}
&\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} = -g\frac{\partial \eta}{\partial x} - \gamma U, \\
&\frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} = -g\frac{\partial \eta}{\partial y} - \gamma V, \\
&\frac{\partial \eta}{\partial t} + \frac{\partial (UH)}{\partial x} + \frac{\partial (VH)}{\partial y} = 0.
\end{aligned}
\tag{1}
$$

Here, η is the free surface elevation above a constant reference level, H is the total fluid depth, U, V are the components of the vertically averaged velocity vector in the x, y directions, respectively, g is the gravitational acceleration and γ is the bottom friction coefficient.

Our starting point will be a finite difference discretization on a Cartesian grid with C-type staggering of the discrete variables, as commonly employed in MAC-type discretizations [20] of two dimensional fluid flow (see Fig. 1). The discrete values of the velocity component U are defined at the half integer nodes $(i+\frac{1}{2}, j)$ and those of the velocity component V are defined at the half integer nodes $(i, j+\frac{1}{2})$, while discrete scalar variables like η are defined at integer nodes (i,j). A finite difference, semi-implicit discretization of Eqs. (1) with this approach yields

$$
U_{i+\frac{1}{2},j}^{n+1} = FU_{i+\frac{1}{2},j}^{n} - g\frac{\Delta t}{\Delta x}\left[\alpha(\eta_{i+1,j}^{n+1} - \eta_{i,j}^{n+1}) + (1-\alpha)(\eta_{i+1,j}^{n} - \eta_{i,j}^{n})\right] - \Delta t\gamma_{i+\frac{1}{2},j}^{n}U_{i+\frac{1}{2},j}^{n+1},
\tag{2}
$$

$$
V_{i,j+\frac{1}{2}}^{n+1} = FV_{i,j+\frac{1}{2}}^{n} - g\frac{\Delta t}{\Delta y}\left[\alpha(\eta_{i,j+1}^{n+1} - \eta_{i,j}^{n+1}) + (1-\alpha)(\eta_{i,j+1}^{n} - \eta_{i,j}^{n})\right] - \Delta t\gamma_{i,j+\frac{1}{2}}^{n}V_{i,j+\frac{1}{2}}^{n+1},
\tag{3}
$$

$$
\begin{aligned}
\eta_{i,j}^{n+1} = \eta_{i,j}^{n} &- \frac{\Delta t}{\Delta x}\alpha(U_{i+\frac{1}{2},j}^{n+1}H_{i+\frac{1}{2},j}^{n} - U_{i-\frac{1}{2},j}^{n+1}H_{i-\frac{1}{2},j}^{n}) - \frac{\Delta t}{\Delta y}\alpha(V_{i,j+\frac{1}{2}}^{n+1}H_{i,j+\frac{1}{2}}^{n} - V_{i,j-\frac{1}{2}}^{n+1}H_{i,j-\frac{1}{2}}^{n}) \\
&- \frac{\Delta t}{\Delta x}(1-\alpha)(U_{i+\frac{1}{2},j}^{n}H_{i+\frac{1}{2},j}^{n} - U_{i-\frac{1}{2},j}^{n}H_{i-\frac{1}{2},j}^{n}) - \frac{\Delta t}{\Delta y}(1-\alpha)(V_{i,j+\frac{1}{2}}^{n}H_{i,j+\frac{1}{2}}^{n} - V_{i,j-\frac{1}{2}}^{n}H_{i,j-\frac{1}{2}}^{n}).
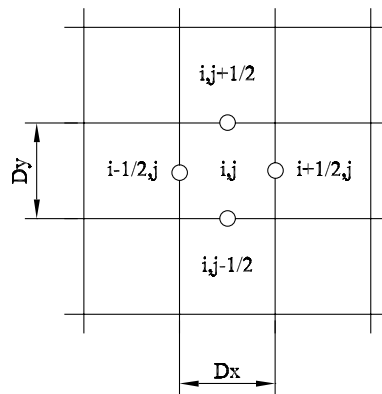\end{aligned}
\tag{4}
$$

Fig. 1. Variable arrangement on a staggered C-grid cell.

Here, $FU^n_{i+\frac{1}{2},j}$, $FV^n_{i,j+\frac{1}{2}}$ represent the discretization of the nonlinear momentum advection terms. In the models we are concerned with, this will be performed by a semi-Lagrangian method (see the more detailed description and the references in Section 3). α is the averaging parameter used in the Crank–Nicolson time-stepping approach. Linear stability analysis shows that for $\alpha \geqslant \frac{1}{2}$ the method is unconditionally stable. Furthermore, substitution of Eqs. (2) and (3) into Eq. (4) yields an algebraic problem to be solved at each timestep whose matrix is symmetric, positive definite and diagonally dominant. This discretization was introduced in [8] and has been used successfully in a number of practical applications, see e.g. [9,11,19,39]. However, due to the use of rectangular cells, results can be quite inaccurate when the boundaries are not parallel to the coordinate directions. For demonstration purposes, we will consider the application of this method to simulation of open channel flow in a straight channel that is not aligned with the coordinate directions. In this case, a steady state analytic solution exists, which allows to compute the discretization errors. The details of the test configuration are described in Section 8.1. The spatial resolution is such that the channel width corresponds to approximately 10 grid cells. The relative errors obtained with discretization (2)–(4) are shown in the first line of Table 1. It can be observed that results are quite poor, as expected, in spite of the fact that a high order interpolation method (see Section 7) was employed in the semi-Lagrangian scheme. Since the steady state solution is also the solution of the equations obtained from (1) by omitting the nonlinear advection terms, the same test was repeated by computing the solution to this simplified problem. The results are reported in the second line of Table 1. It can be observed that, while results for the velocity field still display large errors, the error on the free surface elevation has decreased by one order of magnitude. This identifies the discretization of the advection terms as one of the main sources of error when using regular Cartesian grids to approximate irregular boundaries.

We consider now the introduction of an improved description of the boundaries in the context of a Cartesian grid approach. We assume that the domain boundaries are described by segments of straight lines and that each boundary cell is intersected by at most a single straight line (see Fig. 2). This a priori limitation rules out too complicated subgrid behavior of the boundaries, but it is sufficiently general to allow for appropriate boundary description in most environmental flows.

Let $\delta_{i+\frac{1}{2},j}$, $\delta_{i,j+\frac{1}{2}}$ denote the fractions of the cut cell sides within the computational domain in the x, y directions, respectively (see Fig. 3). It should be remarked that velocity component locations along the cut cell edges are now moved to the midpoint of the part of the edge that belongs to the computational domain.

Denote now the area of the cut cell (i,j) by $A_{i,j}$ and let $\Theta_{i,j} = \frac{A_{i,j}}{\Delta x \Delta y}$. The analog of the previous discretization of the continuity equation is now

$$\Theta_{i,j}\eta^{n+1}_{i,j} = \Theta_{i,j}\eta^n_{i,j} - \frac{\Delta t}{\Delta x}\alpha(U^{n+1}_{i+\frac{1}{2},j}H^n_{i+\frac{1}{2},j}\delta_{i+\frac{1}{2},j} - U^{n+1}_{i-\frac{1}{2},j}H^n_{i-\frac{1}{2},j}\delta_{i-\frac{1}{2},j}) - \frac{\Delta t}{\Delta y}\alpha(V^{n+1}_{i,j+\frac{1}{2}}H^n_{i,j+\frac{1}{2}}\delta_{i,j+\frac{1}{2}}$$

$$- V^{n+1}_{i,j-\frac{1}{2}}H^n_{i,j-\frac{1}{2}}\delta_{i,j-\frac{1}{2}}) - \frac{\Delta t}{\Delta x}(1-\alpha)(U^n_{i+\frac{1}{2},j}H^n_{i+\frac{1}{2},j}\delta_{i+\frac{1}{2},j} - U^n_{i-\frac{1}{2},j}H^n_{i-\frac{1}{2},j}\delta_{i-\frac{1}{2},j})$$

$$- \frac{\Delta t}{\Delta y}(1-\alpha)(V^n_{i,j+\frac{1}{2}}H^n_{i,j+\frac{1}{2}}\delta_{i,j+\frac{1}{2}} - V^n_{i,j-\frac{1}{2}}H^n_{i,j-\frac{1}{2}}\delta_{i,j-\frac{1}{2}}) \qquad (5)$$

In principle, the discretization of the momentum equations should also be modified to take into account the cut cells. In the approach we propose, however, no modification is introduced in the discretization of the free surface gradients in the momentum equations, along the lines of the approach used in the *partial*

Table 1
l_∞ relative errors with respect to analytical solution for steady state straight channel flow for various model configurations

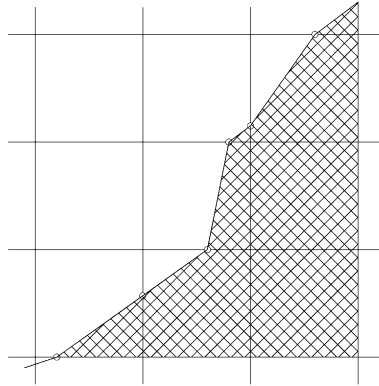|	Water depth	U	V
Square cells	0.1	0.56	0.73
Square cells, no advection	1.45×10^{-2}	0.4	0.65
Cut cells, no advection	3.39×10^{-5}	2.2×10^{-3}	2.02×10^{-2}

Fig. 2. Cut cells produced by piecewise linear approximation of the domain boundary.
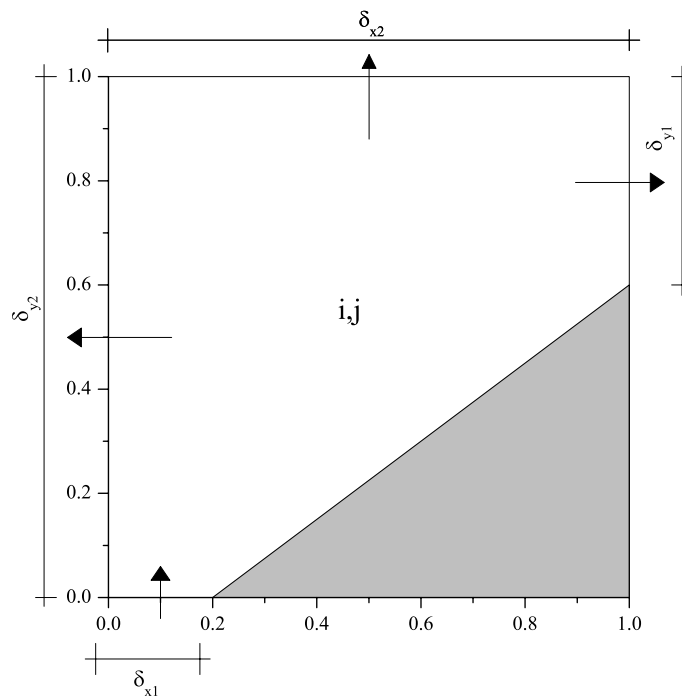

Fig. 3. Definition of the cut cell side fractions.

step discretizations (see e.g. [4,10,11]). This results in a loss of second order accuracy at the boundaries for the approximation of the free surface gradients. On the other hand, this avoids the introduction of skew symmetric terms in the algebraic problem resulting from the semi-implicit discretization, in contrast to approaches employing more accurate discretizations of the gradient operator at the boundary (see e.g. [24]). Our point is that this loss of accuracy does not make the method useless for environmental applications. In order to give evidence for this, the previous channel flow test was repeated, using now the discretization (5) and omitting again the nonlinear momentum advection terms. The results are shown in the third line of Table 1. It can be seen that now the solution is computed with a discretization error of about 2% for

the velocity field and much smaller for the free surface elevation. This level of accuracy is sufficient for many realistic applications. For example, typical tidal gauges only allow to measure the free surface fluctuations to within approximately 1%. Thus, provided that the same accuracy achieved in this test can be obtained also in the discretization of the complete equation set (1), the proposed approach could be a useful tool for environmental applications. For this reason, an accurate extension of the semi-Lagrangian method to Cartesian grids with cut cells will be considered in the next section.

3. The semi-Lagrangian method

The semi-Lagrangian method (SL) has been widely used in environmental applications (see e.g. [36]), since it allows to achieve very accurate discretization of the nonlinear advection terms. SL is especially efficient when applied to structured Cartesian grids. However, in applications to domains with complex geometry, great care has to be taken when implementing the two key steps of SL methods – i.e., approximation of the Lagrangian paths and interpolation at the departure point of the Lagrangian trajectory – to achieve an effectively accurate discretization close to the boundary. Typical results of inaccurate implementations are characteristic lines that cross domain boundaries as well as excessive momentum loss and spurious vorticity production at the boundaries.

The semi-Lagrangian approximation of the advection equation

$$\frac{\mathrm{d}c}{\mathrm{d}t} = \frac{\partial c}{\partial t} + \vec{v} \cdot \nabla c = 0 \tag{6}$$

for a quantity $c(\vec{x}, t)$ transported by the velocity field $\vec{v}(\vec{x}, t)$ is based on the numerical approximation of the Lagrangian particle paths, that can be identified as solutions of the ordinary differential equations

$$\frac{\mathrm{d}\vec{x}}{\mathrm{d}t} = \vec{v}(\vec{x}(t), t). \tag{7}$$

More specifically, consider the case of a two-dimensional problem and denote by $c_{i,j}^{n+1}$ the numerical solution of Eq. (6) at time $t^{n+1} = \Delta t(n + 1)$, where Δt is the constant timestep of the time discretization and nodes (i,j) are the cell centers of a staggered, C-type finite difference grid, with spacings Δx, Δy, in the x, y directions, respectively. The semi-Lagrangian discretization approach aims at determining an approximate value $\vec{x}^*$ for the starting point at time t^n of the Lagrangian trajectory reaching $(i\Delta x, j\Delta y)$ at time t^{n+1}. For this purpose, Eq. (7) is solved backward in time with initial datum $\vec{x}(t^{n+1}) = (i\Delta x, j\Delta y)$ and up to time t^n, thus determining a value for $\vec{x}^*$. For each gridpoint (i,j), the new value of c is then defined as

$$c_{i,j}^{n+1} = c(\vec{x}^*, t^n), \tag{8}$$

where an appropriate interpolation method is used to reconstruct the value of the solution at $\vec{x}^*$, which generally does not belong to the discretization grid. It is to be remarked that, in most implementations of the semi-Lagrangian methods, also the computation of the approximated backward trajectory requires an interpolation of the velocity field in some of the intermediate steps necessary for the solution of Eq. (7).

The choice of the most appropriate interpolation methods depends on the problem under consideration. Cubic Lagrange or spline interpolation have been used in many applications on Cartesian rectangular grids (see e.g. [36]). In the context of finite element discretizations, the finite element interpolation operators are generally used, see e.g. [27,29,31]. More recently, other interpolation techniques such as kriging have also been applied successfully in [26]. In this paper, a similar approach will be pursued, using radial basis functions reconstruction algorithms. These techniques have already been used successfully in the context of semi-Lagrangian methods (see e.g. [3]) and have been applied to the functional reconstruction step in accurate conservative schemes on unstructured grids (see e.g. [35]). It will be shown in Section 7 that this type of interpolation allows to provide sufficient accuracy also close to cut cell boundaries.

For the determination of $\vec{x}^*$ by solution of Eq. (7) backwards in time, implicit methods are usually employed in most operational implementations of the semi-Lagrangian method (see e.g. [36]). The convergence criterion of these methods requires in general a restriction on the so-called Lipschitz number (see e.g. [34]). However, other approaches based on the approximation of the trajectories by Runge–Kutta methods with timesteps shorter than the global timestep Δt have also been applied successfully (see e.g. [4,8,17,18]), especially in problems with complex geometries. For these approaches, a local Courant–Friedrichs–Lewy criterion must be satisfied, in order to avoid that the approximated trajectories cross each other or intersect the boundaries. These techniques will be reviewed in Section 4 along with their extension to computational domains with cut cells.

4. Substepping methods for the approximation of trajectories

Substepping methods for the approximation of the backward trajectory in a semi-Lagrangian method consist simply in the discretization of Eq. (7) by standard ODE solvers with a timestep $\Delta\tau$ that is a fraction of the timestep Δt. More specifically, let $\Delta\tau = \frac{\Delta t}{N}$, where N is to be conveniently chosen later, let $s^k = t^{n+1} - k\Delta\tau$ and let $\vec{x}^{(N)} = (i\Delta x, j\Delta y)$ denote the gridpoint reached at time t^{n+1} by the trajectory to be approximated. Euler or Runge–Kutta methods can be used to go back along the trajectories. For second order accuracy one has for $k = 0, \ldots, N-1, N$

$$
\begin{aligned}
\vec{x}^{(k-1),*} &= \vec{x}^{(k)} - \Delta\tau \vec{v}(\vec{x}^{(k)}, s^k), \\
\vec{x}^{(k-1)} &= \vec{x}^{(k)} - \frac{\Delta\tau}{2}\left[\vec{v}(\vec{x}^{(k-1),*}, s^k - \Delta\tau) + \vec{v}(\vec{x}^{(k)}, s^k) \right].
\end{aligned}
\tag{9}
$$

The desired approximation of $\vec{x}^*$ is obtained by setting $\vec{x}^* = \vec{x}^{(0)}$. In practical applications of the semi-Lagrangian method, the velocity field is also the output of a numerical method (in the case of nonlinear momentum advection, exactly the same method is used). Furthermore, the flow field is assumed to be sufficiently smooth. Therefore, velocities at intermediate time values such as $\vec{v}(\vec{x}, s^k)$ can be replaced by consistent extrapolations from the values at previous timesteps without degrading the formal accuracy of the method. Taking $s \in [t^n, t^{n+1}]$, the appropriate extrapolation formula for the second order scheme is given by

$$
\vec{v}(\vec{x}, s) = \left(1 + \frac{s - t^n}{\Delta t} \right) \vec{v}(\vec{x}, t^n) - \left(\frac{s - t^n}{\Delta t} \right) \vec{v}(\vec{x}, t^{n-1}).
\tag{10}
$$

In many applications the velocity field can be also taken to be constant in time over the interval $[t^n, t^{n+1}]$ and equal to its extrapolated value at time $t + \frac{\Delta t}{2}$, without degrading the effective accuracy of the method. Furthermore, as remarked previously, for each substep k an interpolation must be performed in order to compute the velocity at $\vec{x}^{(k)}$ and at the other intermediate points. At this stage, bilinear interpolation is usually recognized to be sufficiently accurate (see e.g. [36]) and piecewise linear interpolation has also been used with good results in [4,8]. Therefore, each of the N substeps can be performed at a low computational cost.

In absence of boundaries, the substeps $\Delta\tau$ are chosen so that the departure point is tracked back crossing at most one cell for each substep, in order to avoid different trajectories crossing each other (see e.g. [5,8]). If linear interpolation is used, it was shown in [8] that an appropriate upper bound is given by

$$
\Delta\tau \leqslant \min_{(i,j)} \left[\frac{\Delta x}{|u_{i+\frac{1}{2},j}|}, \frac{\Delta y}{|v_{i,j+\frac{1}{2}}|} \right].
\tag{11}
$$

Since in general only a few cells in the whole computational domain display large Courant numbers, it is convenient to compute local estimates for $\Delta\tau$, as suggested e.g. in [5]:

$$\Delta\tau_{i,j} \leqslant \min_{(h,l)\in I_{i,j}} \left[\frac{\Delta x}{|u_{h+\frac{1}{2},l}|}, \frac{\Delta y}{|v_{h,l+\frac{1}{2}}|} \right], \tag{12}$$

where

$$I_{i,j} = \{(h,l) : i-2 \leqslant h \leqslant i+1, j-2 \leqslant l \leqslant j+1\}.$$

In the case of simple boundaries, these conditions are also sufficient to ensure that trajectories do not leave the computational domain. However, this does not hold for an arbitrary shape of the boundary. Consider for example the configuration in Fig. 4. Assuming that the discretization step has unit length, that the velocity components have their maximum value $u_{i+\frac{1}{2},j} = u_{i-\frac{1}{2},j} = v_{i,j+\frac{1}{2}} = v_{i,j-\frac{1}{2}} = 1$ in the corner cell (i,j) and that the final point for the trajectory is in the center of the cell, it is easy to see that for $\sqrt{2}/2 < \Delta\tau < 1$ condition (11) is satisfied, but the departure point is beyond the domain boundary.

In the case of cut cells, the problem is even more complex and the upper bound for $\Delta\tau$ has to take into account the irregular shape of the boundary cell. In principle, the optimal value of $\Delta\tau$ can be obtained by the following steps:

(1) given a point P with coordinates (x_0,y_0) along the computed trajectory, evaluate the velocity vector;
(2) a point (X,Y) along an approximated backward trajectory is obtained using this constant value of the velocity vector:

$$X = x_0 - t \, u(x_0,y_0),$$

$$Y = y_0 - t \, v(x_0,y_0),$$

it is to be remarked that (X,Y) may represent an intermediate value of the solution, as required for example in Runge–Kutta methods
(3) the intersection (x_c,y_c) of the approximated trajectory with the straight line which represents the cut cell boundary is computed
(4) the collision time $t_c(x_0,y_0)$ necessary to reach the boundary in a single step is obtained dividing the distance L between the starting point (x_0,y_0) and the crossing point (x_c,y_c) by the norm of the velocity vector:
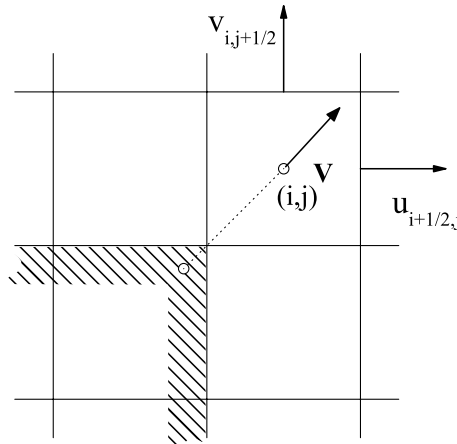

Fig. 4. Configuration with semi-Lagrangian trajectory leaving the computational domain.

$$t_c(x_0, y_0) = \frac{L}{|V|} = \frac{\sqrt{(x_0 - x_c(x_0, y_0))^2 + (y_0 - y_c(x_0, y_0))^2}}{\sqrt{u^2(x_0, y_0) + v^2(x_0, y_0)}} \tag{13}$$

(5) the upper bound for $\Delta\tau$ is taken to be the minimum value of $t_c(x_0, y_0) \; \forall \; x_0, y_0 \in$ the cut cell domain.

The function (13) has been computed for various specific cell configurations and velocity fields satisfying the inviscid free-slip boundary condition, using the reconstruction algorithm described in Section 5. Empirical evidence shows that $t_c(x_0, y_0)$ is generally almost constant throughout the domain of a cut cell. Thus, there is some confidence that a reasonable upper bound for $\Delta\tau$ can be achieved. In practical implementations, such an upper bound is derived as follows. Let $\delta^x_{i+\frac{1}{2}, j}, \delta^y_{i, j+\frac{1}{2}}$ denote the cut cell sides fractions that are within the computational domain (see Fig. 3), as defined in Section 2. The estimate is then given by

$$\Delta\tau_{i,j} \leqslant \phi \min \left[\frac{\min\limits_{(h,l) \in I_{i,j}} \delta^x_{h+\frac{1}{2}, l}}{\max\limits_{(h,l) \in I_{i,j}} |u_{h+\frac{1}{2}, l}|}, \frac{\min\limits_{(h,l) \in I_{i,j}} \delta^y_{h, l+\frac{1}{2}}}{\max\limits_{(h,l) \in I_{i,j}} |v_{h, l+\frac{1}{2}}|} \right], \tag{14}$$

where $u_{i+\frac{1}{2}, j}, \; v_{i, j+\frac{1}{2}}$ are the value of the velocity components defined on the sides of the cell and ϕ is a security factor, that was generally taken to be equal to $\frac{1}{2}$. It should be remarked that, in this way, only heuristically justified conditions are obtained, which ensure that the trajectory does not cross the cut cell solid boundary, as long as the particle is still within the same cell at the end of the substep. If the position at the end of the substep is in a neighbouring cell, these conditions are not in principle sufficient to guarantee that the solid boundary within that neighbouring cell has not been crossed. Generally, however, since imposing the proper boundary condition along the whole boundary line yields velocity fields whose normal components close to the boundary are effectively small (see e.g. Figs. 5–7 in Section 5), criterion (14) was found to be sufficient and no trajectory was observed to leave the computational domain in any of the tests performed.

5. Reconstruction of the velocity field within a cut cell

A linear reconstruction algorithm will now be introduced, which allows to recover a vector field from the staggered velocity components in cut boundary cells. The purpose of this algorithm will be mainly to allow for reconstruction of the velocity field in the computation of the Lagrangian trajectories. As explained in Section 1, only cut cells intersected by a single straight line will be considered, along with a C-type variable staggering. A local coordinate system $\mathbf{x} = (x, y)$ is introduced, whose origin is at the lower left corner of the Cartesian rectangular cell. Denote by $\mathbf{x}_r = (x_r, y_r)$ the intersection of the straight boundary line with the y axis. The boundary line is then given by $y = -\frac{n_1}{n_2}(x - x_r) + y_r$, where $\mathbf{n} = (n_1, n_2)$ is the normal direction to the boundary. For vertical boundary lines the shape of the cell is the same as for full cells, so this special case will not be considered further. Since discrete velocity components are defined at different grid locations, some interpolation or reconstruction algorithm must be applied if a two-dimensional vector field $\mathbf{U} = \mathbf{U}(\mathbf{x})$ is to be defined. A generic linear reconstruction of the velocity field is given by

$$\mathbf{U}(\mathbf{x}) = \hat{\mathbf{U}} + \mathbf{A}\mathbf{x}, \tag{15}$$

where

$$\mathbf{A} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \hat{\mathbf{U}} = \begin{bmatrix} \hat{u} \\ \hat{v} \end{bmatrix}$$
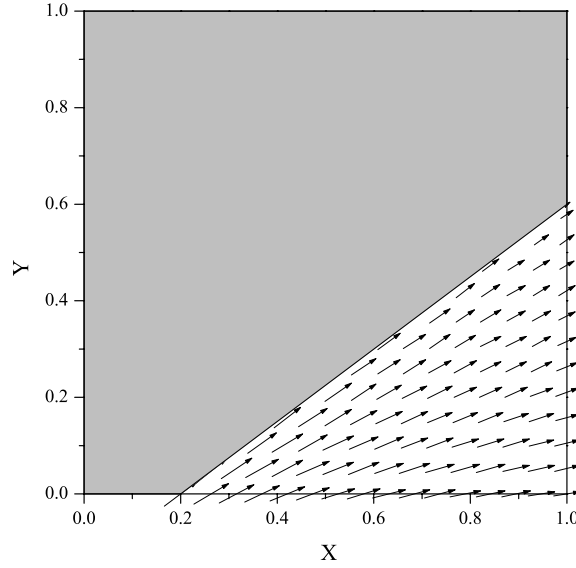
Fig. 5. Reconstructed velocity field in a triangular cut cell with side length L, $u(1,0.3)/L = 0.8$ s^{-1}, $v(0.6,0)/L = 1.5$ s^{-1}.
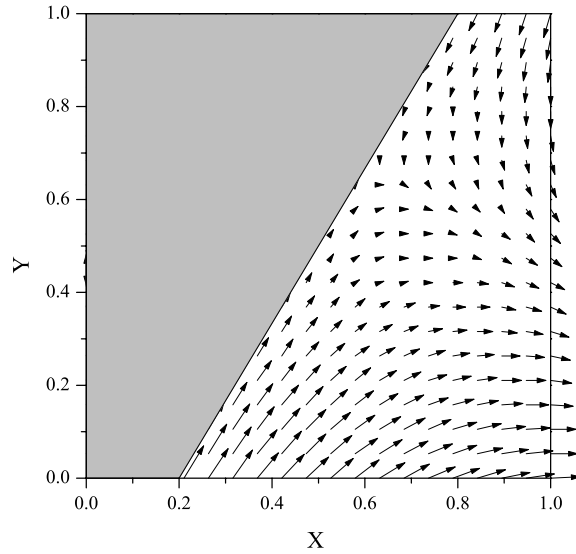

Fig. 6. Reconstructed velocity field in a trapezoidal cut cell of side L and $u(1,0.5)/L = 1$ s^{-1}, $v(0.9,1)/L = -1.7$ s^{-1}, $v(0.6,0)/L = 1$ s^{-1}.

and $\hat{u}$, $\hat{v}$ are constants that represent the value of $\mathbf{U}(\mathbf{x})$ at the origin of the coordinate system. The six coefficients $\hat{u}$, a, b, $\hat{v}$, c, d are to be determined by imposing appropriate conditions. At each point x_0, y_0 for which a velocity component is known one has e.g.

$$u_0 = \hat{u} + ax_0 + by_0.$$

Furthermore, appropriate boundary conditions have to be prescribed along the boundary line. In this work, only free-slip boundary conditions will be considered. This is not entirely general and reconstructions not relying on this assumption could also be investigated. It is possible that, when surface stresses are pre-
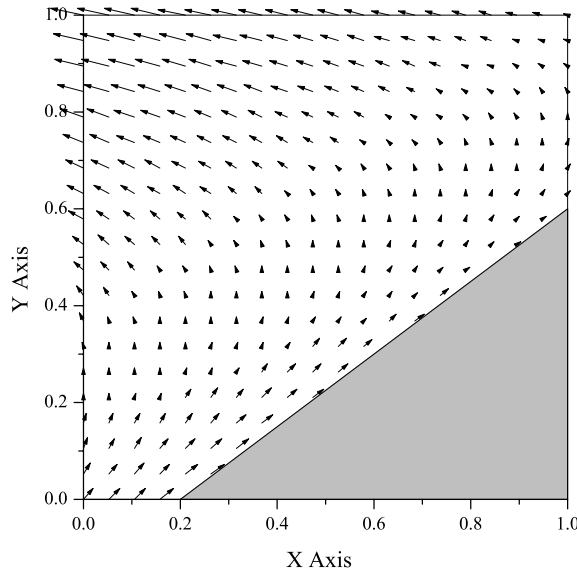
Fig. 7. Reconstructed velocity field in a pentagonal cut cell with side L and $u(0,0.5)/L = -1$ s^{-1}, $u(1,0.8)/L = -0.1$ s^{-1}, $v(0.5,1)/L = 0.1$ s^{-1}, $v(0.1,0)/L = 1$ s^{-1}.

sent, a different boundary condition would be necessary. On the other hand, in all the applications of the semi-Lagrangian approach to the Euler or shallow water equations with turbulent viscosity terms, the fluid parcel trajectories are computed by approximate solution of Eq. (7) (see e.g. [36]), which strictly speaking only holds in the case of inviscid advection. Thus, if the algorithm is applied in the context of the trajectory approximation, this appears to be consistent with a general implicit assumption underlying the semi-Lagrangian method. Requiring the free-slip condition

$$\mathbf{U} \cdot \mathbf{n} = 0$$

to be satisfied for the vector field given by Eq. (15) along the whole line segment of equation $y = -\frac{n_1}{n_2}(x - x_r) + y_r$, results into the conditions

$$x\left[an_1 - b\frac{n_1^2}{n_2} + cn_2 - dn_1\right] + \hat{u}n_1 + bn_1\left[\frac{n_1}{n_2}x_r + y_r\right] + \hat{v}n_2 + dn_2\left[\frac{n_1}{n_2}x_r + y_r\right] = 0. \tag{16}$$

This will be true for all the x values within the cell if

$$an_1 - b\frac{n_1^2}{n_2} + cn_2 - dn_1 = 0, \tag{17}$$

$$\hat{u}n_1 + bn_1\left[\frac{n_1}{n_2}x_r + y_r\right] + \hat{v}n_2 + d[n_1x_r + n_2y_r] = 0. \tag{18}$$

Other conditions on the coefficients $\hat{u}, a, b, \hat{v}, c, d$ can be derived assuming that the flow within the cell is irrotational,

$$\frac{\partial u}{\partial y} - \frac{\partial v}{\partial x} = 0,$$

which yields either the condition $b - c = 0$ or the conditions $b = c = 0$. Since this assumption only concerns the subgrid reconstruction of the velocity field, it does not imply any restriction for the applicability of the

method to rotational flows. Imposing these conditions allows to determine uniquely a velocity field reconstruction in each of the possible cases of cut cell shape:

- *Triangular cut cells:* two normal velocity components are known, two conditions result from imposing zero normal velocity along the boundary and two conditions result from imposing $b = c = 0$.
- *Trapezoidal cut cells:* three normal velocity components are known, two conditions result from imposing zero normal velocity along the boundary and one condition results from imposing $b - c = 0$
- *Pentagonal cut cells:* four normal velocity components are known, two conditions result from imposing zero normal velocity along the boundary.

The resulting linear system in the six unknowns $\hat{u}$, $\hat{v}$, a, b, c, d is solved by the Gauss method with pivoting. It is to be remarked that also the customary bilinear interpolation in square cells away from the boundary can be derived along the same lines, assuming the four normal velocity components to be known and requiring the coefficients b,c to be equal to zero. This allows to avoid spurious vorticity production and yields automatically the correct value for the discrete divergence.

Examples of reconstructed velocity fields on these three types of cut cells are shown in Figs. 5–7, respectively. It can be seen how the free-slip boundary condition is effectively imposed along the whole boundary.

In order to assess the accuracy of the resulting method, the velocity field for potential flow around a cylinder of radius R_0 was considered, so that

$$
\begin{aligned}
U &= U_0 + \frac{U_0 R_0^2}{2R^3} - \frac{3}{2}\frac{R_0^3 U_0(x - x_0)^2}{R^5}, \\
V &= -\frac{3}{2}\frac{R_0^3 U_0(x - x_0)(y - y_0)}{R^5}
\end{aligned}
\tag{19}
$$

for the Cartesian velocity components. Here, $R^2 = (x - x_0)^2 + (y - y_0)^2$ and (x_0,y_0) are the coordinates of the cylinder center. This velocity field was reconstructed on the Cartesian grid with cut cells by the approach outlined in this section. The values $U_0 = 10$ m s^{-1} and $R_0 = 3$ m were chosen. The computational domain was taken to be a square with sides 40 m long. The l_2 norm errors at various resolutions are shown in Table 2 for the approximation of the U velocity component. The error behavior in the l_∞ norm is much worse and is not reported here. Actually, some of the errors are related to the piecewise linear representation of the curved boundary and to the large gradients displayed by the specific field (19) close to the boundaries. Furthermore this can be considered as a worst case scenario, since the velocity field considered has a highly localized maximum at the downstream boundary of the cylinder. A milder test can be performed by considering a simple rotation around the cylinder, whose corresponding vector field is given by

$$
U = -(y - y_0) \quad V = (x - x_0).
\tag{20}
$$

Results for this case yield l_2 norm errors approximately one order of magnitude smaller, with similar estimated convergence rates. As it will be shown in Section 8, this poor convergence behavior does not have a significant impact on the quality of the simulation results, especially if this linear reconstruction technique

Table 2
Accuracy of the linear reconstruction for the velocity field on a Cartesian grid with cut cells: potential flow

Δx	l_2 error	Estimate of convergence rate in l_2
1	4.0×10^{-4}	–
0.5	1.1×10^{-4}	1.8
0.25	7.9×10^{-5}	0.5
0.125	2.5×10^{-5}	1.7

is used only in the computation of the approximated trajectories. In this case, the use of the linear recon-struction procedure is actually very convenient to make sure that approximated trajectories are not crossing the domain boundaries.

6. Numerical tests on the accuracy of the trajectory approximation

In order to analyze the impact of the proposed reconstruction algorithm on the accuracy of the trajectory approximation, as a first test a stationary velocity field was prescribed and the approximated semi-Lagrangian trajectories were compared to the field streamlines.

The velocity field was given by assuming potential flow around a cylinder as in (19). The semi-Lagrangian trajectories were computed numerically by backward integration in time, as in the semi-Lagrangian approaches described in [5,8]. The total integration time T and the asymptotic horizontal velocity U_0 were chosen so that the ratio $T U_0/R_0$, which can be interpreted as a Courant number based on the dimension of the obstacle, is equal to $\frac{1}{2}$. Avoiding that the particles jump over the obstacle within one semi-Lagrangian time step appears to be an appropriate requirement for an accurate representation of the flow (see e.g. the discussion in [2]). Varying the spatial resolution Δx, different values of $TU_0/\Delta x$ are obtained, which corre-spond to the Courant number in a semi-Lagrangian timestep of length T.

In each substep, an interpolation is needed to reconstruct the complete velocity vector from its pre-scribed normal components. This was done using the algorithm of Section 5. The exact streamlines were computed solving the system of ODE (7) with velocity components given by Eqs. (19) without any inter-polation and using a Runge–Kutta–Verner sixth-order method with a timestep approximately equal to $10^{-6}T$. A number of trajectories crossing various cut boundary cells were computed and i, l_∞^i and l_2^i errors were evaluated along each trajectory i. Some of the exact and computed trajectories are shown in Fig. 8, for the upstream and downstream part of the boundary, respectively.
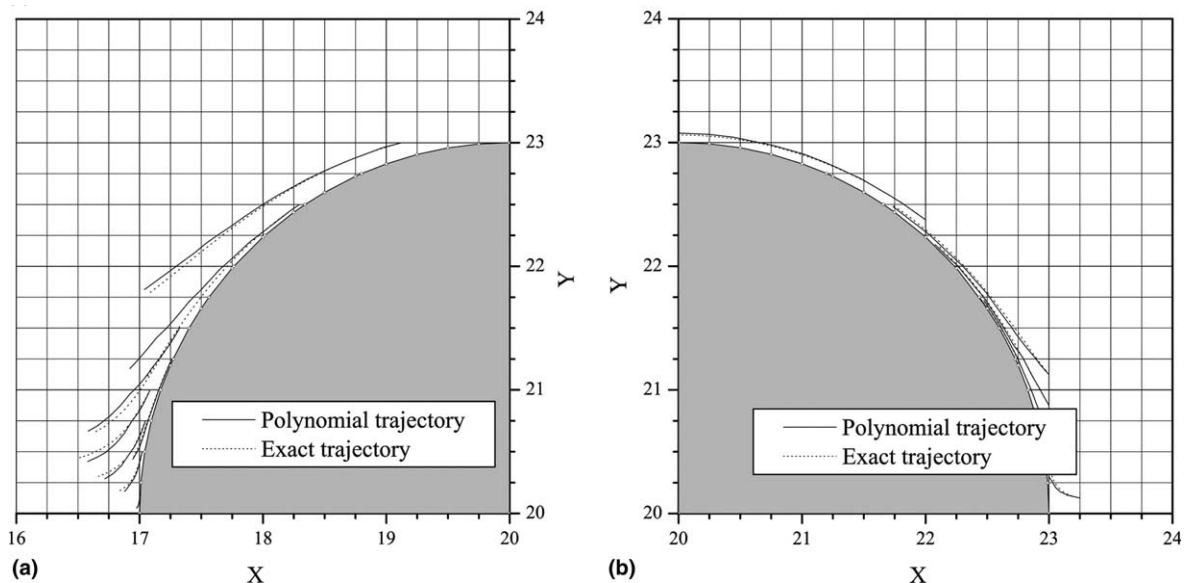

Fig. 8. Exact and computed backward trajectories in potential flow around a cylinder. Trajectory departure points are placed either upstream (a) or downstream (b) of the obstacle.

The maximum value and the mean value over the ensemble of l^i_∞ and l^i_2 were then considered. These error estimates, normalized by the value of R_0, are shown in Fig. 9 for various values of the ratio $R_0/\Delta x$, which is approximately half the number of cells used to represent the obstacle. As a comparison, the values of the error indicators are reported for the same test with the cylinder approximated by full square cells. The values in the test with full cells are denoted by full symbols and have been computed only for the largest ratio $R_0/\Delta x$. It can be observed that cut cells yield on average a reasonable approximation even when the obstacle is only marginally resolved, which is extremely important in many environmental applications to domains with complex topographic and orographic features.

7. Interpolation algorithms based on radial basis functions

For applications to problems with complex geometries, the stencils required by standard high order Lagrange or spline interpolations include in general a number of points beyond the domain boundaries. To overcome this problem, radial basis function (RBF) interpolators will be used in this paper. In this section it will be shown that RBF interpolators allow to achieve higher accuracy throughout the computational domain on a Cartesian grid with cut cells. RBF are an interpolation technique for scattered multivariate data that has been studied in depth over the last two decades (see e.g. the review in [6]). When appropriate radial basis functions are chosen (for example the multiquadric function $\phi(t) = \sqrt{1 + (t\epsilon)^2}$, where ϵ represents the reciprocal of a spatial scale), spectral convergence rates can be achieved for RBF interpolation (see e.g. [28,32]). In some cases, it can be shown that also the interpolated function derivatives can be approximated just as accurately by the derivative of the RBF interpolator. RBF techniques have been applied e.g.
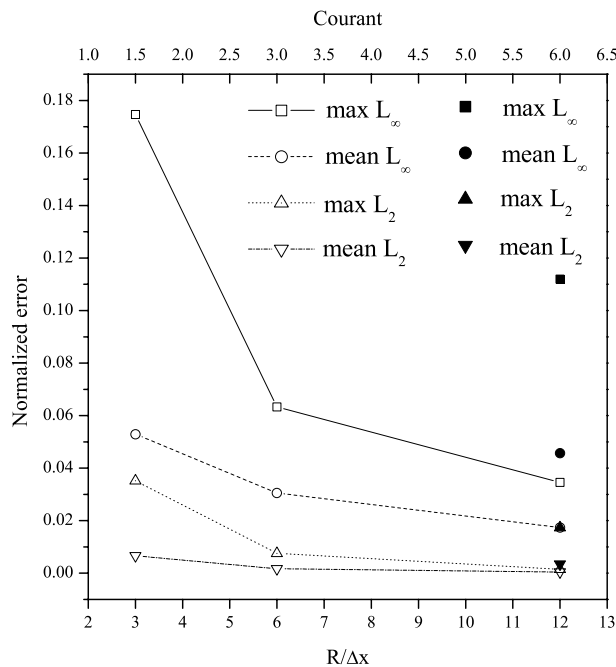

Fig. 9. Errors in the computation of the trajectories in potential flow around a cylinder. Empty symbols refer to the values obtained with cut cells.

in [35] to high order reconstructions for accurate finite volume schemes and in [3] to semi-Lagrangian advection on adaptive unstructured grids. Similar interpolation approaches based on the kriging technique have also been applied successfully in [26].

Although the purpose of this paper is not to discuss RBF interpolation as such, their effective accuracy and robustness for the reconstruction of scalar and vector fields on Cartesian grids with cut cells will be demonstrated in this section by simple examples. We believe that these apparently trivial tests are necessary, since the theoretical framework in which the convergence proofs are usually carried out (no boundaries, infinite stencils) does not provide sufficient confidence that the technique is applicable under more realistic conditions. In [23], a convergence theorem for local (i.e., with finite size stencil) RBF interpolation with thin plate splines is given. However, this does not extend to multiquadrics interpolation, so that an empirical estimation of the convergence rates for these interpolation procedures is appropriate. Furthermore, the role of the spatial scale was not investigated in any of the previously quoted papers.

First of all it should be remarked that, even for standard Cartesian grids, RBF interpolators can provide accurate, fully multidimensional interpolation algorithms whose computational cost is not much greater than that of more standard techniques, such as e.g. tensor product Lagrange interpolation. As an example, the asymptotic behavior of the l_2 and l_∞ errors in the RBF interpolation of the function $f(x,y) = \sin(2k\pi x) \cdot \sin(2k\pi y)$ on the unit square is shown in Table 3 for the case $k = 4$. The estimated convergence rates have been computed by the formula

$$\text{rate} = -\log_2 \frac{\text{error}(\Delta x)}{\text{error}(2\Delta x)},$$

where the error is computed either in the l_2 or l_∞ norm.

For this test, a stencil of $N = 10$ points was used and the RBF interpolator was constrained to fit linear polynomials in x,y exactly (see e.g. [6]). It can be noticed how fourth order l_∞ estimated convergence rate is reached using a stencil that is in fact smaller than that of tensor product Lagrange interpolation. The corresponding results for the approximation of $\frac{\partial f}{\partial x}$ are shown in Table 4.

Table 3
Accuracy of RBF interpolators on a Cartesian grid

Δx	l_∞ error	l_2 error	Estimate of convergence rate in l_∞	Estimate of convergence rate in l_2
5×10^{-2}	7.0×10^{-2}	1.8×10^{-3}	–	–
2.5×10^{-2}	5.0×10^{-3}	6.5×10^{-5}	3.8	4.8
1.25×10^{-2}	3.3×10^{-4}	2.1×10^{-6}	3.9	4.9
6.25×10^{-3}	2.1×10^{-5}	6.6×10^{-8}	4.0	5.0
3.125×10^{-3}	1.3×10^{-6}	2.1×10^{-9}	4.0	5.0

Table 4
Accuracy of RBF interpolators for x-derivative reconstruction on a Cartesian grid. Results for the y direction are similar

Δx	l_∞ error	l_2 error	Estimate of convergence rate in l_∞	Estimate of convergence rate in l_2
5×10^{-2}	9.7×10^{-2}	2.6×10^{-3}	–	–
2.5×10^{-2}	1.3×10^{-2}	1.7×10^{-4}	2.9	3.9
1.25×10^{-2}	1.7×10^{-3}	1.1×10^{-5}	2.9	3.9
6.25×10^{-3}	2.1×10^{-4}	6.8×10^{-7}	3.0	4.0
3.125×10^{-3}	2.6×10^{-5}	4.1×10^{-8}	3.0	4.1

Furthermore, ϵ was chosen so that the corresponding spatial scale was 1/4 of the domain size. It is to be remarked that the quality of the results was found to be sensitive to the appropriate choice of this scale. Essentially, when higher wave numbers were considered, better results could be obtained with smaller values of the spatial scale (i.e., larger values of ϵ). In view of the application to standard hydrodynamical models without multiscale features, some robust general rule must be found, which ensures sufficiently accurate results independently of the specific features of the flow. In the tests we have performed, good results were obtained when the scale was determined by the geometry of the domain, as a fraction of the domain size or of the typical size of the boundary features. For example, it could be observed that with a fixed scale corresponding to 1/4 of the domain size, third order accuracy was maintained up to wavenumber $k = 32$.

Other tests concerned the convergence behavior on grids with cut cells. The potential flow velocity field given by (19) was again reconstructed on the Cartesian grid with cut cells. For this test, the values $U_0 = 10$ m s^{-1} and $R_0 = 3$ m were chosen. The computational domain was taken to be a square with sides 40 m long. The behavior of the l_2 and l_∞ errors is shown in Table 5 for the approximation of the U velocity component.

For this test, a stencil of $N = 10$ points was used, the RBF interpolator was constrained to fit linear polynomials in x,y exactly and the spatial scale was fixed as $R_0/2$. The stencil was built in such a way as to include only gridpoints within the computational domain, belonging either to cut or full cells. It can be observed that the estimated convergence rates for the values of the vector components are always above first order in the l_∞ norm, while the derivative approximation displays estimated convergence rates of approximately first order in the same norm. Comparing the results with those obtained with the linear interpolator described in Section 5, it was observed that, in general, the errors obtained with the RBF interpolators were about one order of magnitude smaller in cut cells or cells close to the boundary. The typical error distribution around the cylinder is shown in Fig. 10 at resolutions $\Delta x = 0.25$ m (see Table 6).

It should be remarked again that this can be considered as a worst case scenario, since the velocity field considered has a highly localized maximum at the downstream boundary of the cylinder. As in Section 6, a milder test can be performed by considering solid body rotation around the cylinder. In this case, the radial basis function gives a relative l_∞ error equal to zero, within machine accuracy, even on the coarsest of the grids considered in the previous test. This shows that a basic property of RBF interpolators with linear polynomial constraint (i.e., exact reconstruction of linear functions) is preserved also on grids with cut cells.

8. Numerical tests on the accuracy of semi-implicit semi-Lagrangian models

Several numerical tests were performed with two-dimensional models of fluid flow, in order to assess the effective accuracy of discretizations of the complete nonlinear equations of motion using the semi-implicit discretization outlined in Section 2 and the reconstruction algorithms introduced in the previous sections. Nonlinear advection is computed by a semi-Lagrangian scheme, as it would happen in a realistic application. The tests considered include two-dimensional open channel flow, as an example of an application to

Table 5
Accuracy of RBF interpolators for the x component of the velocity on a Cartesian grid with cut cells

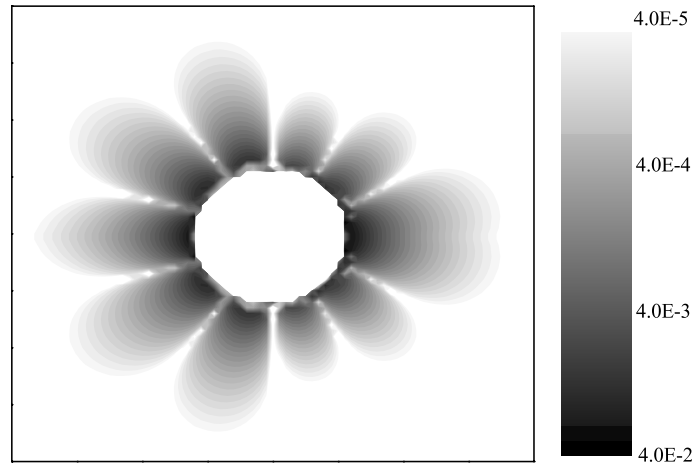| Δx	l_∞ Error	l_2 Error	Estimate of convergence rate in l_∞	Estimate of convergence rate in l_2
1	3.7×10^{-2}	3.1×10^{-4}	–	–
0.5	1.1×10^{-2}	4.5×10^{-5}	1.8	2.8
0.25	2.8×10^{-3}	4.4×10^{-6}	1.9	3.4
0.125	1.0×10^{-3}	5.5×10^{-7}	1.5	3.0

Fig. 10. Error distribution in RBF reconstruction of potential flow field (*U* component) at resolution $\Delta x = 0.25$ m. The color scale is logarithmic.

Table 6
Accuracy of RBF interpolators for $\partial u / \partial x$ reconstruction on a Cartesian grid with cut cells

Δx	l_∞ Error	l_2 Error	Estimate of convergence rate in l_∞	Estimate of convergence rate in l_2
1	5.7×10^{-1}	1.6×10^{-3}	–	–
0.5	3.3×10^{-1}	5.5×10^{-4}	0.8	1.6
0.25	1.8×10^{-1}	1.3×10^{-4}	0.9	2.0
0.125	1.0×10^{-1}	4.7×10^{-5}	0.8	1.5

free surface hydrodynamics, and stratified air flow over an obstacle, in the context of typical lee wave test cases for an $x - z$ vertical slice mesoscale model.

8.1. Open channel flow

As a first test, uniform flow in a straight open channel with rectangular section was considered. The channel was oriented at $\alpha = 22.8°$ with respect to a rectangular Cartesian mesh with $\Delta x = \Delta y$ equal to 1 m. The width of the channel was taken to be $10 \cos \alpha$ m and the discharge at the inflow boundary was 10 m^3 s^{-1}; the Chézy coefficient was taken to be equal to 70 m$^{1/2}$ s^{-1} and the bottom slope i_b was taken to be 0.001. The analytical solution is given by a water depth of 0.618 m and the velocity field with absolute value equal to 1.741 m s^{-1} and inclination parallel to the channel boundaries. The Froude number was equal to 0.707 and therefore the flow was subcritical.

For this test, the linear reconstruction algorithm introduced in Section 5 was applied for the computation of the trajectories and for the interpolation at the departure points $\vec{x}^*$. Simulations were performed using a timestep equal to 2 s and therefore the Courant numbers with respect to the velocity were equal to 3.23 and 1.28 in x and y directions, respectively. At the upstream boundary, the water discharge was imposed. For Lagrangian paths leaving the domain, ghost cells were added at the beginning of the channel assuming uniform flow with the same features as in the first row of cells inside the computational domain. At the downstream boundary, the water depth was prescribed. The simulations were initialized with various possible initial conditions and were run until a steady solution was found. Results were found to be

independent on the initial conditions. The relative errors in the l_∞ and l_2 norm with respect to the analytic solution for the water depth and the components of the velocity vector are shown in Table 7 (absolute errors in the case of the vertical component ζ of the vorticity, whose exact value is 0).

It can be observed that the analytic solution is very well captured even though the number of square cells in the cross flow direction is relatively small and the Courant number in each direction is relatively high. Again, this is essential when trying to approximate complex channel networks without using unstructured grids.

Another test was run with analogous boundary condition in the case of a curved channel. The channel had rectangular section $b = 2.5$ m wide, slope $i_b = 0.35 \times 10^{-3}$ and the Chézy coefficient was taken to be equal to 83.4 m$^{1/2}$ s^{-1}. Two orthogonal straight reaches were connected with a curved one, bounded by two quarters of circle with radii 12.5 and 15 m, respectively. A rectangular Cartesian mesh with $\Delta x = \Delta y$ equal to 0.1 m was used. The Froude number in the downstream straight reach was equal to 0.49 and therefore the flow was subcritical. Simulations were performed using a timestep equal to 0.9 s and therefore the Courant number (with respect to the velocity) was approximately equal to 4.8. The simulations were initialized with various initial conditions (all very different from the steady state solution) and were run until a steady state solution was found. The same steady state results were obtained with all the initial conditions. The water depth field is shown in Fig. 11(a), which can be compared with the analogous result obtained without the cut cell interpolation, shown in Fig. 11(b). It can be observed that the spurious effects at the boundaries have disappeared entirely, along with the spurious jump in the free surface elevation that can be seen in the full cell simulation. For this case, no analytic solution is available. However, it has been verified, also experimentally, that in a radial section of the curve at 45° of inclination, the flow can be approximated with a free vortex with potential flow (see e.g. [14,21]). In this situation, the difference $\Delta\eta$ in the free surface level between the inner and the outer side of the channel can be obtained by integrating Euler's equation radially between the inner (r_i) and the outer (r_o) value of the radius of curvature: $\Delta\eta = \frac{1}{g} \int_{r_i}^{r_o} \frac{U^2}{r} dr$. This integral can be approximated assuming a constant value for the velocity equal to the value in the middle streamline (i.e., the value U_0 obtained in a straight channel with the same bottom slope of the curved channel). For the test case, with $r_i = 12.5$, $r_o = 15.0$ and $U_0 = 0.532$, one obtains:

$$\Delta\eta = \frac{1}{9.81} U_0^2 \ln\left(\frac{r_o}{r_i}\right) = 5.260 \times 10^{-3} \text{ m}.$$

The corresponding values computed by the cut cell model are shown in Table 8 for various horizontal resolutions. It can be observed that the computed values converge to a value that is close to the estimated value. More generally, convergence to the finest resolution results was observed throughout the computational domain. In the simulations at different resolutions, the timestep was chosen so that the Courant number remained constant and approximately equal to 4.8.

The convergence of the free surface profiles in a radial section of the curve at 45° of inclination is also shown in Fig. 12, where it can also be observed that the values of the free surface in the middle of the channel converge to the constant steady state value as predicted by the theory (see again e.g. [21]). For this test, the linear reconstruction algorithm introduced in Section 5 was applied for the computation of the trajectories, while the RBF interpolator described in Section 7 was used for the interpolation at the departure

Table 7
Relative errors with respect to analytical solution for steady state straight channel flow computed with cut cells (absolute errors in the case of vorticity)

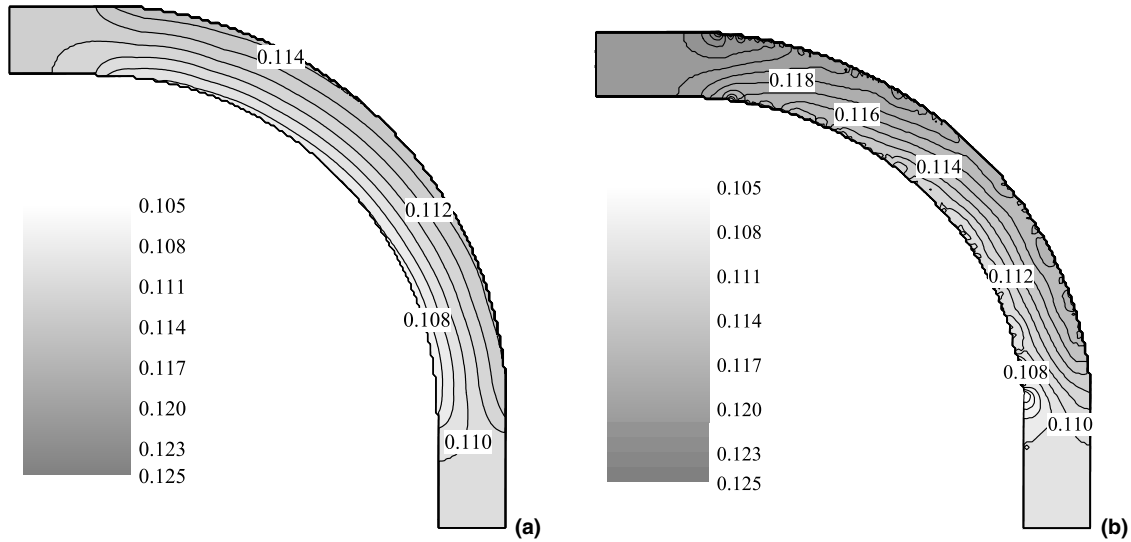|	water depth	U	V	ζ
l_∞	6.8×10^{-5}	3.55×10^{-4}	2.53×10^{-2}	1.3×10^{-3}
l_2	3.20×10^{-8}	1.60×10^{-7}	5.31×10^{-8}	2.47×10^{-3}

Fig. 11. Water depth (m) at steady state for open channel flow in a curved channel with cut boundary cells (a) and with full boundary cells (b).

Table 8
Values of $\Delta\eta$ computed for steady state curved channel flow computed with cut cells at various horizontal resolutions

	$b/\Delta x = 5$	$b/\Delta x = 10$	$b/\Delta x = 25$	$b/\Delta x = 50$	$b/\Delta x = 100$
$\Delta\eta$ [m]	4.23×10^{-3}	4.36×10^{-3}	5.00×10^{-3}	5.22×10^{-3}	5.23×10^{-3}

points $\vec{x}^{*}$. It is to be remarked that the role of the linear reconstruction is essential in imposing the appropriate boundary condition. If for example the RBF interpolator is used in the trajectory computation as well, erroneous results similar to those in Fig. 11(b) are obtained.

8.2. Stratified mesoscale flows

The velocity reconstruction algorithms described above were also implemented in the context of the semi-implicit, semi-Lagrangian scheme for the fully elastic Euler equations described in [4]. An implementation of this discretization on an x–z two dimensional vertical slice was used. The cells intersected by the lower boundary were treated as cut cells in the sense explained in the previous sections. The hydrostatic lee wave test case analyzed in [16] was considered first. An initially uniform horizontal flow with horizontal speed at inflow equal to $U = 10 \text{ m s}^{-1}$ hits a hill whose shape is given by a versiera of Agnesi curve and whose height and half-width are $h_0 = 400$ m and $a = 10{,}000$ m, respectively. A uniformly stratified atmosphere is assumed, with Brunt–Väisälä frequency equal to $N = 10^{-2} \text{ s}^{-1}$. The vertical resolution was taken to be $\Delta z = 150$ m and the horizontal resolution was taken to be $\Delta x = 2000$ m. All the further details concerning the test case can be found in [16]. In the lowermost cells of the computational domain, the linear reconstruction algorithm described in Section 5 was applied for the computation of the trajectories, while the RBF interpolator described in Section 7 was used for the interpolation at the departure points $\vec{x}^{*}$. The stencils used for the RBF interpolator include up to 16 points surrounding the cells where interpolation is to be performed. In cells away from the boundary, standard bilinear interpolation was used for the computation of the trajectories, while cubic Lagrange tensor product interpolation was used
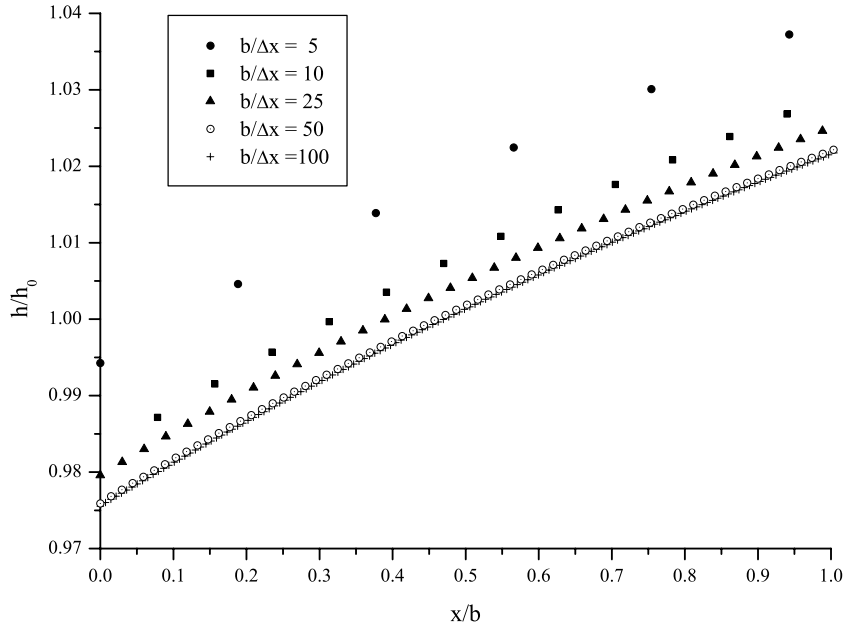
Fig. 12. Convergence of free surface profiles in a curved channel flow (radial section at 45°) with cut boundary cells. Profiles are normalized with the constant steady state value h_0 in the downstream straight reach.

for the interpolation at the departure points $\vec{x}^*$ (see e.g. [4,30]). The results in Section 7 would justify the application of RBF interpolation throughout the whole computational domain. However, since cubic Lagrange tensor product interpolation is generally regarded as sufficiently accurate and considering that effi-
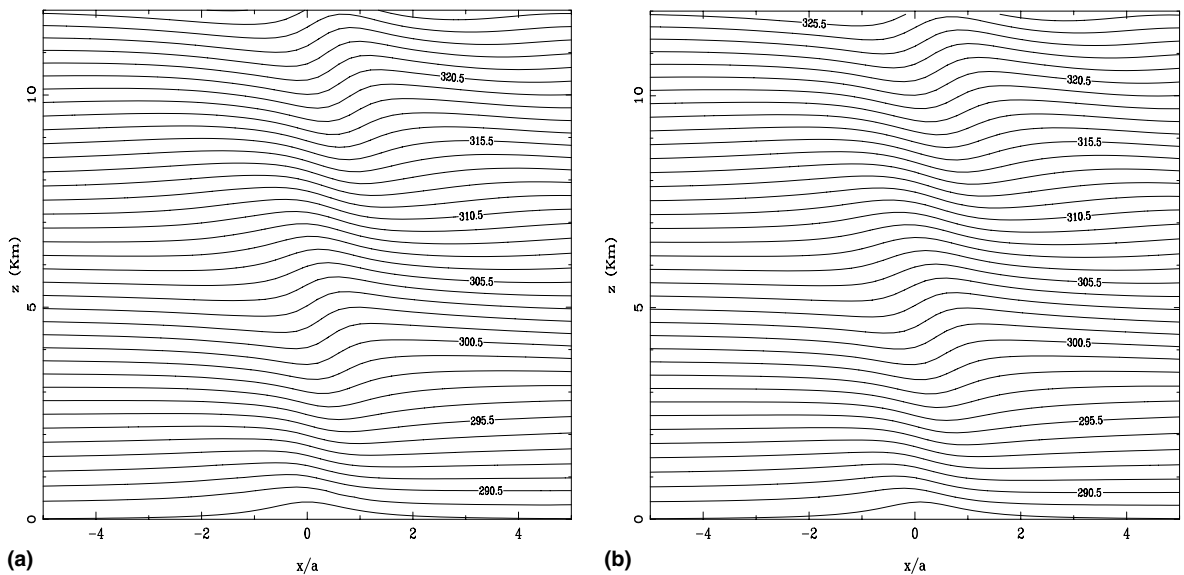

Fig. 13. Potential temperature field in the hydrostatic lee wave test case, computed with cut cell model (a) and with LM terrain following model (b). Contour spacing is 1 K.

cient implementation of RBF requires storage of a number of auxiliary fields, the simpler Lagrange inter-
polation approach was used away from the boundaries. The results obtained with the cut cell approxi-
mation are shown along with a reference solution obtained with Lokal Modell (shortly LM, see e.g. [38]),
a mesoscale model developed at the German Weather Forecasting Service (DWD), which is used oper-
ationally for local weather forecasting at many European weather centers including ARPA Bologna. LM
employs terrain following coordinates, a split-explicit time discretization for the sound waves and a spa-
tial discretization based on centered finite differences. It employs temperature and pressure as prognostic
variables, while the semi-implicit, semi-Lagrangian cut cell model uses Exner pressure and potential tem-
perature. In Fig. 13 the contours of the potential temperature field are shown, which correspond to the
streamlines in this adiabatic case. In Fig. 14 the deviation of the horizontal velocity component from the
mean inflow value is shown. In Fig. 15 the vertical velocity component is shown, while in Fig. 16 a plot
of the $\zeta = \partial u/\partial z - \partial w/\partial x$ component of vorticity is shown. It can be observed that the results are in good
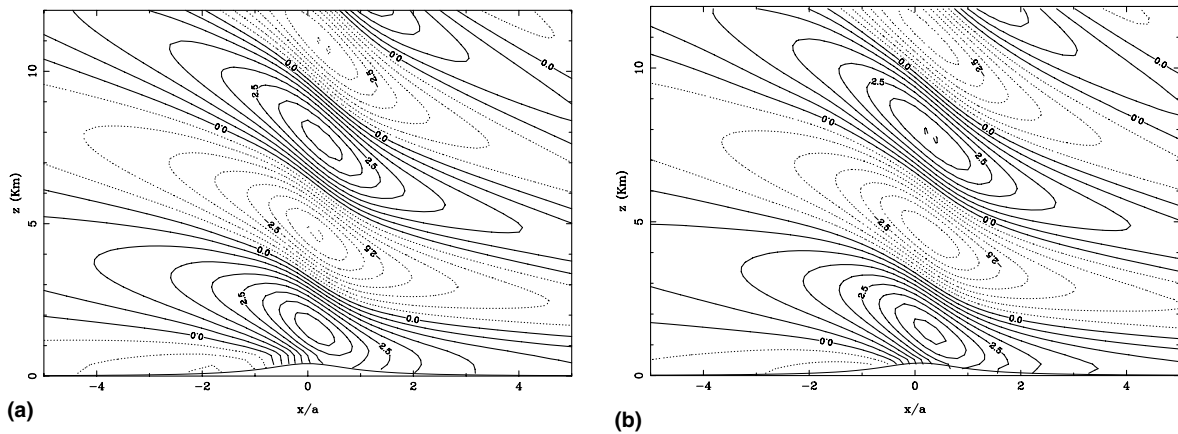

Fig. 14. Deviation of u velocity component from the inflow value in the hydrostatic lee wave test case, computed with cut cell model (a) and with LM terrain following model (b). Contour spacing is 0.5 m s^{-1}.
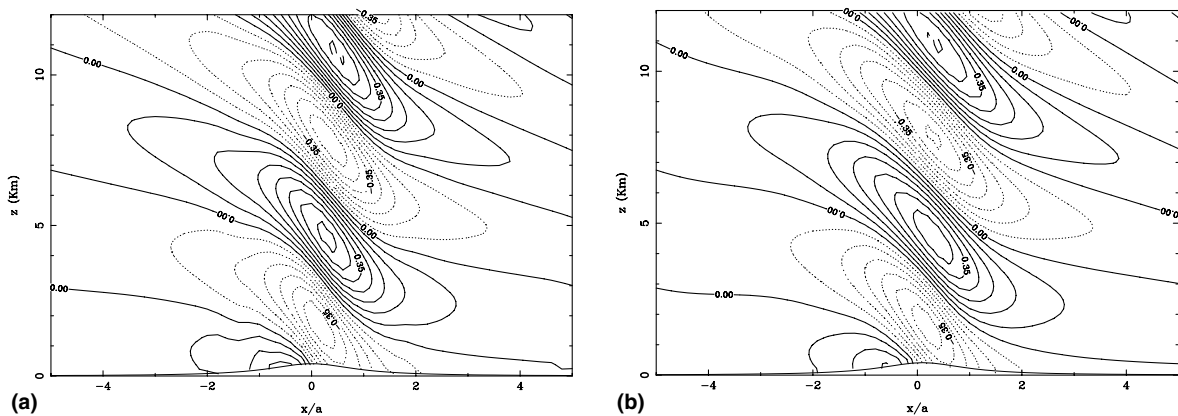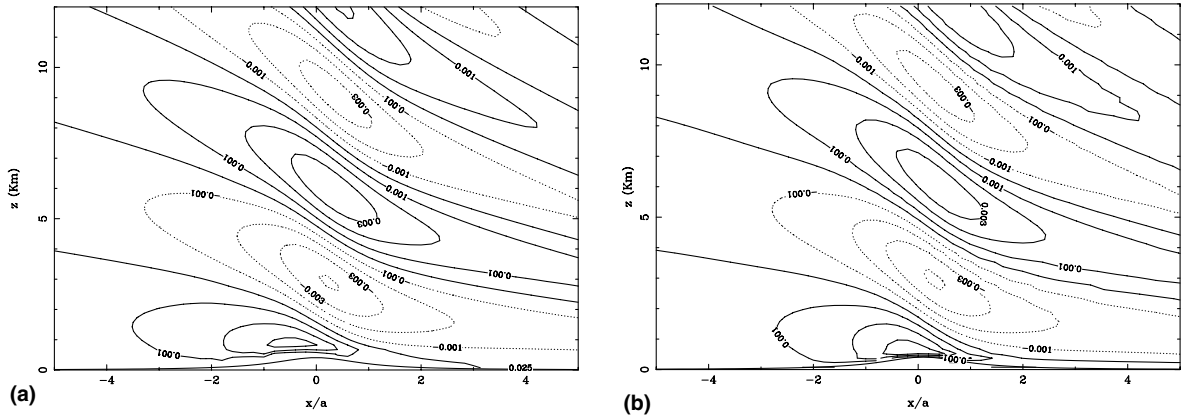

Fig. 15. w velocity component in the hydrostatic lee wave test case, computed with cut cell model (a) and with LM terrain following model (b). Contour spacing is 0.07 m s^{-1}.

Fig. 16. Vorticity in the hydrostatic lee wave test case, computed with cut cell model (a) and with LM terrain following model (b). Contour spacing is 10^{-3} s^{-1}.

qualitative and quantitative agreement with the reference solution. Both the cut cell results and those computed with LM are in good agreement with the analytic solution for the corresponding linearized equations, as reported for example in [15,16]. For example, the lowest minimum in the vertical velocity field has the same value around -0.45 m s^{-1} at a height of approximately 1700 m. The nonhydrostatic lee

Fig. 17. w velocity component in the nonhydrostatic lee wave test case, computed with cut cell model. Contour spacing is 0.25 m s^{-1}.

wave test case analyzed in [16] was also considered. The setup is the same as in the previous case, but the mountain half-width is now and $a = 1000$ m. The vertical resolution was taken to be $\Delta z = 150$ m and the horizontal resolution was taken to be $\Delta x = 200$ m. In Fig. 17 the vertical velocity component is shown. It can be observed again that the results are in good qualitative and quantitative agreement with the analytic solution for the corresponding linearized equations (see e.g. [16], Fig. 1(c)). Furthermore, it can be observed that these results are achieved at a much coarser vertical resolution than used for example in the nonhydrostatic Eta model discussed in [16], for which $\Delta z = 10$ m was necessary to obtain comparable results (see Figs. 7(a) and (b)).

9. Conclusions and future developments

Semi-implicit, semi-Lagrangian models for environmental applications have been extended to staggered Cartesian meshes with cut boundary cells. Radial basis function interpolation and a linear reconstruction algorithm have been used to recover an accurate velocity field close to the boundary. Timestep control techniques in the computation of the semi-Lagrangian trajectories have also been proposed. The accuracy of the resulting numerical method has been assessed in a number of test cases, relevant for river hydraulics and mesoscale atmospheric modelling. Extensions of the techniques introduced here are being investigated for the application to three-dimensional numerical models of coastal hydrodynamics and mesoscale flow. In order to extend the algorithm to the three-dimensional case in a straightforward way, it is planned to develop linear reconstruction procedures analogous to that of Section 5 for the whole stack of cells intersected by the bathymetry or orography.

Acknowledgements

References

[1] A. Adcroft, C. Hill, J. Marshall, Representation of topography by shaved cells in a height coordinate ocean model, Monthly Weather Review 125 (1997) 2293–2315.
[2] P. Bartello, S.J. Thomas, The cost effectiveness of semi-Lagrangian advection, Monthly Weather Review 124 (1996) 2883–2895.
[3] J. Behrens, A. Iske, Grid-Free Adaptive semi-Lagrangian advection using radial basis functions, Computers and Mathematics with Applications 43 (2002) 319–327,.
[4] L. Bonaventura, A semi-implicit, semi-Lagrangian scheme using the height coordinate for a nonhydrostatic and fully elastic model of atmospheric flows, Journal of Computational Physics 158 (2000) 186–213.
[5] L. Bonaventura, A second order, semi-Lagrangian scheme with accurate approximation of trajectories, in: C. Taylor, J. Cross (Eds.), Proceedings of the 10th International Conference on Numerical Methods in Fluids, Pineridge Press, Swansea, 1997.

[6] M.D. Buhmann, Radial basis functions, Acta Numerica (2000) 1–38.

[7] D. Calhoun, R.J. LeVeque, A Cartesian grid finite-volume method for the advection diffusion equation in irregular geometry, Journal of Computational Physics 157 (2000) 143–180.

[8] V. Casulli, Semi-implicit finite difference methods for the two dimensional shallow water equations, Journal of Computational Physics 86 (1990) 56–74.

[9] V. Casulli, E. Cattani, Stability, accuracy and efficiency of a semi-implicit method for three-dimensional shallow water flow, Computers and Mathematics with Applications 27 (1994) 99–112.

[10] V. Casulli, G.S. Stelling, Numerical simulation of 3D quasi-hydrostatic free-surface flows, Journal of Hydraulic Engineering ASCE 124 (1998) 678–686.

[11] V. Casulli, A semi-implicit finite difference method for non-hydrostatic, free-surface flows, International Journal of Numerical Methods in Fluids 30 (1999) 425–440.

[12] D.M. Causon, D.M. Ingram, C.G. Mingham et al., Calculation of shallow water flows using a Cartesian cut cell approach, Advances in Water Resources Research 23 (2000) 545–562.

[13] D.M. Causon, D.M. Ingram, C.G. Mingham, A Cartesian cut cell method for shallow water flows with moving boundaries, Advances in Water Resources Research 24 (2001) 899–911.

[14] V.T. Chow, Open Channel Hydraulics, McGraw-Hill, New York, 1959.

[15] J. Dudhia, A nonhydrostatic version of the Penn State – NCAR mesoscale model: validation, tests and simulation of an Atlantic cyclone and cold front, Monthly Weather Review 121 (1993) 1493–1513.

[16] W. Gallus, J. Klemp, Behaviour of flow over step orography, Monthly Weather Review 128 (2000) 1153–1164.

[17] F.X. Giraldo, Trajectory computations for spherical geodesic grids in Cartesian space, Monthly Weather Review 127 (1999) 1651–1662.

[18] F.X. Giraldo, The Lagrange–Galerkin method for the two-dimensional shallow water equations on adaptive grids, International Journal of Numerical Methods in Fluids 33 (2000) 789–832.

[19] E.S. Gross, J.R. Koseff, S.G. Monismith, Three-dimensional salinity simulations of South San Francisco Bay, ASCE Journal of Hydraulic Engineering 125 (1999) 1199–1209.

[20] F.H. Harlow, J.E. Welch, Numerical calculation of time dependent viscous incompressible flow, Physics of Fluids 8 (1965) 2182–2189.

[21] F.M. Henderson, Open Channel Flow, MacMillan, New York, 1966.

[22] The Hamburg Ocean Primitive Equation Model HOPE, DKRZ Report No. 13.

[23] A. Iske, On the approximation order and numerical stability of local Lagrange interpolation by polyharmonic splines, in: W. Haussmann, K. Jetter, M. Reimer, J. Stöckler (Eds.), Modern Developments in Multivariate Approximation, International Series of Numerical Mathematics, Birkhäuser Verlag, Basel, 2003, pp. 153–165.

[24] H. Johansen, P. Colella, A Cartesian grid embedded boundary method for Poisson's equation on irregular domains, Journal of Computational Physics 147 (1998) 60–85.

[25] A. Kasahara, W.W. Washington, General circulation experiment with a six-layer NCAR model, Monthly Weather Review 28 (1971) 657–701.

[26] D.Y. Le Roux, C.C. Lin, A. Staniforth, An accurate interpolating scheme for semi-Lagrangian advection on an unstructured grid for ocean modelling, Tellus A 49 (1997) 119–138.

[27] D.Y. Le Roux, C.C. Lin, A. Staniforth, A semi-implicit, semi-Lagrangian finite element shallow water ocean model, Monthly Weather Review 128 (1998) 1384–1401.

[28] X. Li, C.A. Micchelli, Approximation by radial bases and neural networks, Numerical Algorithms 25 (2000) 242–262.

[29] E. Miglio, A. Quarteroni, F. Saleri, Finite element approximation of quasi-3d shallow water equations, Comparative Methods in Applied Mechanics and Engineering 174 (1999) 355–369.

[30] J.P. Pinty, R. Benoit, E. Richard, R. Laprise, Simple tests of a semi-implicit semi-Lagrangian model on 2D mountain wave problems, Monthly Weather Review 123 (1995) 3042–3058.

[31] O. Pironneau, On the transport-diffusion algorithm and its applications to the Navier–Stokes equations, Numerische Mathematik 38 (1982) 309–332.

[32] R. Schaback, Z. Wu, Local error estimates for radial basis function interpolation of scattered data, IMA Journal of Numerical Analysis 13 (1993) 13–27.

[33] C. Schär, O. Leuenberger, D. Fuhrer, D. Lüthi, C. Girard, A new vertical coordinate formulation for atmospheric prediction models, Monthly Weather Review 130 (2002) 2459–2480.

[34] P.K. Smolarkiewicz, J. Pudykiewicz, A class of semi-Lagrangian approximations for fluids, Journal of Atmospheric Sciences 49 (1992) 2082–2096.

[35] T. Sonar, Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws, IMA Journal of Numerical Analysis 16 (1996) 549–581.

[36] A. Staniforth, J. Coté, Semi-Lagrangian integration schemes for atmospheric models – a review, Monthly Weather Review 119 (1991) 2206–2223.

[37] J. Steppeler, H.W. Bitzer, M. Minotte, L. Bonaventura, Nonhydrostatic atmospheric modelling using a Z coordinate representation, Monthly Weather Review 130 (2002) 2143–2149.

[38] J. Steppeler, G. Doms, U. Schättler, Meso-gamma scale forecasts using the nonhydrostatic model LM, Meteorology and Atmospheric Physics 82 (2003) 75–96.

[39] P.F. Wang, R.T. Cheng, K. Richter, E.S. Gross, D. Sutton, J.W. Gartner, Modeling tidal hydrodynamics of San Diego Bay California, Journal of the American Water Resources Association 34 (1998) 1123–1140.