CWI

A two-dimensional embedded-boundary method for convection problems with moving boundaries

Y.J. Hassen, B. Koren

MAC-1003

Centrum Wiskunde & Informatica

Centrum Wiskunde & Informatica (CWI) is the national research institute for Mathematics and Computer Science. It is sponsored by the Netherlands Organisation for Scientific Research (NWO).
CWI is a founding member of ERCIM, the European Research Consortium for Informatics and Mathematics.

CWI's research has a theme-oriented structure and is grouped into four clusters. Listed below are the names of the clusters and in parentheses their acronyms.

Probability, Networks and Algorithms (PNA)

Software Engineering (SEN)

Modelling, Analysis and Computing (MAC)

Information Systems (INS)

P.O. Box 94079, 1090 GB Amsterdam (NL)
Science Park 123, 1098 XG Amsterdam (NL)
Telephone +31 20 592 9333
Telefax +31 20 592 4199

ISSN 1386-3703

A two-dimensional embedded-boundary method for convection problems with moving boundaries[☆]

Yunus Hassen[*], Barry Koren

Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Abstract

In this work, a two-dimensional embedded-boundary algorithm for convection problems is presented. A moving body of arbitrary boundary shape is immersed in a Cartesian finite-volume grid, which is fixed in space. The boundary surface is reconstructed in such a way that only certain fluxes in the immediate neighbourhood indirectly accommodate effects of the boundary conditions valid on the moving (immersed-)body. Over the majority of the domain, where these boundary conditions have 'no' effect, the fluxes are computed using standard schemes. We employ the method of lines, with higher-order spatial discretizations and the explicit Euler scheme for the time integration. To validate the method, two cases, a *rectilinear discontinuity* of arbitrary orientation, moving in a uniform two-dimensional flow-field, and a *cylindrical discontinuity* of arbitrary initial location, moving in a circular flow-field, are considered. The simulations show promising, globally accurate solutions. It is anticipated that the algorithm can be used for 2D Euler flows,

[☆] The first author's research is funded by the *Delft Centre for Computational Science & Engineering (DCSE)*, TU Delft.

[*] Corresponding author. Science Park 123, 1098 XG Amsterdam, The Netherlands.

Email addresses: `yunus.hassen@cwi.nl` (Yunus Hassen), `barry.koren@cwi.nl` (Barry Koren)

which we foresee to consider next.

Keywords: hyperbolic conservation laws, convection, finite-volume method, immersed-boundary method, embedded boundaries, dimensional splitting

1. Introduction

There are a variety of numerical methods today that can be used to solve fluid-flow problems involving both fluid and solid motions as well as their interactions. Recently, immersed-boundary methods have been favourably popularised by their relatively simple ideas and ease of implementation.

The immersed-boundary method, also synonymously known as embedded-boundary method, in general, is a method in which boundary conditions are indirectly incorporated into the governing equations. It is very suitable for simulating flows around flexible, moving and/or complex bodies (see [1] for a comprehensive review).

Basically, the boundaries of the bodies of interest are just embedded in non-deforming Cartesian grids that do not conform to the shape of the body. The governing equations are modified to include the effect of the embedded boundaries (EBs). By doing so, mesh generation or regeneration difficulties associated with the body-fitted grids are obviated, and the underlying regular fixed-grid allows us to use a simple data structure as well as simpler numerical schemes over a majority of the domain.

In this work, we present a new embedded-boundary approach for advection problems. As is standard in the immersed-boundary methods, moving bodies are embedded in a fixed, Cartesian grid. We employ the method of lines: a higher-order, cell-averaged, fixed-grid, finite-volume method for the spatial discretization, and the explicit Euler scheme for the time integration. The essence of the present method is that body geometries are, without loss of generality, effectively simplified and their presence is restricted to a minimal zone in the computational region so that standard discretization schemes can be readily applied elsewhere. The boundary conditions valid on a possibly moving body are indirectly accommodated by specific fluxes in the vicinity of the boundary.

The purpose of this work is to present a generic framework for the construction of an algorithm for treating (moving or stationary) bodies of arbitrary shape, immersed inside a flowing fluid. This paper focuses on the geometric boundary-reconstruction and treatment. Analyses and implementations of the numerical flux formulae have already been dealt with in previous works. Readers are referred to [2, 3].

The outline of the paper is as follows. The embedded-boundary method is described in § 2, and a brief exposition of the finite-volume method is given in § 3. Section 4 contains the definition, set-up and results of the numerical experiments, which demonstrate the performance of the current algorithm. Some global errors in different norms are also presented; analysis is deferred to future papers. Concluding remarks are given in § 5.

2. The embedded boundary method

As in the previous one-dimensional work [3], our approach uses a finite-volume discretization that embeds the boundary of a body in a regular, fixed grid. Dividing the current 2D computational domain $\mathcal{D}$, of dimension $\ell_x \times \ell_y$, into $N_x \times N_y$ uniform, rectangular finite volumes that are fixed in space, we

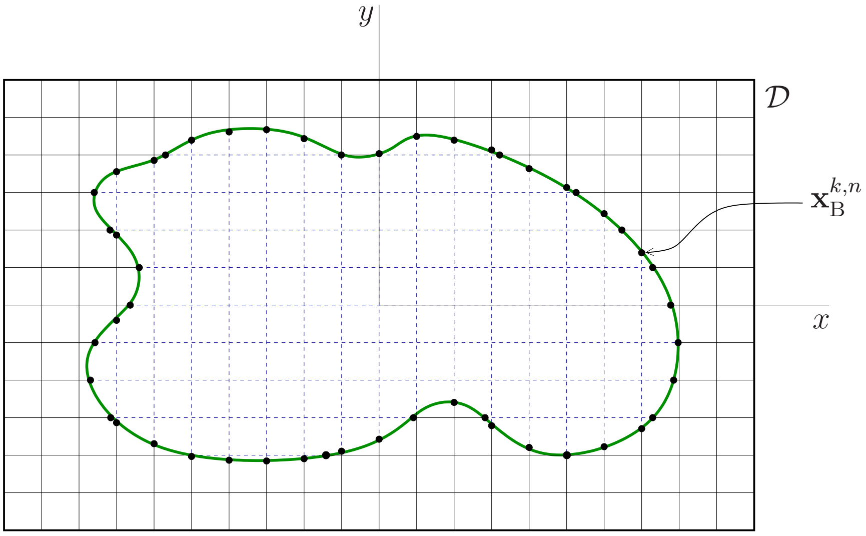


Figure 1: Immersion of a body in a Cartesian, finite-volume grid and detection of boundary intersection points with cell faces, at time t^n.

have:

$$\begin{aligned}\mathcal{D}_{i,j} :=& \Big\{ [x_{_{i-\frac{1}{2}}}, x_{_{i+\frac{1}{2}}}] \times [y_{_{j-\frac{1}{2}}}, y_{_{j+\frac{1}{2}}}] \;\Big|\; x_{_{i-\frac{1}{2}}} = \big(i - \tfrac{1}{2}\big)\, h_x + x_0, \\ & y_{_{j-\frac{1}{2}}} = \big(j - \tfrac{1}{2}\big)\, h_y + y_0;\; i \in [1, N_x],\; j \in [1, N_y] \Big\}, \qquad (1)\end{aligned}$$

where $h_x = \ell_x/N_x$ and $h_y = \ell_y/N_y$ are cell sizes, and x_0 and y_0 some constants. The total simulation time $\mathcal{T} > 0$ is equally divided into N_t time steps of size $\tau = \mathcal{T}/N_t$.

Having generated the (Cartesian) grid, the body is immersed inside the grid, see example in Figure 1. To obtain discrete embedded boundaries (EBs), at a given time t^n, $n = 0, 1, \cdots, N_t$, firstly, the finite volumes that contain (a part of) the boundary of the immersed body are identified, and then the points of intersection of the boundary of the immersed body with the faces of these computational cells, $\mathbf{x}_{\mathrm{B}}^n$, are detected. That is, the coordinates of the boundary points $\mathbf{x}_{\mathrm{B}}^{k,n} := (x_{\mathrm{B}}^{k,n}, y_{\mathrm{B}}^{k,n})$, $k = 1, 2, \cdots, N_{\mathrm{B}}$, where N_{B} is the total number of boundary-face intersections, are computed.

2.1. Round-off error treatment

Care is required in making the underlying uniform fixed grid detect boundary points that lie exactly at or very close to grid vertices. Grid vertices are shared by more than one control volume, and a boundary point lying exactly at a grid vertex, or in the immediate neighbourhood, is, due to the round-off error, arbitrarily assigned to any of the cells that share the vertex. The prevailing arbitrariness can lead to erroneous absence of an EB in a cell, due to discount of a second boundary point (or two boundary points altogether) within the cell. By taking the precision of the machine into account, this undesirable scenario is remedied in the following manner.

For a given machine-precision number ζ, in our case $\zeta = 2.22 \times 10^{-16}$, a *region of round-off error* can be defined. Given a grid vertex with coordinate $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$, the *region* is computed, for a magnified $\tilde{\zeta}$, in our case $\tilde{\zeta} = 10^7\zeta$, where it is assumed that $\tilde{\zeta} \ll h_x$ and $\tilde{\zeta} \ll h_y$, as: $(x_{\mathrm{B}}^{k,n} - x_{i+\frac{1}{2}})^2 + (y_{\mathrm{B}}^{k,n} - y_{j+\frac{1}{2}})^2 = \tilde{\zeta}^2$. Any boundary point $\mathbf{x}_{\mathrm{B}}^{k,n} = (x_{\mathrm{B}}^{k,n}, y_{\mathrm{B}}^{k,n})$ that lies inside the *region* is explicitly pegged 'just-outside' it.

To avoid overruling of correctly identified boundary points and further complications, in case of two boundary points in the *regions* of two (or more) consecutive grid vertices (Figure 2a), we propose a staggered approach, see Figure 2b. This results in one or more boundary points, situated inside the *region of round-off error*, to be exclusively visible to all cells that share the vertex. For instance, in the example of Figure 2a, we have five boundary points, four of them inside *regions* defined by $\tilde{\zeta}$. Taking the left-most bound-

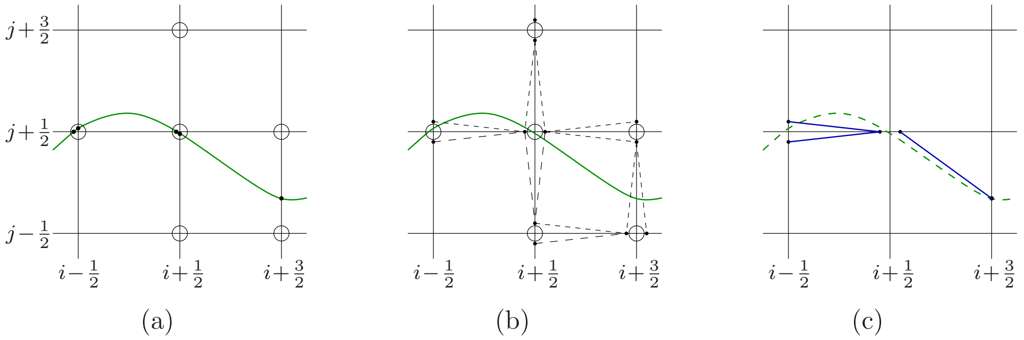


Figure 2: Round-off error treatment for EB points that lie exactly at or very close to grid vertices.

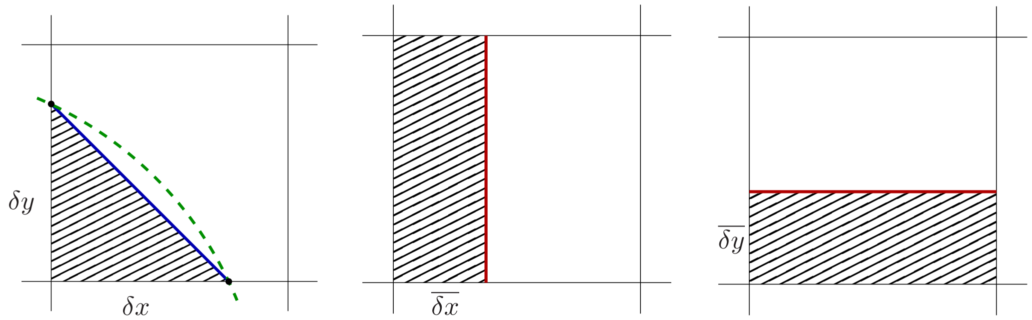
ary point as the k^{th} point, since $\Big\{(x_{\mathrm{B}}^{k,n} - x_{i-\frac{1}{2}})^2 + (y_{\mathrm{B}}^{k,n} - y_{j+\frac{1}{2}})^2,\ (x_{\mathrm{B}}^{k+1,n} - x_{i-\frac{1}{2}})^2 + (y_{\mathrm{B}}^{k+1,n} - y_{j+\frac{1}{2}})^2,\ (x_{\mathrm{B}}^{k+2,n} - x_{i+\frac{1}{2}})^2 + (y_{\mathrm{B}}^{k+2,n} - y_{j+\frac{1}{2}})^2,\ (x_{\mathrm{B}}^{k+3,n} - x_{i+\frac{1}{2}})^2 + (y_{\mathrm{B}}^{k+3,n} - y_{j+\frac{1}{2}})^2\Big\} < \tilde{\zeta}^2$, we then have: $\Big\{\mathbf{x}_{\mathrm{B}}^{k,n},\ \mathbf{x}_{\mathrm{B}}^{k+1,n},\ \mathbf{x}_{\mathrm{B}}^{k+2,n},\ \mathbf{x}_{\mathrm{B}}^{k+3,n}\Big\} := \Big\{(x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}} - \tilde{\zeta}),\ (x_{i-\frac{1}{2}}, y_{j+\frac{1}{2}} + \tilde{\zeta}),\ (x_{i+\frac{1}{2}} - \tilde{\zeta}, y_{j+\frac{1}{2}}),\ (x_{i+\frac{1}{2}} + \tilde{\zeta}, y_{j+\frac{1}{2}})\Big\}$. Since $(x_{\mathrm{B}}^{k+4,n} - x_{i+\frac{3}{2}})^2 + (y_{\mathrm{B}}^{k+4,n} - y_{j+\frac{1}{2}})^2 \not< \tilde{\zeta}^2$ and also $(x_{\mathrm{B}}^{k+4,n} - x_{i+\frac{3}{2}})^2 + (y_{\mathrm{B}}^{k+4,n} - y_{j-\frac{1}{2}})^2 \not< \tilde{\zeta}^2$, the coordinate of the right-most boundary point, $\mathbf{x}_{\mathrm{B}}^{k+4,n} = (x_{\mathrm{B}}^{k+4,n}, y_{\mathrm{B}}^{k+4,n})$, remains unchanged. Further details will be given in a future paper.

Notice that, this procedure may result in two EBs adjacent to a cell face, say a twin, as in Figure 2c. A scenario like this can also emerge from boundary points which lie outside the *regions of round-off error*. This is discussed and taken care of in § 2.3.

2.2. Determination of EBs

Once the boundary points are effectively detected, the actual boundary of the body is readily degenerated as a piece-wise continuous, closed or open, poly-line, with N_{B} and $N_{\mathrm{B}}-1$ segments, respectively. This representation facilitates explicit association of each side/segment of the polygon/poly-line with individual control volumes, resulting in one discrete EB, at most, in a cell.

An example of a two-dimensional EB segment is shown Figure 3a. To take advantage of the one-dimensional method proposed in [3], we resort to dimensional splitting. Dimensional splitting has already been successfully applied in many discretization methods for convection operators; and, as such, it is a commonly used practice in CFD.



(a) Continuous and discrete EB

(b) $\delta x \leq \delta y$

(c) $\delta x > \delta y$

Figure 3: Alignment of a two-dimensional EB, situated in a cell, with the relevant grid (coordinate) direction.

The manner in which a generic two-dimensional EB, situated in a cell, is projected on to the grid (coordinate) directions is crucial. Our method of projection is concisely described as follows.

To project the discrete EB, shown in Figure 3a, into the relevant grid direction and to get a single orthogonal EB in a cell, the following two geometrical properties are used:

1. *Orientation*: the relative inclination (of the EB) with respect to the orthogonal coordinate axes, and
2. *Location*: the area the EB subsets in the cell.

In Step 1, by determining to which orthogonal direction the EB is closer to, as shown in Figure 3, the EB is orthogonalized in the either direction. For instance, if $\delta x > \delta y$, the EB is aligned in the x-direction (Figure 3c). In retrospect, as crude approximation as this sounds, the fluxes at the vertical faces of the cell still feel the effects of the orthogonalized EB (indirectly) through the average solution the cell holds. Whereas, the fluxes at the horizontal faces indeed feel the orthogonalized EB directly. Vice versa, the same holds good for vertical faces instead of horizontal faces, and for horizontal faces instead of vertical faces, if $\delta x < \delta y$. If $\delta x = \delta y$, the EB is equally inclined to both (grid) directions, and it can be arbitrarily projected on any of the axes. For consistency, we invariably align the EB, under such circumstances, with the y-direction (Figure 3b).

In Step 2, the actual location of the orthogonalized EB, at t^n, inside a cell is represented by $\boldsymbol{\beta}^n = (\beta_x^n, \beta_y^n)$, a normalised variable that discerns the orthogonalized EB's position relative to the left- or bottom-face of the cell [3]. It captures the geometric data of all EBs at a given time, and it is the primary variable that passes EB information to the relevant fluxes (§ 3.1). It is determined by the area, subset in the cell by the non-orthogonalized EB. This is beneficial for conservation.

Pure solid cells are assigned with $\beta_x^n = \beta_y^n := 2$, and pure fluid cells with $\beta_x^n = \beta_y^n := -2$. Whereas, for cells with an EB, depending on the orientation of the orthogonalized EB, either $\beta_x^n := 2$ or $\beta_y^n := 2$, for largely solid cells; and either $\beta_x^n := -2$ or $\beta_y^n := -2$, for largely fluid cells. For the example depicted in Figure 3, it holds:

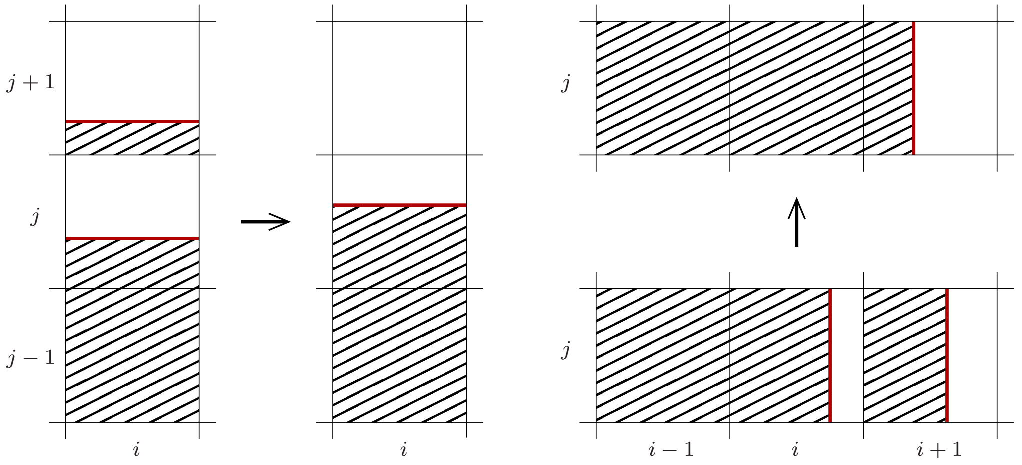
- $\overline{\delta x} = \frac{1}{2}\delta x\, \delta y/h_y$, resulting in $\beta_x^n := \overline{\delta x}/h_x$ and $\beta_y^n := -2$ if $\beta_x^n \in [0, 1/2)$, or $\beta_y^n := 2$ if $\beta_x^n \in [1/2, 1]$, Figure 3b, and
- $\overline{\delta y} = \frac{1}{2}\delta x\, \delta y/h_x$, resulting in $\beta_y^n := \overline{\delta y}/h_y$ and $\beta_x^n := -2$ if $\beta_y^n \in [0, 1/2)$, or $\beta_x^n := 2$ if $\beta_y^n \in [1/2, 1]$, Figure 3c.

Once all the discrete EBs in the domain are aligned with the relevant grid direction and are appropriately positioned inside a cell, we achieve the desired sub-cell resolution of the immersed body boundary. In the latter respect, the present method essentially differs from the stair-case approach wherein the boundary is projected on cell faces [1].

2.3. Merging of EBs

Occasionally, it might arise a scenario with two successive cells, along a column or row of cells, each having orthogonalized EBs in the same direction (see Figure 4). Technically, with the outlined procedure, this is a natural outcome and it can be accommodated. However, this situation may be non-physical. It might be a *'numerical spray'* and, as such, it perturbs an evolving solution, not to mention the algorithmic (flux-computation and time-stepping) complications it creates during implementation.

In the event of two such distinct, orthogonalized EBs, we can effectively get rid of the *'numerical spray'* by properly merging these EBs, obtaining a single equivalent EB in one of the cells. The *'numerical spray'* is superseded by reuniting it with its *'parent material'* that originally gave it off.



(a) $\beta_y(i,j) := \beta_y(i,j) + \beta_y(i,j+1)$

(b) $\beta_x(i+1,j) := \beta_x(i,j) + \beta_x(i+1,j) - 1$

Figure 4: Merging two EBs and determination of an equivalent EB.

Therefore, the merging is done in the direction of the *'parent material'* and such that the conservation law is satisfied. The procedure is illustrated in Figure 4, and is concisely described, only in one grid direction, in Table 1. Note, it is presumed, that the body or *'parent material'*, represented by the shaded areas, extends in the negative j- and i-direction, in Figures 4a and 4b, respectively. The same presumption holds for the algorithm in Table 1.

3. Finite-volume discretization

In this section, we briefly formulate the governing (partial differential) equation and give a concise description of the spatial (finite-volume) and temporal discretization schemes. Only settings and the relevant formulae, for the latter, are described. For detailed exposition, readers are referred to [3].

Table 1: Algorithm for merging EBs

```
1. iff $\exists\{\beta_x^n(i,j),\ \beta_x^n(i+1,j)\}\in[0,1],\ i\in[1,N_x-1],\ j\in[1,N_y]$,
    i) if $\beta_x^n(i,j)+\beta_x^n(i+1,j)<1$,
       then $\beta_x^n(i,j):=\beta_x^n(i,j)+\beta_x^n(i+1,j)$, and $\beta_x^n(i+1,j):=-2$;
   ii) else
       then $\beta_x^n(i,j):=2$, and $\beta_x^n(i+1,j):=\beta_x^n(i,j)+\beta_x^n(i+1,j)-1$.
  iii) endif
2. endiff
```

A multi-dimensional convection problem, in conservation form, with the associated initial condition, can be written as :

$$\frac{\partial c}{\partial t}+\nabla\cdot\mathbf{F}=0, \qquad \text{in } \mathcal{D}\in\mathbb{R}^2\times(0,\mathcal{T}], \tag{2a}$$

$$c(\mathbf{x},t\!=\!0)=c_0(\mathbf{x}), \qquad \text{in } \mathcal{D}\in\mathbb{R}^2, \tag{2b}$$

where $c(\mathbf{x},t)$ is the scalar field that is convected by the flow field $\mathbf{u}=\Big(u\left(\mathbf{x}\right),v\left(\mathbf{x}\right)\Big)^{\mathrm{T}}$. The components of the flux vector $\mathbf{F}=\Big(f\left(u,c\right),g\left(v,c\right)\Big)^{\mathrm{T}}$ are defined, at t^n, as: $f^n(u,c):=u(\mathbf{x})\,c(\mathbf{x},t^n)$ and $g^n(v,c):=v(\mathbf{x})\,c(\mathbf{x},t^n)$.

For a cell-averaged discrete solution in $\mathcal{D}_{i,j}$, at t^n, i.e.,

$$c_{i,j}^n\equiv\frac{1}{h_xh_y}\int_{y_{j-1/2}}^{y_{j+1/2}}\int_{x_{i-1/2}}^{x_{i+1/2}}c(\xi,\eta,t^n)\ \mathrm{d}\xi\ \mathrm{d}\eta, \tag{3a}$$

the convection equation (2a) can be rewritten, in the integral form, as:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\mathcal{D}_{i,j}}c_{i,j}(t)\ \mathrm{d}\mathbf{x}+\int\limits_{\partial\mathcal{D}_{i,j}}\mathbf{F}\!\cdot\!\mathbf{n}\ \mathrm{d}S=0, \tag{3b}$$

where $\mathbf{n}$ is the unit outward-pointing normal to $\partial\mathcal{D}_{i,j}$. Assuming fluxes to be constant along cell faces, i.e.,

$$f_{i+1/2,j} = f_{i+1/2,j}(t) \equiv \frac{1}{h_y} \int_{y_{j-1/2}}^{y_{j+1/2}} f(x_{i+1/2}, \eta, t)\, \mathrm{d}\eta\,,\ \ldots\,, \tag{3c}$$

we have the semi-discrete equation:

$$h_x h_y \frac{\mathrm{d}c_{i,j}}{\mathrm{d}t} \ +\ \Big(f_{i+1/2,j}(t) - f_{i-1/2,j}(t)\Big) + \Big(g_{i,j+1/2}(t) - g_{i,j-1/2}(t)\Big) = 0. \tag{3d}$$

3.1. Flux computation

Let us consider an orthogonalized EB situated in cell $\mathcal{D}_{i,j}$, i.e., with either $\beta_x^n(i,j) \in [0,1]$ or $\beta_y^n(i,j) \in [0,1]$, with its associated (Dirichlet) boundary conditions $c^{\mathrm{EB_l}}(i,j)$ and $c^{\mathrm{EB_r}}(i,j)$, which are solution values appended to the left- and right-side of the EB, respectively. Based on a three-point upwind-biased state-interpolation, only three neighbouring fluxes feel its presence.

Deferring the detailed derivation to [3], these EB-affected fluxes accommodate the corresponding boundary conditions. Omitting the time indices and denoting $\beta_x(i,j)$ by β, for convenience, the final (EB-affected) flux formulae, only on vertical cell faces for $u_{l,j} > 0$, $l = \{i - \frac{1}{2}, i + \frac{1}{2}, i + \frac{3}{2}\}$, can be described as:

$$\begin{aligned}
\tilde{f}_{i-\frac{1}{2},j} &:= u_{i-\frac{1}{2},j}\Big(c_{i-1,j} + \tfrac{8}{(3+6\beta)(3+2\beta)}\left(c_{i,j}^{\mathrm{EB_l}} - c_{i-1,j}\right) + \tfrac{1+6\beta}{18+12\beta}\left(c_{i-1,j} - c_{i-2,j}\right)\Big),\\
\tilde{f}_{i+\frac{1}{2},j} &:= u_{i+\frac{1}{2},j}\Big(c_{i,j}^{\mathrm{EB_r}} + \tfrac{2-2\beta}{3-2\beta}\left(c_{i+1,j} - c_{i,j}^{\mathrm{EB_r}}\right)\Big),\\
\tilde{f}_{i+\frac{3}{2},j} &:= u_{i+\frac{3}{2},j}\Big(c_{i+1,j} + \tfrac{11-6\beta}{30-12\beta}\left(c_{i+2,j} - c_{i+1,j}\right) + \tfrac{4}{(9-6\beta)(5-2\beta)}\left(c_{i+1,j} - c_{i,j}^{\mathrm{EB_r}}\right)\Big).
\end{aligned} \tag{4}$$

In [3], relevant EB-sensitive limiters have also been derived for the corresponding fluxes.

Elsewhere, where the EBs have 'no' effect, the standard κ-scheme, $\kappa \in [-1,1]$, is used, see [4] for details. A typical flux formula for $\partial\mathcal{D}_{i,j+1/2}$, for

$\kappa = 1/3$ and $v_{i,j+1/2} > 0$, can be written as:

$$g_{i,j+1/2} := v_{i,j+1/2}\Big(c_{i,j} \ + \ (c_{i,j+1} - c_{i,j})\,/3 \ + \ (c_{i,j} - c_{i,j-1})\,/6\Big). \tag{5a}$$

Correspondingly, we also use the standard limiter:

$$\varphi(r_{i,j+1/2}) = \max(0, \min(2r_{i,j+1/2}, \min(1/3 + 2/3\, r_{i,j+1/2}, 2))), \tag{5b}$$

where $r_{i,j+1/2} := (c_{i,j+1} - c_{i,j} + \varepsilon)\,/\,(c_{i,j} - c_{i,j-1} + \varepsilon)$ is the local solution-gradient, for the relevant standard fluxes, with $\varepsilon \ll 1$, some small number, to avoid undefined ratio of consecutive solution gradients in uniformly constant solution regions. See [5] for details. Notice that the time indices are again omitted, for brevity.

3.2. Time integration

We are mainly interested in the spatial discretization, and with a very small Courant number $\nu \ll 1$, we compute a temporally accurate solution using the forward Euler scheme:

$$c^{n+1}_{i,j} = c^n_{i,j} - \frac{\tau}{h_x h_y}\Big(f^n_{i+\frac{1}{2},j} - f^n_{i-\frac{1}{2},j}\Big) - \frac{\tau}{h_x h_y}\Big(g^n_{i,j+\frac{1}{2}} - g^n_{i,j-\frac{1}{2}}\Big). \tag{6a}$$

We choose τ based on the worst case, i.e., the maximum ν corresponding to the combined effect of a large flow velocity and small cell size, and it is uniformly taken to be:

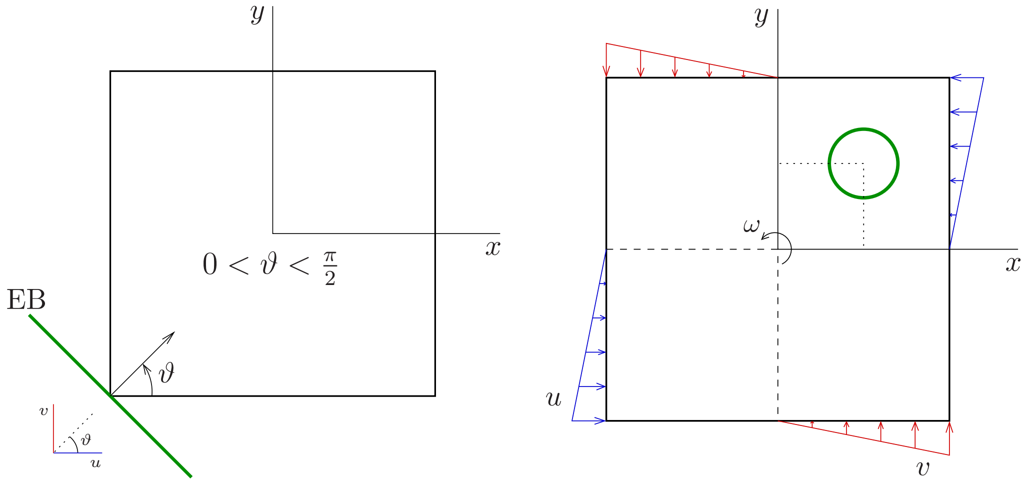
$$\tau = \min\left\{ \nu h_x / \max_{\substack{1\le i\le N_x+1\\ 1\le j\le N_y}} \left|u_{i-1/2,j}\right|,\ \nu h_y / \max_{\substack{1\le i\le N_x\\ 1\le j\le N_y+1}} \left|v_{i,j-1/2}\right| \right\}, \tag{6b}$$

which ensures the CFL condition $\nu \le 1$ everywhere.

Note that some fluxes are discontinuous in $t \in [t^n, t^{n+1}]$ because of discrete EBs crossing cell faces, and stability of the evolving solution (6a) is not guaranteed. For such cases, we have devised a locally adapted time-stepping scheme, see [3].

4. Numerical examples

To validate the algorithm presented in this paper, we consider two test cases, a translating rectilinear discontinuity and a revolving cylindrical discontinuity, with prescribed two-dimensional flow fields inside a rectangular domain. Settings of the problems are depicted in Figure 5. Here, we take $\ell_x = \ell_y = 2$, with $\mathcal{D} = [-1, 1] \times [-1, 1]$.



(a) translating rectilinear discontinuity (b) revolving cylindrical discontinuity

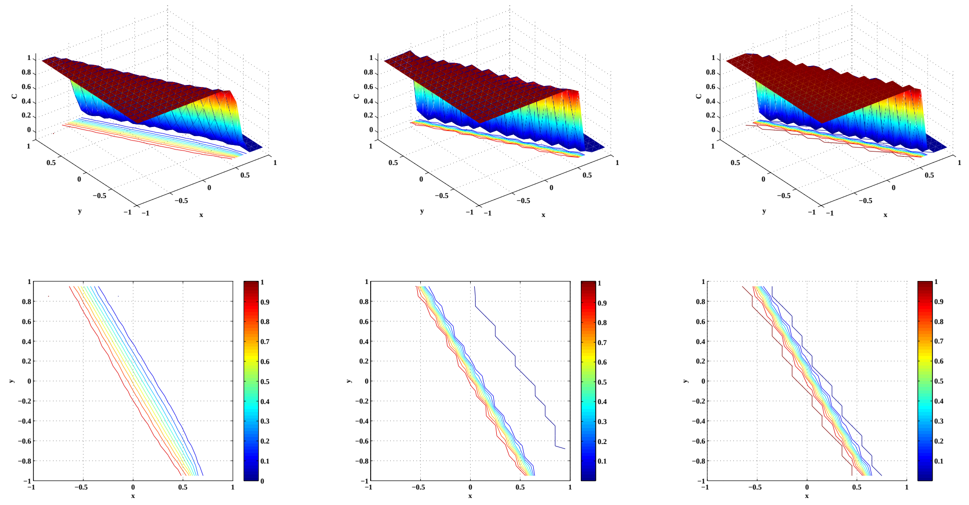
Figure 5: Domain, flow fields and problem settings.

4.1. A translating rectilinear discontinuity

Consider a rectilinear discontinuity of arbitrary orientation $\vartheta \in [0, \pi/2]$, initially situated at the bottom-left corner of the domain, and moving in a uniform two-dimensional flow field with velocity $\mathbf{u} = (\cos\vartheta,\ \sin\vartheta)^{\mathrm{T}}$. The flow field, as shown in Figure 5a, is normal to the discontinuity.

The discontinuity, which goes with the flow, is assumed to model a rigid, infinitely thin plate that separates two quantities of different values, i.e., $c = 1$ (at the upstream side) and $c = 0$ (at the downstream side). These solution values are taken as *embedded boundary conditions*, i.e., $c^{\mathrm{EB_l}} = 1$ and $c^{\mathrm{EB_r}} = 0$, to be used in the relevant fixed-grid fluxes in the immediate neighbourhood (see Eq. (4)).

Figure 6 shows results for the translating rectilinear discontinuity, of typical orientation $\vartheta = \pi/6$, on a grid $(N_x, N_y) = (20, 20)$ at final time $\mathcal{T} = \sqrt{2}$. The results obtained with the current method appear to be significantly more accurate than those obtained with the standard method.



(a) Standard solution

(b) EB solution

(c) Exact discrete solution

Figure 6: Results for a translating rectilinear discontinuity, of orientation $\vartheta = \pi/6$, obtained with limited fluxes and $\nu \ll 1$.

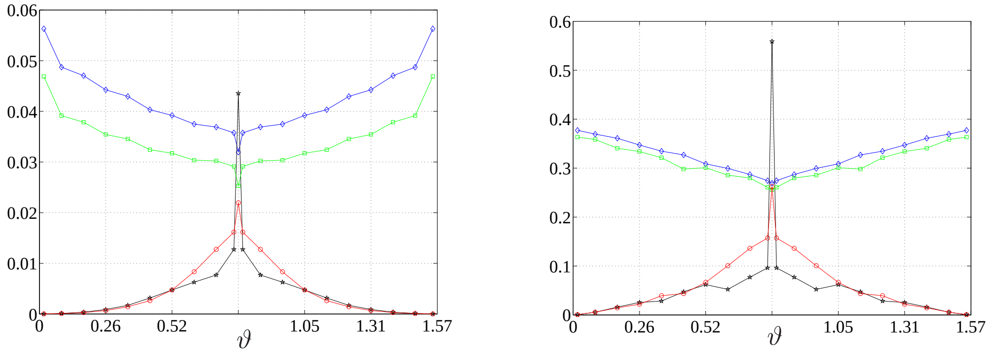
Figure 7 shows the global errors in L_1 and L_∞ norms for various orientations $\vartheta \in [0, \pi/2]$. We can see that, for the current EB method, the errors, in both norms, are significantly lower for all ϑ, but $\vartheta = \pi/4$, than those of the standard method. The errors of the standard method show a minimum for $\vartheta = \pi/4$. This experimentally found error behaviour can be explained analytically. This corroboration will be presented in a future paper.

Interestingly, the errors of the current EB method increase towards $\vartheta = \pi/4$. This is because of the orthogonalization of the EB. Apparently, the orthogonalization is least justified for $\vartheta = \pi/4$. In this particular case, even though the EB is then equally inclined to both grid directions, we arbitrarily project it on to one of them, which makes the EB to be seen in only one direction. Nonetheless, regardless of this 'crude' approximation, the errors of the current EB method for $\vartheta = \pi/4$ are still bounded and not worse when compared to those of the standard method.

4.2. A revolving cylindrical discontinuity

Consider a cylindrical discontinuity of radius $R = 0.2$, and unit height, initially located at $(x, y) = (1/2, 1/2)$, as shown in Figure 5b, which revolves with a circular flow-field with velocity $\mathbf{u} = (-\omega y, \omega x)^{\mathrm{T}}$, where $\omega = 2\pi$ is the angular velocity (a solid-body rotation).

Similarly, as in § 4.1, the discontinuity, which goes with the flow, is assumed to model a rigid, infinitely thin-walled cylinder that separates two quantities of different values, i.e., $c = 1$ and $c = 0$, inside and outside the cylinder, respectively. The solution values $c = 1$ and $c = 0$ are appended to the boundary of the immersed cylinder; $c^{\mathrm{EB_l}} = 1$ at the inner, and $c^{\mathrm{EB_r}} = 0$ at the outer side of the cylinder's wall.



(a) Error in L_1 norm

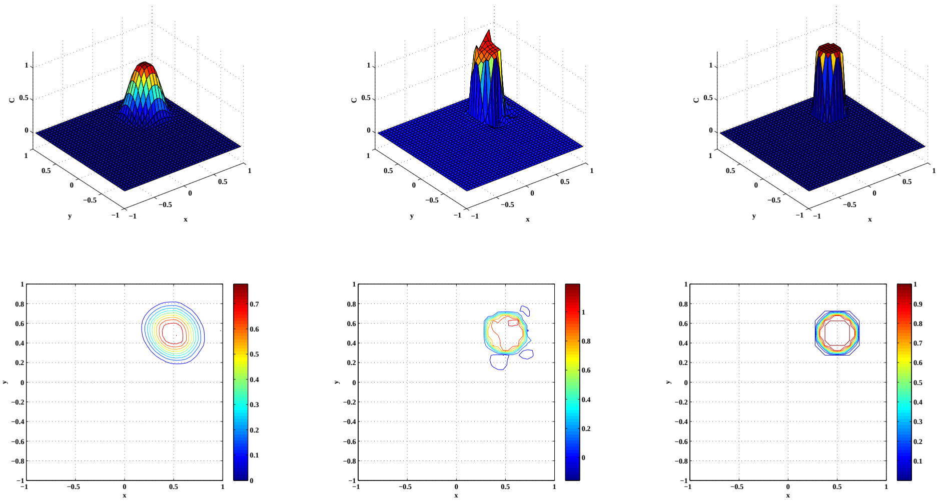
(b) Error in L_∞ norm

Figure 7: Global errors for various orientations, $0 < \vartheta < \pi/2$, of the translating rectilinear-discontinuity, obtained on a 20×20 grid, at $\mathcal{T} = \sqrt{2}$ and with $\nu \ll 1$. Blue: standard scheme; Black: current EB scheme; Green: standard scheme with limited flux; Red: EB scheme with limited flux.

Figure 8 shows the results for the revolving cylindrical discontinuity obtained on a grid of $(N_x, N_y) = (40, 40)$, at a final time $\mathcal{T} = 1$ (i.e., after one full revolution). Clearly, the results of the current EB method, on this grid, have higher resolution than those of the standard method, but they are not yet monotone.

5. Conclusion

In this work, a new immersed-boundary method, which effectively embeds boundary conditions, valid on a moving body, only in certain fluxes in the immediate neighbourhood, has been introduced. The algorithm has been tested with two problems and the results obtained are promising. They have higher resolution compared to those obtained by standard methods, but are



(a) Standard solution (b) EB solution (c) Exact discrete solution

Figure 8: Results for a revolving cylindrical discontinuity after one full revolution on a 40×40-grid, obtained with limited fluxes and $\nu \ll 1$.

not yet entirely monotone. It is a simple and elegant algorithm, and we anticipate to use it for 2D Euler flows, which we foresee to consider next.

References

[1] Mittal R, Iaccarino G. Immersed boundary methods. Ann Rev Fl Mech 2005;37:239–61.

[2] Hassen Y, Koren B. A finite-volume method for convection problems with embedded moving-boundaries. In: Choi H, Choi HG, Yoo JY, editors. Computational Fluid Dynamics 2008. Heidelberg: Springer; 2009, p. 499–505.

[3] Hassen Y, Koren B. Finite-volume discretizations & immersed bound-

aries. In: Koren B, Vuik C, editors. Advanced Computational Methods in Science and Engineering. Lect. Notes Comput. Sci. Eng. 71; Heidelberg: Springer; 2010, p. 229–68.

[4] van Leer B. Upwind-difference methods for aerodynamic problems governed by the Euler equations. In: Engquist BE, Osher S, Somerville RCJ, editors. Large-Scale Computations in Fluid Mechanics. Lect. Appl. Math. 22.2; Providence, RI: Am. Math. Soc.; 1985, p. 327–36.

[5] Koren B. A robust upwind finite-volume method for advection, diffusion and source terms. In: Vreugdenhil CB, Koren B, editors. Numerical Methods for Advection-Diffusion Problems. Notes on Num. Fl. Mech. 45; Vieweg, Braunschweig; 1993, p. 117–38.

Centrum Wiskunde & Informatica (CWI) is the national research institute for mathematics and computer science in the Netherlands. The institute's strategy is to concentrate research on four broad, societally relevant themes: earth and life sciences, the data explosion, societal logistics and software as service.

Centrum Wiskunde & Informatica (CWI) is het nationale onderzoeksinstituut op het gebied van wiskunde en informatica. De strategie van het instituut concentreert zich op vier maatschappelijk relevante onderzoeksthema's: aard- en levenswetenschappen, de data-explosie, maatschappelijke logistiek en software als service.

Bezoekadres:
Science Park 123
Amsterdam

Postadres:
Postbus 94079, 1090 GB Amsterdam
Telefoon 020 592 93 33
Fax 020 592 41 99
info@cwi.nl
www.cwi.nl

CWI
Centrum Wiskunde & Informatica