

# Acceleration of Contour Dynamics Simulations with a Hierarchical-Element Method

P. W. C. Vosbeek,<sup>\*,†,1</sup> H. J. H. Clercx,<sup>†</sup> and R. M. M. Mattheij<sup>\*</sup>

<sup>\*</sup>*Scientific Computing Group, Department of Mathematics and Computing Science, †Fluid Dynamics Laboratory, Department of Applied Physics, Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands*

E-mail: vosbeek@knmi.nl

Received July 7, 1999; revised March 8, 2000

---

This paper presents a so-called *hierarchical-element method* that can be used to accelerate complex contour dynamics simulations. The method is based on a modified fast multipole method where the multipole approximations are replaced by Poisson integrals. In this paper, attention is being paid to the theoretical derivation of the method. Furthermore, numerical and implementation aspects are considered. Various numerical simulations show that the speed-up of the method is significant, while the accuracy of the results is not being influenced. © 2000 Academic Press

*Key Words:* contour dynamics; fast multipole method; Euler equations; fast summation; Poisson equation.

---

## 1. INTRODUCTION

In this paper a method to accelerate *contour dynamics* simulations is discussed. Contour dynamics is a powerful method for simulating vortices in two-dimensional flows of an incompressible, inviscid fluid. The method and many improvements thereof have been brought to full growth by the pioneering work of Dritschel [7, 8].

Contour dynamics is based on the observation that the evolution of a patch of uniform vorticity is fully determined by the evolution of its boundary contour. The method is not limited to just one region of uniform vorticity; indeed, several contours can be nested in order to obtain an approximation of a patch of distributed vorticity [7, 8, 23].

In contour dynamics simulations, the contours are approximated by a finite but adjustable number of nodes. The velocity field at a certain point in space depends on the position of each node. To determine the evolution of the contours, the velocity field is computed at

<sup>1</sup> Present address: Royal Netherlands Meteorological Institute, P.O. Box 201, 3730 AE De Bilt, The Netherlands.

every node on the contours thus requiring order  $\mathcal{O}(N^2)$  operations where  $N$  is the total number of nodes.

The initial number of nodes on a contour is usually not large enough to approximate the contour sufficiently accurate during a whole simulation. Therefore nodes are added to a contour where necessary and they are removed from regions where redundant. However, in many situations the total number of nodes increases rapidly during a computation and as a result, the calculations become computationally very expensive due to the  $\mathcal{O}(N^2)$  complexity of the algorithm.

One can imagine that in certain complex situations where high numbers of nodes are required, the simulations become computationally too expensive to carry them out within a reasonable amount of time. In order to be able to perform such computationally expensive calculations, it is necessary to accelerate the method. In the past, methods have been developed to accelerate contour dynamics already, for example, moment accelerated contour surgery by Dritschel [9, 10] and a fast adaptive vortex method by Buttké [5]. In both methods, contour dynamics is combined with a fast multipole technique [11], however, in very different ways. In the first method, which was developed for many vortex simulations, vortices far away from the evaluation point are replaced by multipole expansions in order to reduce the computation time of the velocity calculations. In the second method [5], a similar effect is achieved by calculating the velocity at a given point by a fast summation of the contributions of vortex elements inside the patch. Vortex elements of different sizes are used to approximate the patch accurately.

The method presented in this paper combines a modified fast multipole method, a so-called *hierarchical-element method*, with contour dynamics. This hierarchical-element method is based on the one developed by Anderson [1] which actually is a fast multipole method with the multipole expansions replaced by Poisson integrals. The advantage of this approach is that the method can be applied to a large variety of flow problems like, for example, many vortex calculations and complex flow problems of only few vortices. Also more geophysically relevant flow problems like, for example, the evolution of vortices in the presence of non-uniform background vorticity (where contours are also necessary outside the vortices [20]) can be studied using this method. An additional advantage is that the method is fairly easy to generalise to three dimensions as discussed by Anderson [1].

The remainder of the paper is organised as follows. In the next section, the contour dynamics method as used throughout this paper is discussed briefly. Section 3 gives a small review of the hierarchical-element methods and in particular the approach by Anderson [1]. Subsequently, Sections 4 and 5 discuss the necessary adaptations to Anderson's method that make this particular hierarchical-element method suitable for application to contour dynamics simulations. In Section 6, the accuracy and computational efficiency of the new method is illustrated by some numerical examples. Finally, in Section 7, some conclusions and recommendations are given.

## 2. CONTOUR DYNAMICS

The equations of motion for a flow of an incompressible, inviscid fluid, are given by conservation of mass, i.e.,

$$(\nabla, \mathbf{u}) = 0, \tag{1}$$

where  $\mathbf{u}$  is the velocity vector, and the Euler equation, i.e.,

$$\frac{D\mathbf{u}}{Dt} = \frac{\partial\mathbf{u}}{\partial t} + (\mathbf{u}, \nabla)\mathbf{u} = -\frac{1}{\rho}\nabla p, \quad (2)$$

with  $p$  the pressure and  $\rho$  the density.

For a 2D flow, a stream function  $\psi$  can be introduced because of (1),

$$\begin{cases} u = \frac{\partial\psi}{\partial y}, \\ v = -\frac{\partial\psi}{\partial x}. \end{cases} \quad (3)$$

The vorticity vector  $\boldsymbol{\omega}$ , which is defined as  $\boldsymbol{\omega} := \nabla \times \mathbf{u}$ , only has a vertical component in this case, i.e.,  $\boldsymbol{\omega} = \omega \mathbf{e}_z$ . By taking the curl of the Euler equation (2), the equations of motion can be written in terms of the stream function ( $\psi$ ) and the vorticity ( $\omega$ ) and take the form

$$\frac{D\omega}{Dt} = \frac{\partial\omega}{\partial t} + (\mathbf{u}, \nabla)\omega = 0, \quad (4)$$

$$\nabla^2\psi = -\omega. \quad (5)$$

The first equation expresses conservation of vorticity of a fluid particle. The solution of the second, the Poisson equation, in an infinite domain is formally given by

$$\psi(\mathbf{x}, t) = -\iint_{\mathbb{R}^2} \omega(\mathbf{x}', t) G(\mathbf{x}; \mathbf{x}') dx' dy', \quad (6)$$

where  $G(\mathbf{x}; \mathbf{x}') = \frac{1}{2\pi} \ln \|\mathbf{x} - \mathbf{x}'\|$ , i.e., Green's function of the Laplace operator for an infinite domain, and  $\mathbf{x} = (x, y)$ . The norm  $\|\cdot\|$  is defined by  $\|\mathbf{x}\| := \sqrt{x^2 + y^2}$ , for each  $\mathbf{x} \in \mathbb{R}^2$ .

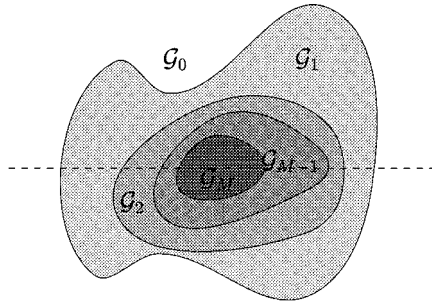
For contour dynamics, an initially continuous distribution of vorticity  $\omega(\mathbf{x}, 0)$  is replaced by a piecewise uniform distribution  $\tilde{\omega}(\mathbf{x}, 0)$  given by

$$\tilde{\omega}(\mathbf{x}, 0) = \sum_{l=0}^m \omega_l, \quad \mathbf{x} \in \mathcal{G}_m(0) \setminus \mathcal{G}_{m+1}(0), \quad m = 0, \dots, M, \quad (7)$$

where the regions  $\mathcal{G}_m(0)$  are nested,  $\mathcal{G}_{m+1}(0) \subset \mathcal{G}_m(0)$ ,  $\mathcal{G}_0(0) = \mathbb{R}^2$ , and  $\mathcal{G}_{M+1}(0) = \emptyset$ , i.e.,  $\mathcal{G}_{M+1}(0)$  is empty. Unless specified otherwise,  $\omega_0$  is considered to be zero. The  $\omega_{m+1}$ ,  $m = 0, \dots, M-1$ , can be thought of as the jump in vorticity when moving from region  $\mathcal{G}_m(0) \setminus \mathcal{G}_{m+1}(0)$  to  $\mathcal{G}_{m+1}(0) \setminus \mathcal{G}_{m+2}(0)$ , with  $\mathcal{G}_m(0) \setminus \mathcal{G}_{m+1}(0)$  the region  $\mathcal{G}_m(0)$  without  $\mathcal{G}_{m+1}(0)$ . Figure 1 shows an example of regions of uniform vorticity  $\mathcal{G}_m$ ; Fig. 2 shows the corresponding piecewise-uniform distribution of vorticity.

Conservation of vorticity (4) now ensures that the piecewise-uniform distribution remains piecewise uniform throughout time. Furthermore, it can be derived that the velocity field  $\mathbf{u}(\mathbf{x}, t)$ , anywhere in the flow, and in particular on the contours  $\mathcal{C}_m$  where  $\tilde{\omega}(\mathbf{x}, t)$  is discontinuous, can be determined by the computation of contour integrals [7, 8, 20, 22, 23],

$$\mathbf{u}(\mathbf{x}, t) = -\sum_{m=1}^M \omega_m \oint_{\mathcal{C}_m(t)} G(\mathbf{x}; \mathbf{x}') dx'. \quad (8)$$



**FIG. 1.** An arbitrary patch of piecewise-uniform vorticity distribution. The regions  $\mathcal{G}_m$  are nested, i.e.,  $\mathcal{G}_{m+1} \subset \mathcal{G}_m$  for  $m = 0, \dots, M - 1$ .

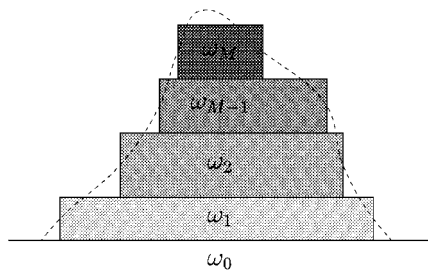
The contour integrals in (8) have to be computed numerically and the contours therefore have to be approximated by a finite, but adjustable, number of nodes. Between two subsequent nodes on a contour, linear interpolation is used to determine the contour integrals in (8). The addition and removal of nodes is based on the local curvature of the contours, minimum and maximum distance between two successive nodes, and quasi-uniformity of the distribution of the nodes [20, 22].

The evolution of the contours can be found by integrating the velocities, determined at the nodes on the contours, over a small time step. The time integration is carried out using the second-order (symplectic) midpoint rule [19]. The reason for choosing this scheme is that it conserves quantities like the area and circulation of the regions of uniform vorticity better than ordinary integration methods [20, 22].

### 3. HIERARCHICAL-ELEMENT METHODS

In this section a hierarchical-element method (HEM) as described by Anderson [1] is discussed. The method is based on the fast multipole technique developed by Greengard and Rokhlin [11] but does not employ multipoles themselves. Instead, approximations based on Poisson's formula are used.

The fast multipole method (FMM) itself has been developed in order to accelerate computations of  $N$ -body interactions. For example, given  $N$  charged particles at positions  $\mathbf{x}_n$  with strength  $\kappa_n$ ,  $n = 1, \dots, N$ , the potential  $\Phi$  at every particle has to be calculated. Here,



**FIG. 2.** A cross-section (along the dashed line in Fig. 1) of the piecewise-uniform vorticity profile approximating the continuous profile (dashed line).

$\Phi$  is a solution of

$$\nabla^2 \Phi = \sum_{n=1}^N \kappa_n \delta(\mathbf{x} - \mathbf{x}_n),$$

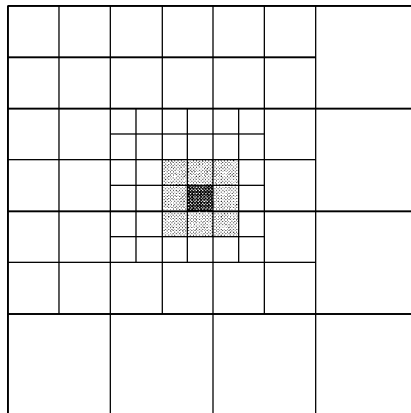
where  $\delta(\mathbf{x})$  is Dirac's delta function. The solution is given by

$$\Phi(\mathbf{x}) = \sum_{n=1}^N \frac{\kappa_n}{2\pi} \ln \|\mathbf{x} - \mathbf{x}_n\|. \quad (9)$$

Clearly, the evaluation of  $\Phi$  at every particle requires  $\mathcal{O}(N^2)$  operations. The FMM reduces the operation count to  $\mathcal{O}(N \ln(N))$  or even  $\mathcal{O}(N)$ . Note that there is a strong resemblance with point vortices: the charged particles can be replaced by point vortices and in that case,  $\Phi$  should be replaced by the stream function  $\psi$ . The FMM is thus very suitable for accelerating (many) point vortex interactions as well.

The FMM basically consists of two parts. The first part is based on the concept of combining a large number of particles into a single computational element. When a cluster of particles is far away from a certain point at which the potential has to be calculated, the potential of the cluster is approximated by the potential induced by a single computational element inside the cluster. To this end a multipole expansion [11] around the centre  $z_0$  of a disk containing the cluster of particles is used.

The second part of the FMM concerns the organisation of the computations in such a way that the technique of combining particles is efficient and does not lead to inaccuracies. For example, when combining particles into single elements, the more widely distributed in space the particles of a given cluster are, the more inaccurate the multipole expansion becomes for a fixed order  $K$  of the multipole and a fixed point  $z$  of evaluation. However, if the evaluation point  $z$  is moved away from the centre of the disk  $z_0$ , then the accuracy of the approximation improves. So, if a certain degree of accuracy is desired, the potential should be approximated by a hierarchy of multipole expansions [1]. Far away from the evaluation point, particles are combined over large regions; particles closer to the evaluation point are combined over smaller regions as indicated in Fig. 3. This figure shows an example of a



**FIG. 3.** A hierarchical clustering of particles which is used to create a multipole approximation to the potential at a point in the dark grey box (adapted from [1]).

hierarchical clustering of particles which is used to create a multipole approximation to the potential at a point in the dark grey box. The potential induced by particles in the white boxes is combined into multipole approximations; the potential induced by particles in the light grey and dark grey boxes is computed using the direct interaction formula.

In the HEM described by Anderson, instead of multipole expansions, Poisson's formula is used. According to Poisson's integral the potential outside a disk with radius  $a$ , containing the particles  $\mathbf{x}_n$  for  $n = n_1, \dots, n_2$ , is given by

$$\Phi(r, \varphi) = \kappa \ln(r) + \frac{1}{2\pi} \int_0^{2\pi} (\Phi(a, \vartheta) - \kappa \ln(a)) g_0\left(\frac{a}{r}, \varphi, \vartheta\right) d\vartheta, \quad \text{for } r > a, \quad (10)$$

where

$$g_0(\varrho, \varphi, \vartheta) := \frac{1 - \varrho^2}{1 - 2\varrho \cos(\varphi - \vartheta) + \varrho^2}, \quad (11)$$

and

$$\kappa := \frac{1}{2\pi} \sum_{n=n_1}^{n_2} \kappa_n.$$

The coordinates  $(r, \varphi)$  indicate the position in polar coordinates of the evaluation point with respect to the centre of the disk [15].

An advantage of this approach is that also collections of sources which are more general than point charges or point vortices, such as given areas of certain charge distributions or vorticity distributions, can be treated. The application of multipole expansions without using the Poisson's integral approach might be very difficult or even impossible for these particular problems.

The integral in (10) can be determined numerically. Problems arise when integrating it straightforwardly by the trapezoidal rule; however, by modifying the kernel  $g$  of the Poisson integral appropriately, the numerical integration appears to be super-convergent (see Subsection 4.2 and the paper by Anderson [1]).

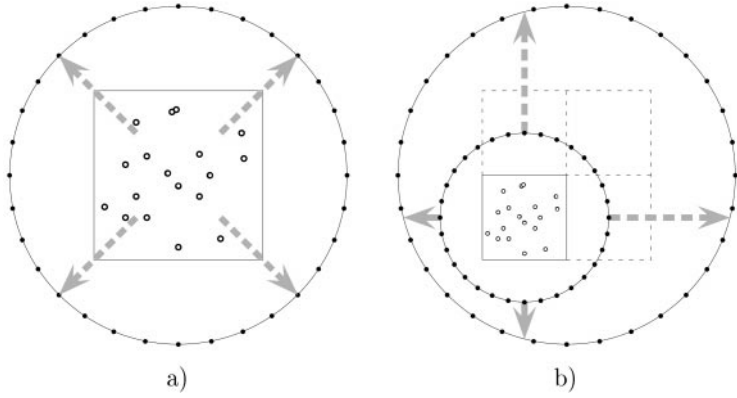
The numerical approximation of the Poisson integral (10) is referred to by Anderson as an *outer-ring approximation*. In a similar way a so-called *inner-ring approximation* can be defined, which is the numerical approximation of

$$\Phi(r, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} \Phi(a, \vartheta) g_0\left(\frac{r}{a}, \varphi, \vartheta\right) d\vartheta, \quad \text{for } r < a. \quad (12)$$

This inner-ring approximation represents the potential inside a ring with radius  $a$ .

In the following, the evaluation of the potential at the integration points of the outer-ring by means of the outer-ring approximation or by direct summation of the appropriate terms in (9) is referred to as the *construction of the outer-ring* and likewise for the inner-ring.

Now, following Anderson, the method proceeds as follows. First, a square domain is chosen which encloses all particles. Furthermore, a finest level of refinement  $l_f$  is chosen (a way to do this is discussed later). At this finest level  $l_f$ , the domain is divided into



**FIG. 4.** Construction of an outer-ring at the finest level using the direct interaction formula (a) and at coarser levels from a “child” box at the previous finer level (b). Particles are denoted with small circles; integration points on the ring with black dots. The boxes indicated with dashed lines in (b) contribute in a similar way as the box drawn with a solid line (adapted from [1]).

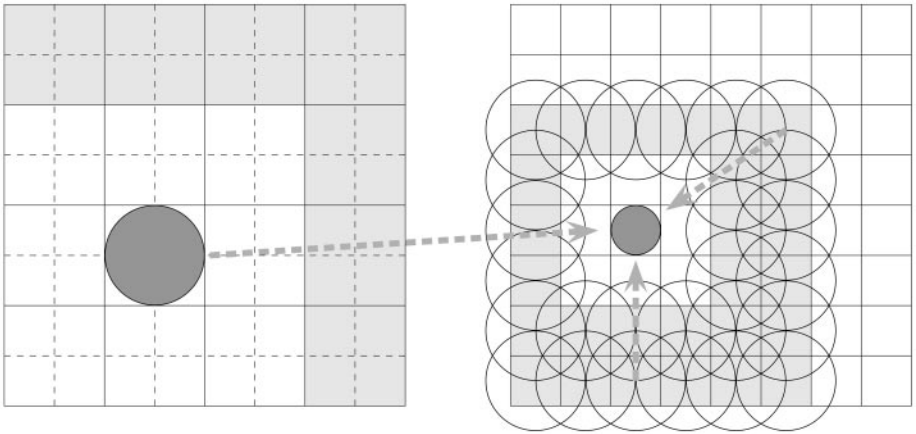
$2^{l_f} \times 2^{l_f}$  square boxes.<sup>2</sup> Similarly, at the coarser levels the domain contains  $2^l \times 2^l$  boxes,  $l = 2, \dots, l_f$ . Like the FMM, this method also consists of two parts.

In the first part, outer-rings are constructed at each level, starting with the finest level. Here, a ring of radius equal to the box width is chosen around each box. The centre of the ring is located at the centre of the box. Then, at the outer-ring, the potential due to the particles inside the box is determined (see also Fig. 4a) by means of direct summation of the appropriate terms in (9), i.e., only the terms concerning the particles inside the box are taken into account. After finishing the finest level, one proceeds to the coarser level where the outer-rings (again with radius equal to the box size and with centre located at the centre of the box) are constructed from the finer level by combining the contributions of the four “child” boxes inside the coarser box by means of the outer-ring approximation. In Fig. 4b it is shown how the contribution to the outer-ring of a “child” box (solid lines) is determined; the other three boxes (dashed lines) contribute similarly. This procedure can be repeated at each coarser level. At the end of the first part, outer-rings have been constructed for each box at every level.

In the second part of the algorithm, the contributions of the outer-rings are organised in a rather smart way. To this end, the concept of being *well separated* [1, 11] is used. If the boxes at a certain level  $l$  are identified by a pair  $(i, j)$ ,  $i, j = 1, \dots, 2^l$ , with box  $(1, 1)$  being the box at the bottom left and box  $(2^l, 2^l)$  the box at the top right of the domain, a box  $(i_1, j_1)$  is *well separated* from box  $(i_2, j_2)$  with distance  $D$ , if the maximum of the difference between their indices  $\max(|i_1 - i_2|, |j_1 - j_2|)$  is larger than  $D$ . At a certain level, this second part consists of constructing inner-rings for each box which are used to represent the contributions to the potential from two sources.

The first source is the inner-ring associated with the parent box at the previous coarser level (see left part of Fig. 5). This inner-ring represents the contributions of the grey boxes in the left part of Fig. 5. The second source is the contribution from all the outer-rings of the boxes

<sup>2</sup> The method thus requires at least  $2^{l_f} \times 2^{l_f}$  operations and memory storage. This makes the method more difficult to use for large values of  $l_f$ . However, currently parallelisation of the algorithm is being worked on which will take care of this problem.



**FIG. 5.** Two sources contribute to the inner-rings: the inner-rings of the parent box at the previous coarser level (left) and outer-rings of boxes which are well separated from the current box at the current level but the parents of those boxes are not well separated from the parent box of the current box (right) (adapted from [1]).

that are well separated from the given box (with distance  $D = 1$ ) and are contained within boxes at the previous coarser level that are not well separated from the parent box (grey boxes in right part of Fig. 5). The radius of the ring in the inner-ring approximations is taken equal to half the box width. This procedure is carried out for all levels ( $l = 2, l_f$ ). However, at level  $l = 2$  (the coarsest level), only contributions of the second source contribute to the inner-rings of that level, while at the finest level ( $l = l_f$ ), the inner-rings are constructed from the inner-rings of the parents only.

At the end of this part, the potential at any given evaluation point is obtained by computing the potential of the inner-ring approximation associated with the finest level box where the point is residing. This potential is then added to the potential induced by the particles in those neighbouring finest level boxes that are not well separated from the given evaluation point box and which is obtained by using the direct interaction formula. A pseudo-code of the method can be found in the paper by Anderson [1] and in [20].

In the foregoing, the number of refinements  $l_f$  was fixed. However, it should of course be chosen in such a way that the method is the most efficient. It may be clear that the value of  $l_f$  depends on the number of particles in the domain and also on the way they are distributed over the domain. Since in practice the particles are not uniformly distributed in space, Anderson suggests estimating a priori the necessary computation time for several values of  $l_f$  by counting the operations required: execute the hierarchical method but instead of performing all operations required, just update some counters. The counter increments are based on the density of particles in each box and are a measure for the computational time necessary to carry out the specific computational tasks. Based on this procedure, the time, necessary for the work required for different levels of refinement, is estimated, and the level with the least amount of anticipated time is selected. This suggestion appears to work fine in practice.

As mentioned earlier, this variant of the HEM can also be used for problems with given areas of charge distribution or vorticity distribution instead of point charges or point vortices, respectively. This makes the method suitable for accelerating contour dynamics simulations dealing with piecewise-uniform vorticity distributions. In the next two sections, the necessary adaptations of Anderson's method are discussed.



#### 4. THE POISSON INTEGRALS

To accelerate the contour dynamics method with the HEM discussed in the previous section, two aspects related to the implementation of the HEM need to be changed. The first change concerns the Poisson integrals. For contour dynamics, the velocities at the nodes are needed instead of the stream function. Therefore, Poisson integrals as in (10) and (12) have to be derived both for the radial component and the azimuthal component of the velocity. It would also be possible to use the Poisson integrals (10) and (12) to calculate the stream function at the nodes, but then the velocities at the nodes have to be determined numerically, introducing additional errors. Therefore, analytically derived Poisson integrals for the velocity components are to be preferred. The second modification concerns the piecewise-uniform vorticity distribution instead of point vortices, which affects both the Poisson integrals and the construction of the outer-rings at the finest levels.

In Subsection 4.1, the derivation of the Poisson integrals of the outer-rings is discussed, whereas the numerical calculation is discussed in Subsection 4.2. The derivation and numerical calculation of the Poisson integrals of the inner-rings are completely similar to those of the outer-rings and will therefore not be discussed separately here. The construction of outer-rings at the finest level is discussed in Section 5.

*4.1. Theoretical derivation.* Consider a stream function  $\psi$  that satisfies the Laplace equation, i.e.,

$$\nabla^2 \psi = \frac{1}{r} \frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial r^2} + \frac{1}{r^2} \frac{\partial^2 \psi}{\partial \varphi^2} = 0, \quad (13)$$

outside a disk of radius  $a$  containing an area of piecewise-uniform vorticity  $\tilde{\omega}(r, \varphi) = \omega_m$ ,  $(r, \varphi) \in \mathcal{G}_m$ ,  $m = m_1, \dots, m_2$ . The regions  $\mathcal{G}_m$  are nested as in (7). The radial and azimuthal velocity  $u_r$  and  $u_\varphi$  are related to  $\psi$  by

$$u_r(r, \varphi) = \frac{1}{r} \frac{\partial \psi}{\partial \varphi},$$

$$u_\varphi(r, \varphi) = -\frac{\partial \psi}{\partial r}.$$

For the behaviour of the radial and azimuthal velocity at infinity, it can easily be shown that

$$ru_r = \mathcal{O}(1/r), \quad \text{for } r \rightarrow \infty, \quad (14)$$

$$ru_\varphi = \kappa + \mathcal{O}(1/r), \quad \text{for } r \rightarrow \infty, \quad (15)$$

where  $\kappa = \sum_{m=m_1}^{m_2} (\omega_m A_m) / (2\pi)$  with  $A_m$  the area of  $\mathcal{G}_m$ , i.e.,  $\kappa$  is equal to the circulation  $\Gamma$  inside the the disk divided by  $2\pi$ . Furthermore, it can be shown that both  $ru_r$  and  $ru_\varphi$  satisfy the Laplace equation outside the disk, i.e., for  $r > a$ ,  $\nabla^2(ru_r) = \nabla^2(ru_\varphi) = 0$ .

Now consider the following, rather standard, exterior boundary value problem,

$$\begin{cases} \nabla^2 f(r, \varphi) = 0, & r > a, 0 \leq \varphi < 2\pi, \\ f(r, \varphi) = f(a, \varphi), & r = a, 0 \leq \varphi < 2\pi, \\ f(r, 0) = f(r, 2\pi), & r \geq a, \\ f(r, \varphi) = \mathcal{O}(1), & r \rightarrow \infty, 0 \leq \varphi < 2\pi. \end{cases} \quad (16)$$

This problem has a unique solution, given by

$$f(r, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} f(a, \vartheta) g_0\left(\frac{a}{r}, \varphi, \vartheta\right) d\vartheta, \quad \text{for } r > a, \tag{17}$$

with  $g_0$  as defined in (11). The behaviour of  $f$  at infinity follows from rearranging  $g_0$ ,

$$f(r, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} f(a, \vartheta) \left(1 + g\left(\frac{a}{r}, \varphi, \vartheta\right)\right) d\vartheta, \quad \text{for } r > a,$$

with

$$g(\varrho, \varphi, \vartheta) = 2\varrho \frac{\cos(\varphi - \vartheta) - \varrho}{1 - 2\varrho \cos(\varphi - \vartheta) + \varrho^2}, \tag{18}$$

so that

$$f(r, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} f(a, \vartheta) d\vartheta + \mathcal{O}(1/r), \quad \text{for } r \rightarrow \infty.$$

It is clear from this that  $f = \mathcal{O}(1/r)$  at infinity if and only if

$$\int_0^{2\pi} f(a, \vartheta) d\vartheta = 0 \tag{19}$$

and in that case,  $f$  is given by

$$f(r, \varphi) = \frac{1}{2\pi} \int_0^{2\pi} f(a, \vartheta) g\left(\frac{a}{r}, \varphi, \vartheta\right) d\vartheta, \quad \text{for } r > a. \tag{20}$$

Now the Poisson integrals for the radial and azimuthal velocity components easily follow from the above theory by replacing  $f$  by  $ru_r$  and  $ru_\varphi - \kappa$ , respectively, since both  $ru_r$  and  $ru_\varphi - \kappa$  satisfy (19) according to

$$\int_0^{2\pi} au_r(a, \varphi) d\varphi = \int_0^{2\pi} \frac{\partial \psi(a, \varphi)}{\partial \varphi} d\varphi = 0,$$

and

$$\int_0^{2\pi} au_\varphi(a, \varphi) d\varphi = \Gamma = 2\pi\kappa.$$

Thus, the Poisson integrals for the radial and azimuthal velocity components are given by

$$u_r(r, \varphi) = \frac{a}{2\pi r} \int_0^{2\pi} u_r(a, \vartheta) g\left(\frac{a}{r}, \varphi, \vartheta\right) d\vartheta, \tag{21}$$

$$u_\varphi(r, \varphi) = \frac{\kappa}{r} + \frac{a}{2\pi r} \int_0^{2\pi} \left(u_\varphi(a, \vartheta) - \frac{\kappa}{a}\right) g\left(\frac{a}{r}, \varphi, \vartheta\right) d\vartheta, \tag{22}$$

with  $\kappa = \sum_{m=m_1}^{m_2} (\omega_m A_m)/(2\pi)$ ,  $A_m$  being the area of  $\mathcal{G}_m$  and  $g$  defined by (18).

Note that the term  $(\kappa/a) g(\frac{a}{r}, \varphi, \vartheta)$  in (22) does not contribute to the integral since

$$\int_0^{2\pi} g\left(\frac{a}{r}, \varphi, \vartheta\right) d\vartheta = 0.$$

4.2. *Numerical integration.* The numerical integration of the Poisson integrals (21) and (22) appears to be inaccurate when integrating straightforwardly by a trapezoidal rule, similar to the Poisson integral (10). In this section, the source of this inaccuracy is discussed and a solution (similar to that of Anderson [1]) to the problem is presented; the integration turns out to be super-convergent. For the sake of simplicity, the analysis is performed for the function  $f$  being the solution of the boundary-value problem (16) and satisfying (19).

The function  $g$  as defined in (18) can be expanded in a Fourier series

$$\begin{aligned} g(\varrho, \varphi, \vartheta) &= 2\varrho \frac{\cos(\varphi - \vartheta) - \varrho}{1 - 2\varrho \cos(\varphi - \vartheta) + \varrho^2} \\ &= \sum_{n=-\infty, n \neq 0}^{\infty} \varrho^{|n|} e^{in(\varphi - \vartheta)}, \quad \text{for } \varrho < 1. \end{aligned} \quad (23)$$

Substituting this in expression (20) for  $f$  and exchanging summation and integration, yields

$$f(r, \varphi) = \sum_{n=-\infty, n \neq 0}^{\infty} c_n \left(\frac{a}{r}\right)^{|n|} e^{in\varphi}, \quad \text{for } r > a,$$

where

$$c_n = \frac{1}{2\pi} \int_0^{2\pi} f(a, \vartheta) e^{-in\vartheta} d\vartheta.$$

Note that requirement (19) is equivalent to  $c_0 = 0$ . From the foregoing it follows that numerical calculation of  $f$  is actually equivalent to the numerical calculation of the Fourier coefficients  $c_n$ . Now approximate these coefficients using a  $K$ -point trapezoidal rule, where  $K$  is an odd number, i.e.,  $K = 2L + 1$ ,  $L \in \mathbb{N}$ . The approximation  $F_n$  of  $c_n$  is then given by

$$F_n = \frac{1}{K} \sum_{k=1}^K f(a, \vartheta_k) e^{-in\vartheta_k},$$

with  $\vartheta_k = 2\pi k/K$ . Evidently, when applying the same trapezoidal rule, the approximation  $T_K(fg)$  of  $f$  is then given by

$$T_K(fg) = \sum_{n=-\infty, n \neq 0}^{\infty} F_n \left(\frac{a}{r}\right)^{|n|} e^{in\varphi}, \quad \text{for } r > a.$$

The row  $\{F_n\}$ ,  $n = -L, \dots, L$ ,  $n \neq 0$ , is the *discrete Fourier transform* (DFT) of the row  $\{f(a, \vartheta_k)\}$ ,  $k = 1, \dots, K$  and  $F_n$  is periodic in  $n$  with period  $K = 2L + 1$ ; see Briggs and Emden Henson [4]. If  $f(a, \vartheta)$  is a periodic function of  $\vartheta$  and  $f^{(p)}$  is bounded and piecewise monotone, then [4]

$$|F_n - c_n| \leq \frac{C_1(p)}{K^{p+1}}, \quad \text{for } n = -L, \dots, L, \quad n \neq 0, \quad (24)$$

and furthermore

$$|c_n| \leq \frac{C_2(p)}{n^{p+1}}, \quad \text{for } n = -L, \dots, L, \quad n \neq 0. \quad (25)$$

The constants  $C_1$  and  $C_2$  are independent from  $n$ , but they do depend on  $p$ .

Now approximate the function  $g(\varrho, \varphi, \vartheta)$  by a function  $g_L(\varrho, \varphi, \vartheta)$  defined as

$$g_L(\varrho, \varphi, \vartheta) := \sum_{n=-L, n \neq 0}^L \varrho^{|n|} e^{in(\varphi-\vartheta)}, \quad \text{for } \varrho < 1,$$

i.e., the first  $L$  Fourier modes of  $g$  only. Then an approximation  $\tilde{f}$  of  $f$  is given by

$$\tilde{f}(r, \varphi) = \sum_{n=-L, n \neq 0}^L c_n \left(\frac{a}{r}\right)^{|n|} e^{in\varphi}, \quad \text{for } r > a.$$

Numerical integration with the same  $K$ -point trapezoidal rule as before yields

$$T_K(fg_L) = \sum_{n=-L, n \neq 0}^L F_n \left(\frac{a}{r}\right)^{|n|} e^{in\varphi}, \quad \text{for } r > a.$$

The error caused by numerical integration can now be estimated using (24),

$$\begin{aligned} |\tilde{f}(r, \varphi) - T_K(fg_L)| &= \left| \sum_{n=-L, n \neq 0}^L (c_n - F_n) \left(\frac{a}{r}\right)^{|n|} e^{in\varphi} \right| \\ &\leq \frac{C_1(p)}{K^{p+1}} \sum_{n=-L, n \neq 0}^L \left(\frac{a}{r}\right)^{|n|} \leq \frac{C_1(p)}{K^{p+1}} 2L \left(\frac{a}{r}\right) \\ &\leq \frac{C_1(p)}{K^p} \left(\frac{a}{r}\right). \end{aligned} \quad (26)$$

With the estimate (25) for  $c_n$ , it follows that

$$\begin{aligned} |f(r, \varphi) - \tilde{f}(r, \varphi)| &= \left| \sum_{|n| \geq L+1} c_n \left(\frac{a}{r}\right)^{|n|} e^{in\varphi} \right| \leq \sum_{n=L+1}^{\infty} \frac{2C_2(p)}{n^{p+1}} \left(\frac{a}{r}\right)^n \\ &\leq \frac{2C_2(p)}{pL^p} \left(\frac{a}{r}\right)^{L+1}. \end{aligned} \quad (27)$$

Combination of (26) and (27) yields the estimate

$$|f(r, \varphi) - T_K(fg_L)| \leq \frac{C_1(p)}{K^p} \left(\frac{a}{r}\right) + \frac{2C_2(p)}{pL^p} \left(\frac{a}{r}\right)^{L+1}. \quad (28)$$

In a similar way, an estimate can be found for  $|f(r, \varphi) - T_K(fg)|$ . Since

$$f(r, \varphi) - T_K(fg) = f(r, \varphi) - T_K(fg_L) - \sum_{n=L+1}^{\infty} \left(\frac{a}{r}\right)^n (F_n e^{in\varphi} + F_{-n} e^{-in\varphi}),$$

it follows with (28) that

$$|f(r, \varphi) - T_K(fg)| \leq \frac{C_1(p)}{K^p} \left(\frac{a}{r}\right) + \frac{2C_2(p)}{pL^p} \left(\frac{a}{r}\right)^{L+1} + \left| \sum_{|n| \geq L+1} \left(\frac{a}{r}\right)^{|n|} F_n e^{in\varphi} \right|.$$

The last term on the right-hand side can be estimated using the periodicity of  $F_n$ . Substitute  $n = n' + (2L + 1)l = n' + Kl$ , for  $n' = -L, \dots, L$  and  $l = 1, 2, 3, \dots$ . Then, after omitting the primes, it follows that

$$\begin{aligned} \sum_{|n| \geq L+1} \left(\frac{a}{r}\right)^{|n|} F_n e^{in\varphi} &= \sum_{n=-L}^L \sum_{l=1}^{\infty} \left(\frac{a}{r}\right)^{n+Kl} (F_n e^{in\varphi+iKl\varphi} + F_{-n} e^{-in\varphi-iKl\varphi}) \\ &= \sum_{n=-L}^L \left(\frac{a}{r}\right)^n F_n e^{in\varphi} \frac{\left(\frac{a}{r} e^{i\varphi}\right)^K}{1 - \left(\frac{a}{r} e^{i\varphi}\right)^K} \\ &\quad + \sum_{n=-L}^L \left(\frac{a}{r}\right)^n F_{-n} e^{-in\varphi} \frac{\left(\frac{a}{r} e^{-i\varphi}\right)^K}{1 - \left(\frac{a}{r} e^{-i\varphi}\right)^K}. \end{aligned}$$

This equation shows that the infinite number of modes of  $f$  reduces to only  $K$  modes when the integral is integrated numerically and, moreover, the higher modes are represented falsely by lower modes (aliasing). Now inequalities (24) and (25) can be used to find an estimate for  $F_n$ ,  $n \neq 0$ ,

$$|F_n| \leq |F_n - c_n| + |c_n| \leq \frac{C_1(p)}{K^{p+1}} + \frac{C_2(p)}{n^{p+1}}.$$

For  $n = 0$  it simply follows from  $c_0 = 0$  that

$$|F_0| \leq \frac{C_1(p)}{K^{p+1}}.$$

With these estimates for  $|F_n|$ , it now follows that

$$\left| \sum_{|n| \geq L+1} \left(\frac{a}{r}\right)^{|n|} F_n e^{in\varphi} \right| \leq \frac{2C_1(p)}{K^p} \frac{(a/r)^{L+1}}{1 - (a/r)} + \frac{4C_2(p)}{pL^p} \frac{(a/r)^{L+1}}{1 - (a/r)^K},$$

resulting in a final estimate for the integration error according to

$$\begin{aligned} |f(r, \varphi) - T_K(fg)| &\leq \frac{C_1(p)}{K^p} \left(\frac{a}{r}\right) + \frac{2C_2(p)}{pL^p} \left(\frac{a}{r}\right)^{L+1} \\ &\quad + \frac{2C_1(p)}{K^p} \frac{(a/r)^{L+1}}{1 - (a/r)} + \frac{4C_2(p)}{pL^p} \frac{(a/r)^{L+1}}{1 - (a/r)^K}. \end{aligned} \quad (29)$$

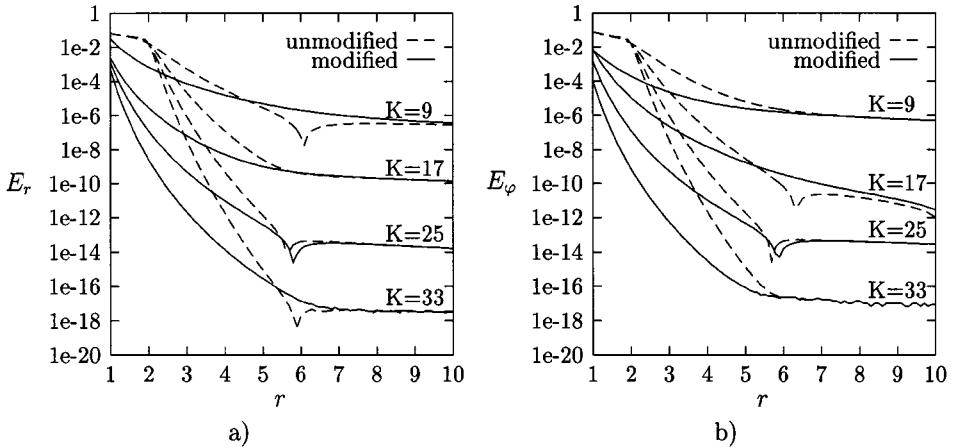
The errors  $E_r$  and  $E_\varphi$  in the velocity components can now be obtained by simply replacing the function  $f$  in the estimates and (28) and (29) by  $ru_r$  and  $ru_\varphi - \kappa$ , respectively. In the case of an unmodified kernel, it then follows that

$$E_r \leq \frac{C_1(p)}{K^p} \left(\frac{a}{r}\right)^2 + \frac{2C_2(p)}{pL^p} \left(\frac{a}{r}\right)^{L+2} + \frac{2C_1(p)}{K^p} \frac{(a/r)^{L+2}}{1 - (a/r)} + \frac{4C_2(p)}{pL^p} \frac{(a/r)^{L+2}}{1 - (a/r)^K},$$

while in the case of the modified kernel

$$E_r \leq \frac{C_1^1(p)}{K^p} \left(\frac{a}{r}\right)^2 + \frac{2C_2^1(p)}{pL^p} \left(\frac{a}{r}\right)^{L+2}.$$

For  $E_\varphi$ , similar estimates are valid, yet with different constants  $C_1$  and  $C_2$ .



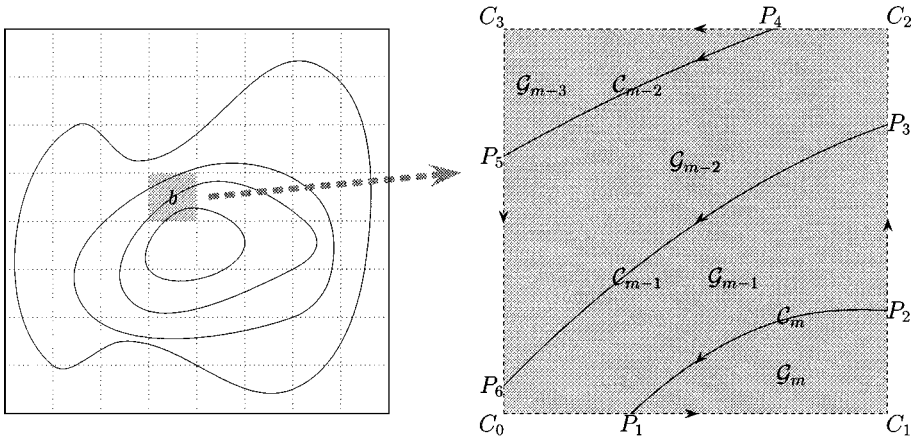
**FIG. 6.** The magnitude of the error  $E_r$  (a) and  $E_\varphi$  (b). The dashed lines indicate the magnitude of the errors in the Poisson integrals when applying the trapezoidal rule to the original kernel (i.e.,  $g$ ); the solid lines indicate the magnitude of the errors when applying the trapezoidal rule to the modified kernel (i.e.,  $g_L$ ).

The estimates for the unmodified kernel are larger than the estimates in case of a modified kernel. Moreover, the former estimates are unbounded when  $r = a$ , while the latter are bounded. This suggests that using  $g_L$  instead of  $g$  will reveal more accurate results. In Fig. 6 the magnitude of the errors  $E_r$  (left) and  $E_\varphi$  (right) are plotted both for an unmodified kernel (dashed lines) and for a modified kernel (solid lines). The test problem is the same problem Anderson used [1], i.e., a ring of radius  $a = 2$  centred around  $(r, \varphi) = (0, 0)$  and a particle of strength  $\kappa = 1$  located at  $(r, \varphi) = (\frac{1}{2}\sqrt{2}, \pi/3)$ . In the figure, the magnitude of the errors is plotted as a function of the distance to the centre of the ring. The evaluation points are all located at the positive  $x$ -axis ( $\varphi = 0$ ). Indeed, at evaluation points close to the ring the modified kernels give better results than the unmodified kernels. The behaviour of the error agrees very well with the predicted behaviour. The cusps that are present in the figure are caused by a change in sign of the error.

## 5. CONSTRUCTION OF FINEST LEVEL OUTER-RINGS

The last part of the method consists of constructing outer-rings at the finest level. At this level, the contributions of the patches of uniform vorticity in each box have to be determined at the outer-rings. Consider the arbitrary piecewise-uniform vorticity distribution as depicted in Fig. 7. In the right part of the figure, some regions  $\mathcal{G}_m$ ,  $m = 0, \dots, M$ , of uniform vorticity  $\omega_m$  are shown. Region  $\mathcal{G}_0$  is equal to the domain on which the HEM is applied (containing all regions of vorticity) and  $\omega_0$  is assumed to be zero. In the left part of the figure, the grid lines of the finest level are indicated with dashed lines. The contribution of the vorticity distribution inside the grey box  $b$  to the velocity at the integration points on the outer-ring of that box is now given by

$$\mathbf{u}(\mathbf{x}, t) = - \sum_{m=1}^M \frac{\omega_m}{2\pi} \oint_{\partial(\mathcal{G}_m|b)} \ln \|\mathbf{x} - \mathbf{x}'\| d\mathbf{x}', \quad (30)$$



**FIG. 7.** The finest level grid (dotted lines in the left part) with a piecewise uniform vorticity distribution. The contours where  $\omega$  jumps discontinuously are drawn with solid lines. On the right, an enlarged view of the grey shaded box in the left part.

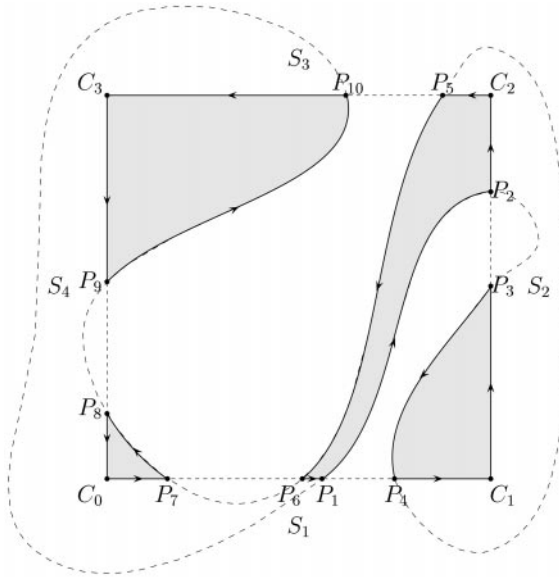
where  $\mathcal{G}_m|_b$  is the region of  $\mathcal{G}_m$  restricted to box  $b$ . In the example as shown in Fig. 7, this results in

$$\begin{aligned} \mathbf{u}(\mathbf{x}, t) = & -\frac{1}{2\pi} \left[ \omega_{m-3} \oint_{C_0 C_1 C_2 C_3 C_0} + \omega_{m-2} \oint_{P_5 C_0 C_1 C_2 P_4 P_5} \right. \\ & \left. + \omega_{m-1} \oint_{P_6 C_0 C_1 P_3 P_6} + \omega_m \oint_{P_1 C_1 P_2 P_1} \right] \ln \|\mathbf{x} - \mathbf{x}'\| d\mathbf{x}', \end{aligned}$$

with the points  $C_0, \dots, C_3$  and  $P_1, \dots, P_6$  as depicted in the right part of the figure. Thus, for the construction of the finest level outer-ring of a certain box it is necessary to determine the contributions from two different sources: the first source being the parts of the contours inside the box and the second source being the correct parts of the box boundaries. In practice, it turns out to be convenient to carry out this procedure for all boxes per contour.

The contributions of the first source can be determined by moving along the contour in positive direction, i.e., counterclockwise, and determining for each node in which box it resides. The number of the box in which the node is situated is stored in such a way that for each node it is clear in which box its preceding node is situated. If the box number of the current node differs from that of its predecessor, the part of the contour between those two nodes crosses at least one grid line and the current segment contributes to the outer-ring of at least two boxes. After the intersection points with those grid lines are determined, the contributions to the various boxes can be determined. If the box number does not differ from that of the preceding node, the whole line segment is located in one box and the contribution of the segment to the outer-ring of that box (and contour nodes in neighbouring boxes<sup>3</sup>) can be determined. Upon treating all nodes on the contour this way, the contributions of the first source have been determined.

<sup>3</sup> These contributions are needed in the last sweep of the HEM (see Section 3), but in the contour dynamics case, it is more convenient to determine them already at this stage of the method. This way, the contours only need to be traversed once.



**FIG. 8.** Example of a box that is crossed by a contour more than once. The grey shaded regions are the regions that contribute to the outer-ring. The parts of the contours and of the box boundaries that have to be taken into account are drawn with solid lines.

The determination of contributions of the second source is more complex. One problem encountered is the possibility of the contour crossing a box more than once. The intersection points of the contour with the box boundaries have already been determined in the previous part, but the sequence in which they have been found is not of any use. This is illustrated by Fig. 8. Here, the intersection points  $P_i$  are, for example, found in the order of increasing index  $i$ . It is clear from this figure that the parts of the box boundaries that have to be taken into account (solid lines) should always be walked along in positive direction, since the contour itself is also walked along in that direction. Thus, it is necessary to sort the array of intersection points of a given box in such a way that (after sorting) the intersection points are encountered successively in positive direction when moving through that array. So, in the case of Fig. 8 the sequence of intersection points should be  $P_7, P_6, P_1, P_4, P_3, P_2, P_5, P_{10}, P_9, P_8$ , after the sorting algorithm has been applied to the original sequence. A heapsort algorithm [6] is used for this purpose.

Another feature that is clear from Fig. 8 is that if the contour moves inward into the box, it has to move outward again, since it is closed. Therefore, the number of intersection points for a given box is always an even number. Moreover, it appears that the parts of the box boundaries that have to be taken into account are the parts between two neighbouring intersection points (not necessarily lying on the same box side) of which at the first the contour is moving outward and at the next (in positive direction) the contour is moving inward again. By giving each intersection point a flag (*sside*) indicating whether the contour is moving inward ( $sside > 0$ ) or outward ( $sside < 0$ ) at that point, it can be determined between which two subsequent intersection points a box boundary part, that should be accounted for, is situated. Since it is not a priori clear how many and which corners of the boundary are lying between such two subsequent intersection points, it is convenient to identify each side and each corner with a number (see Fig. 8). In this way, an intersection point can be assigned an integer value indicating the side it is situated on.



This value can be combined with the flag mentioned above:  $sside(P_i, b) = \pm k$  where  $k$  is the number of the side of box  $b$  where  $P_i$  is lying on and the sign indicates whether the contour moves inward or outward box  $b$  at  $P_i$ . The number of corners between two subsequent intersection points can now be found by subtracting the absolute value of  $sside$  of the first point from that of the next. The first corner which is encountered while moving from one intersection point to the next is the corner with number equal to the absolute value of  $sside$  of the former intersection point. For example, for the two subsequent intersection points  $P_2$  and  $P_5$  in Fig. 8, it follows that  $sside(P_2, b) = -2$  and  $sside(P_5, b) = +3$ , so that  $|sside(P_5, b)| - |sside(P_2, b)| = 3 - 2 = 1$ . This means that there is one corner between  $P_2$  and  $P_5$ , and the number of that corner is equal to  $|sside(P_2, b)| = 2$ . Using this strategy, the contributions of the boundary parts of the boxes that are crossed by the contour can be determined.

Another problem that can be encountered when determining the contribution of the second source is the possibility of a box lying completely inside the interior of the contour. In this case, the whole boundary of the box contributes to the outer-ring. A way to detect these kind of boxes is to move through the boxes from left to right, row by row. For a certain box in a certain row, there are three possibilities: the box is intersected by the contour and thus it is not lying completely inside the contour; the box is not intersected by the contour and it is lying completely outside the contour; the box is not intersected and it is lying completely inside the contour. It is easy to detect whether the box under consideration is of the first kind or not, since it is known how many intersection points there are (if there are intersection points, then it is a box of the first kind). It is more complicated, however, whether in the case of no intersection points the box is of the second or the third kind. In that case it is necessary to have some extra information: if the left side of the current box is lying completely inside the contour and it is not intersected by the contour, then the current box is of the third kind and thus lying completely inside the contour. Note that the left side of the current box is the right side of the previous box. When examining a box a flag is set which indicates whether the right side is completely inside the contour or not. This flag is used when considering the next box in case it is not crossed by the contour to determine of which kind (second or third) it is. A pseudo-code describing this part of the method can be found in [20].

This part of the method may appear rather complicated and time consuming at first glance. The implementation, however, is done quite efficiently so that the total amount of CPU-time required for the construction of the finest level outer-rings is almost completely determined by the calculation of the integrals along contour segments. The algorithms for finding out which part of the contour belongs to which box, and which box is lying completely inside a contour, requires at most a few percent of the total amount of CPU-time needed for the construction of the finest level.

## 6. NUMERICAL EXPERIMENTS AND DISCUSSION

In this section some numerical experiments are discussed in order to demonstrate the accuracy and the speed-up of the hierarchical-element method.

The accuracy is tested on the following example.

**EXAMPLE 6.1.** This example concerns the evolution of a monopolar vortex into a tripolar vortex, which is a vortex consisting of an elliptic core with two satellites of opposite sign

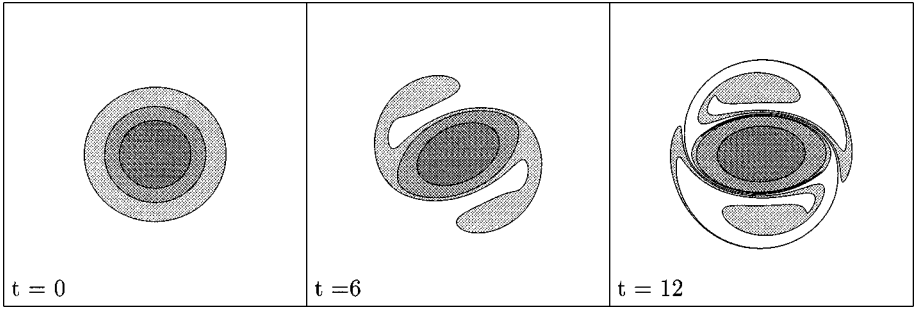


FIG. 9. The evolution of a monopolar vortex into a tripolar vortex.

[16, 18]. The initial configuration consists of three concentric, slightly elliptically disturbed, contours (aspect ratio is equal to 100/95). The outer ring has negative vorticity, while the core (consisting of the area enclosed by the second contour) has positive vorticity (see also the paper by Vosbeek and Mattheij [22]). Due to the elliptical disturbance, the monopole deforms and becomes a tripole while the core is becoming more elliptical. The evolution is shown in Fig. 9. With the new method, four simulations have been performed for various choices of the number of integration points  $K$  on the rings, namely  $K = 9, 17, 25,$  and  $33$ . Obviously, the higher the value of  $K$ , the more accurate the results should become. During the simulations, the number of levels is automatically adapted (see Section 3) and  $l_f$  increases from  $l_f = 1$  at the beginning up to  $l_f = 5$  at later stages. In Fig. 10 the (relative) difference in area, enclosed by the contours, relative to the conventional method, is plotted as a function of time  $t$  for the three different contours. It is clear from this figure that  $K = 9$  yields the largest difference. In fact, in that case, the differences are larger than the errors in the area caused by the time integration (see the paper by Vosbeek and Mattheij [22]) and it can be concluded that the method is not accurate enough with this value of  $K$ . Note that the errors for  $K = 33$  tend to become larger than those for  $K = 17$  and  $K = 25$ . The reason for this is not quite clear.

It is interesting to compare the shapes of the contours of the present simulations with those obtained with the conventional method as used in the paper by Vosbeek and Mattheij [22]. In Fig. 11a, the contours at time  $t = 12$  for both simulations are plotted together in the

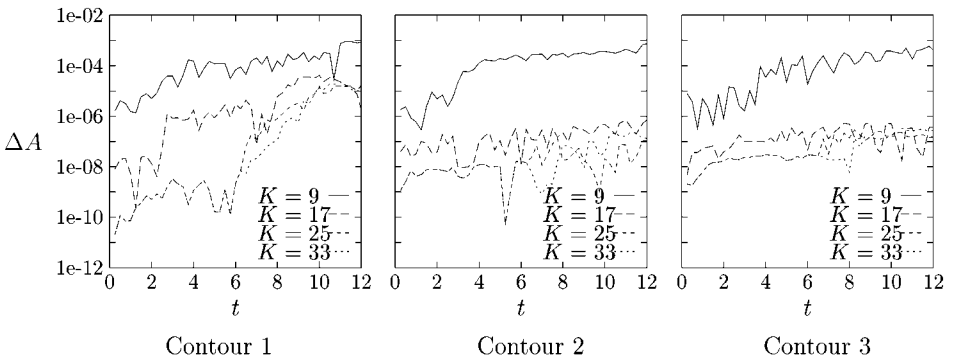
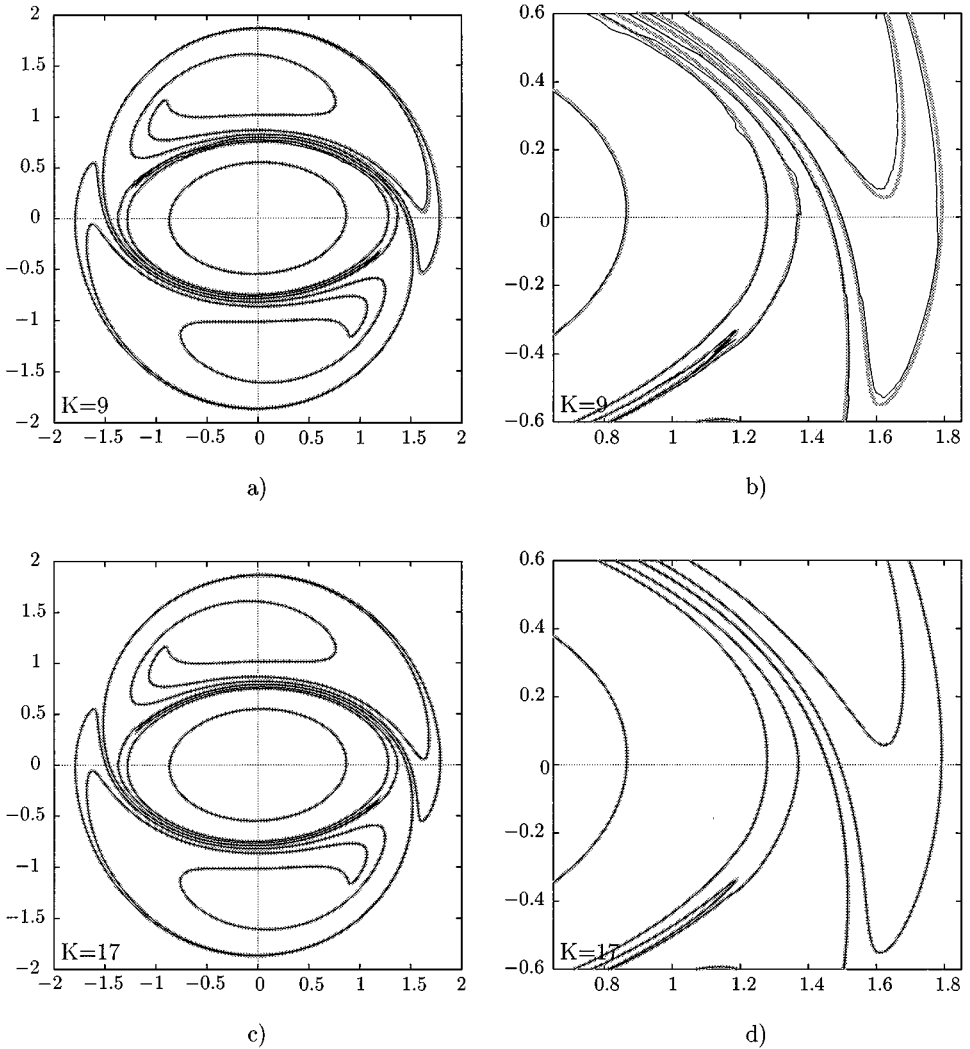


FIG. 10. Difference of the area compared to the results of the conventional method for several values of  $K$  as a function of time  $t$ .



**FIG. 11.** The shape of the tripole at  $t=12$ . In all panels, the contours of the tripole computed with the conventional method (thick grey line) and with the HEM method (thin black line) are drawn in the same graph. In the top panels,  $K=9$  is used in the HEM method; in the bottom panels  $K=17$  is used. In the panels on the right, an enlarged view of a part of the tripole is shown.

same graph. The contours of the conventional method are plotted with thick grey lines, whereas the contours of the new code with  $K=9$  are plotted with thin black lines. To make the difference more clear, an enlarged view of a part of the tripole is given in Fig. 11b. Obviously, the contours are less smooth for the case  $K=9$  than for  $K=17$  and there are substantial differences with the contours produced by the conventional method. Using these results in further calculations will yield even larger differences at later stages. These results confirm that the results for  $K=9$  are not reliable.

For  $K=17$ , however, the calculations are much better, as can be observed from both Fig. 10 and the bottom panels of Fig. 11. For this value of  $K$ , the differences in the area remain smaller than the errors caused by the time integration. Figure 11c shows again both the contours obtained with the conventional method and the ones obtained by HEM.

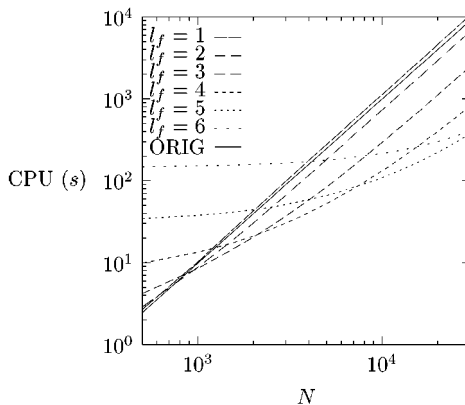
Figure 11d shows an enlarged view of the same part of the tripole as for  $K = 9$ . Now, there is no visible difference in the shape of the contours and also further calculations did not reveal any; so for  $K = 17$ , the method is much more reliable.

For higher values of  $K$  ( $K = 25$  and  $K = 33$ ), the differences in the area are smaller than or comparable to the  $K = 17$  case. Although not shown here, it may be clear that the contours again show no visible differences with the ones produced by the conventional method.

The higher accuracy of the method for larger values of  $K$  has a price, of course. The higher  $K$ , the more computationally expensive the method becomes. However, the speed-up is still significant as illustrated with the following example.

**EXAMPLE 6.2.** To test the speed-up of the new method, several computations are performed. For these simulations, a circular vortex patch is used with five contours. The radii of the contours are 1.0, 2.0, 3.0, 4.0, and 5.0. The domain of the multipole method is chosen  $[-6.0, 6.0] \times [-6.0, 6.0]$ . The five contours all have the same fixed number of nodes in each calculation. Computations are performed for 100, 200,  $\dots$ , 800, 1000, 1200,  $\dots$ , 4000, 5000, 6000 equispaced nodes per contour with number of levels  $l_f = 1, 2, \dots, 6$  and  $K = 25$ . The conventional method was tested for the same numbers of nodes per contour. In Fig. 12, the CPU-time (in seconds on one R8000 processor of a Silicon Graphics Power Challenge) for one calculation of the velocity field is plotted as a function of the total number of nodes. Note that in this figure both the horizontal and vertical axis have logarithmic scales. The curve corresponding to the conventional method is drawn with a solid line; the curves for the HEM method are all drawn with differently dashed lines.

The curve corresponding to the conventional method is a straight line and closer inspection reveals that this agrees with the  $\mathcal{O}(N^2)$  behaviour. Furthermore, it can be observed that the larger the value of  $l_f$  is, the slower the CPU-time increases with  $N$ . But also, the larger  $l_f$ , the higher the CPU-time is for small values of  $N$ . From these two observations it follows that the value of  $l_f$  that should be chosen in order to have maximum speed-up, depends on the number of nodes (as expected, see Section 3): the larger the number of nodes, the higher  $l_f$ . In Fig. 13, the maximum speed-up factor is plotted as a function of  $N$ . This factor was determined by dividing the CPU-time needed by the conventional method by the CPU-time



**FIG. 12.** The CPU-time, for calculating the velocities for a patch with 5 contours, as a function of the number of nodes  $N$  for several values of  $l_f$  (dashed lines) and the conventional method (solid line).

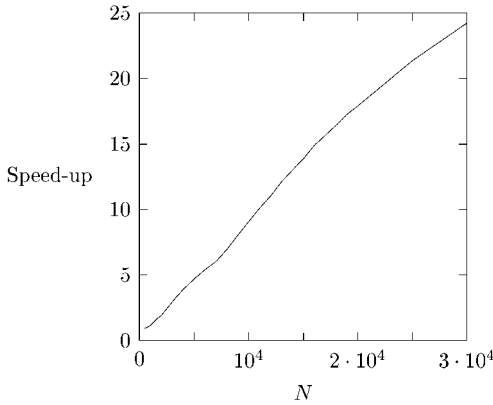


FIG. 13. The speed-up factor as a function of the total number of nodes  $N$  for the most optimal choice of  $l_f$ .

needed by the HEM, this for the most optimal choice of  $l_f$  for  $N$  nodes. Obviously, the speed-up of the HEM method is much larger for high values of  $N$  (speed-up factor  $\approx 25$  for  $3 \cdot 10^4$  nodes!) than for small values of  $N$  (speed-up factor  $\approx 5$  for  $5 \cdot 10^3$  nodes). The speed-up factor increases almost linearly with the number of nodes, which means that if  $l_f$  is chosen optimal in the computations, the method is almost of order  $\mathcal{O}(N)$ . This is a substantial improvement compared to the conventional method and it may be clear that the speed-up is even larger in the case of  $K = 17$ .

The previous example tests the speed-up of the method for a rather simple configuration of contours. The speed-up for more complex problems will probably be smaller since the construction of the outer-rings at the finest level is more complicated in those cases. Due to an efficient implementation of the algorithms discussed in Section 5, however, this effect is expected to be small.

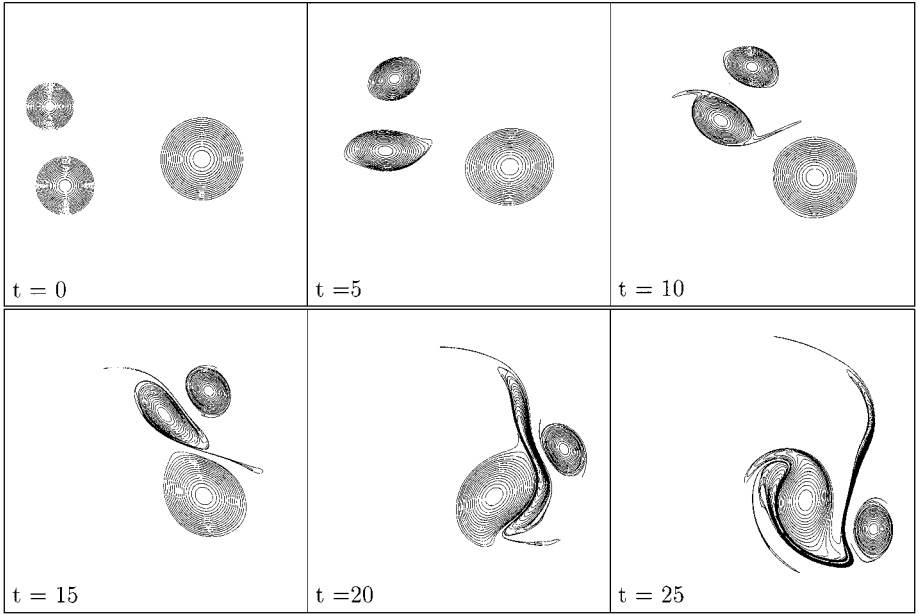
EXAMPLE 6.3. A last example to illustrate the speed-up of the new method concerns a rather special interaction of three initially circular monopoles. The initial configuration of the monopoles is chosen such that a so-called *collapse* of the three vortices would occur in case the monopoles of finite area are replaced by point vortices of the same strengths and locations [2, 3, 12–14, 17]. In the point vortex case, the trajectories of the point vortices have the form of logarithmic spirals with a common origin. During their interaction, the three point vortices move along the trajectories towards the origin and collapse there in finite time.

In the paper by Vosbeek *et al.* [21], the point vortices are replaced by initially circular monopoles of finite size and several numerical simulations were carried out to study the influence of, e.g., viscosity and the size of the monopoles on the interaction behaviour of the vortices. Some additional simulations can be found in [20].

In this example, the same initial configuration is chosen as in [20, 21], i.e.,

$$\begin{cases} x_1(0) = -4, \\ y_1(0) = 0, \\ \Gamma_1 = -3, \end{cases} \quad \begin{cases} x_2(0) = -\frac{9}{2}, \\ y_2(0) = \frac{3}{2}\sqrt{3}, \\ \Gamma_2 = 2, \end{cases} \quad \begin{cases} x_3(0) = \frac{1}{2}, \\ y_3(0) = \frac{1}{2}\sqrt{3}, \\ \Gamma_3 = -6. \end{cases}$$

For each of the three vortices, 16 contours (and thus 16 discrete levels of vorticity) are used



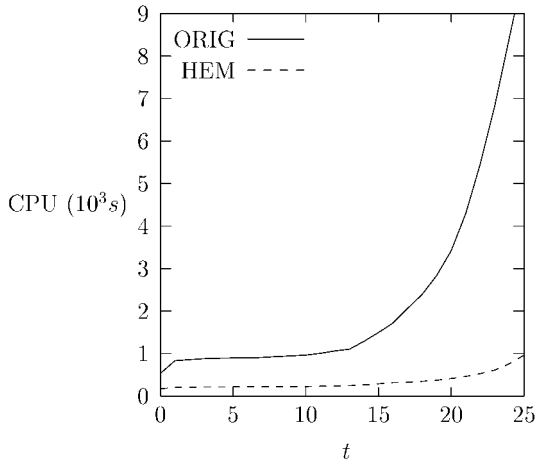
**FIG. 14.** The evolution of the three, unequally sized, circular monopoles.

to mimic the continuous vorticity distribution of a so-called *Bessel vortex*. The vorticity distribution of a Bessel vortex is given by

$$\omega(r) = \begin{cases} \frac{k\Gamma}{2\pi R J_1(kR)} J_0(kr), & r \leq R, \\ 0, & r \geq R, \end{cases}$$

where  $r$  is the radial distance to the centre of the vortex,  $R$  its radius, and  $\Gamma$  its strength or circulation.  $J_0$  and  $J_1$  are Bessel functions of the first kind and  $kR \approx 2.4048$  is the first zero of  $J_0$ . The radii of the three vortices are chosen differently and such that the vortices have the required strength, but the maximum of vorticity is the same for all three vortices. As a consequence, the three discretised vortices have the same 16 discrete levels of vorticity. The evolution of the vortices is shown in Fig. 14. In the case of a similar configuration of three point vortices, a collapse into one single point vortex would take place at  $t = 25.3952$ .

Here, the evolution of the three vortices has been calculated (on the same computer as in the previous example) both using the original method and the HEM (with  $K = 17$ ). During the calculation, the CPU-time necessary to do the velocity calculations (four times per time step) has been monitored. In Fig. 15, the evolution of the CPU-time (in seconds) during the two calculations is plotted. The time on the horizontal axis corresponds to the time in the evolution of the vortices as given in Fig. 14. The number of levels in the case of the HEM method is  $l_f = 5$  in the first part of the calculation (until  $t \approx 16$ ) and  $l_f = 6$  in last part. The total number of nodes increases from 4686 initially up to 17,895 at  $t = 25$ . It is clear from this graph that the computation time increases far less dramatically in the HEM case than with the original method. This results in a total computation time for the HEM method (89,604 s) which is approximately 6.5 times smaller than the original method (579,512 s).



**FIG. 15.** The evolution of the CPU-time (in seconds) needed for the four velocity calculations per time step as a function of time  $t$ . For the original method this is shown with a solid line and for the HEM with a dashed line.

## 7. DISCUSSION AND CONCLUSIONS

In this paper, a method is presented to accelerate contour dynamics simulations. It is based on the hierarchical-element method developed by Anderson [1] which turned out to be very suitable for this purpose.

The hierarchical-element method is very similar to the fast multipole method by Greengard and Rokhlin [11], which is commonly used for the calculations of many particle interactions. However, instead of multipole expansions, this method makes use of Poisson integrals. An advantage of this is that the method is not only applicable to problems concerning particle interactions, but also to problems with a different kind of “charge distribution,” like piecewise uniform distributions of vorticity as in the case of the contour dynamics method. In those more general cases, however, the method by Anderson has to be adapted in some sense. For application to contour dynamics, for example, it is necessary to derive the appropriate Poisson integrals. Furthermore, the construction of the finest level approximations of the Poisson integrals is different and slightly more complex than in the particle case. Nevertheless, the numerical examples presented in this paper show that the resulting method turns out to be very accurate, while the speed-up is significant.

The development of this acceleration method makes it possible to handle very complex flow problems. For example, the behaviour of vortices in the presence of non-uniform background vorticity (e.g., on the  $\beta$ -plane or the  $\gamma$ -plane, being approximations of the rotating earth at midlatitudes and poles, respectively) can be studied now using this method. In those cases, contours are also present outside the vortices [20], in contrast to the examples shown in this paper where only contours were present inside the vortices. As a consequence, the total number of nodes can become much larger than in the examples shown here. Without accelerating contour dynamics, carrying out such simulations would be virtually impossible.

Another advantage of the hierarchical-element method as presented here is that this method is quite suitable for parallelisation. The computations of the outer-rings and inner-rings at a certain level are independent from each other and can thus be carried out simultaneously. This way, even higher speed-up rates would be possible. A similar effect could be achieved by adapting the size of the domain during an evolution in such a way that at

each time step, the domain used in the HEM, is the smallest domain possible. This way, the number of boxes with only few nodes (which are relatively expensive to compute) is minimised. Another way to improve efficiency could be by making use of non-uniform refinements, i.e., refine locally where the density of nodes is high. Finally, choosing a rectangular computational domain instead of a square domain (like in the examples presented here) could also lead to higher speed-up values in some cases [1].

The hierarchical-element method is used in this paper to speed-up simulations of flow problems in an infinite domain (though the computational domain is bounded, of course), but it would also be possible to incorporate boundary conditions according to Greengard and Rokhlin [11]. Furthermore, a generalisation of the method to three dimensions to study, for example, rotating, stratified flows, is possible as well (see, e.g., Anderson [1]). This would make the contour dynamics method even more generally applicable.

### ACKNOWLEDGMENT

The authors thank Professor Dr. J. Boersma for his useful suggestions.

### REFERENCES

1. C. R. Anderson, An implementation of the fast multipole method without multipoles, *SIAM J. Sci. Statist. Comput.* **13**, 923 (1992).
2. H. Aref, Motion of three vortices, *Phys. Fluids* **22**, 393 (1979).
3. H. Aref, N. Rott, and M. Thomann, Gröbli's solution of the three-vortex problem, *Ann. Rev. Fluid Mech.* **24**, 1 (1992).
4. W. L. Briggs and V. Emden Henson, *The DFT: An Owner's Manual for the Discrete Fourier Transform* (SIAM, Philadelphia, 1995).
5. T. F. Buttké, A fast adaptive vortex method for patches of constant vorticity in two dimensions, *J. Comput. Phys.* **89**, 161 (1990).
6. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms* (MIT Press, Cambridge, MA, 1993).
7. D. G. Dritschel, Contour surgery: A topological reconnection scheme for extended integrations using contour dynamics, *J. Comput. Phys.* **77**, 240 (1988).
8. D. G. Dritschel, Contour dynamics and contour surgery: Numerical algorithms for extended, high-resolution modelling of vortex dynamics in two-dimensional, inviscid, incompressible flows, *Comput. Phys. Rep.* **10**, 77 (1989).
9. D. G. Dritschel, A fast contour dynamics method for many-vortex calculations in two-dimensional flows, *Phys. Fluids A* **5**, 173 (1993).
10. D. G. Dritschel, Moment accelerated contour surgery, in *Vortex Flows and Related Numerical Methods*, edited by J. T. Beale *et al.* (Kluwer Academic, Dordrecht, 1993).
11. L. Greengard and V. Rokhlin, A fast algorithm for particle simulations, *J. Comput. Phys.* **73**, 325 (1987).
12. W. Gröbli, Specielle Probleme über die Bewegung geradliniger paralleler Wirbelfäden, *Vierteljahr. Naturforsch. Gesellsch. Zürich* **22**, 37/129 (1877).
13. Y. Kimura, Similarity solution of two-dimensional point vortices, *J. Phys. Soc. Jpn.* **56**, 2024 (1987).
14. Y. Kimura, Chaos and collapse of a system of point vortices, *Fluid Dynam. Res.* **3**, 98 (1988).
15. E. Kreyszig, *Advanced Engineering Mathematics* (Wiley, New York, 1988).
16. S. J. Lin, Contour dynamics of tornado-like vortices, *J. Atmos. Sci.* **49**, 1745 (1992).
17. E. A. Novikov and Yu. B. Sedov, Vortex collapse, *Sov. Phys. JETP* **50**, 297 (1979).
18. P. Orlandi and G. J. F. van Heijst, Numerical simulation of tripolar vortices in 2D flow, *Fluid Dyn. Res.* **9**, 179 (1992).



19. J. M. Sanz-Serna and M. P. Calvo, *Numerical Hamiltonian Problems* (Chapman & Hall, London/New York, 1994).
20. P. W. C. Vosbeek, *Contour Dynamics and Applications to 2D Vortices*, Ph.D. thesis, Eindhoven University of Technology, Eindhoven, The Netherlands, 1998.
21. P. W. C. Vosbeek, J. H. G. M. van Geffen, V. V. Meleshko, and G. J. F. van Heijst, Collapse interactions of finite-sized two-dimensional vortices, *Phys. Fluids* **9**, 3315 (1997).
22. P. W. C. Vosbeek and R. M. M. Mattheij, Contour dynamics with symplectic time integration, *J. Comput. Phys.* **133**, 222 (1997).
23. N. J. Zabusky, M. H. Hughes, and K. V. Roberts, Contour dynamics for the Euler equations in two dimensions, *J. Comput. Phys.* **30**, 96 (1979).