Computing 65, 339–356 (2000)

Computing

Application of the Over-Set Grid Technique to a Model Singular Perturbation Problem

E. D. Havik, P. W. Hemker, and **W. Hoffmann,** Amsterdam

Received October 10, 1999

Abstract

The numerical solution of a singularly perturbed problem, in the form of a two-dimensional convection-diffusion equation, is studied by using the technique of over-set grids. For this purpose the Overture software library is used. The selection of component grids is made on basis of asymptotic analysis. The behavior of the solution is studied for a range of small diffusion parameters. Also the possibilities of rotating the grid with the convection direction is considered.
The basic discretisation method is central differencing, and in order to fit global properties of the solution, the composite grid used is made parameter dependent. In view of possible ε-uniform convergence, in the resulting composite grid the number of grid points is kept constant for the different values of the small parameter. Only the grid spacing is adapted, depending on the parameters. We see that, even with such careful adaptation of the grid, in the discrete maximum norm no ε-uniform convergence is achieved.

AMS Subject Classifications: 65N50 (Mesh generation and refinement).

Key Words: Chimera grids, over-set grids, singular perturbation problem.

1. Introduction

Over-set grids, overlapping grids or chimera grids are names used for the technique in which a discretization domain is covered by a set of overlapping component grids. All of these component grids are regularly structured grids, with a regular rectangular topology. Figure 1 gives a typical example of a overlapping grids. The combination of component grids is called a composite grid. As each component grid is structured, it is suitable for application of finite-difference methods. At overlapping regions, function values must be interpolated from one grid to another and vice versa.

The overlapping grid technique has a lot of advantages. The capability to generate a grid describing a domain with a difficult shape is often mentioned, but also for a simpler domain there are advantages. It is for e.g. possible to avoid polar singularities in a coordinate system, and it is also possible to construct a grid which handles the boundary conditions in a relatively easy way. Furthermore it is easy to rotate/translate a component grid, to change the number of grid lines of a grid in a certain direction, to change stretching parameters etc.. This great flexibility is a major argument for using over-set grids.

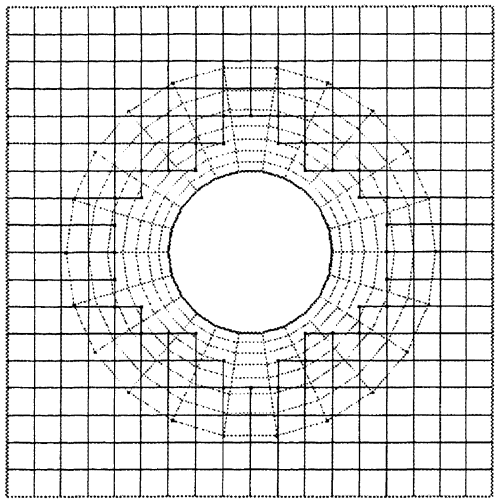

Figure 1. A standard example in which over-set grids are useful. The resulting grid is called a composite grid

We want to apply this technique to a model singular perturbation problem. The Overture software system [1], to be discussed later, will be our tool. The model problem is described by the convection diffusion equation, with a small diffusion parameter ε. If this parameter becomes smaller, the solution tends to a discontinuous function, and sharp boundary and interior layers appear. For any fixed grid with a specified number of points there is always a small $\varepsilon > 0$ for which the numerical solution is a bad approximation. To obtain a better solution one may increase the number of points, but as it increases the complexity of the computation, this is something that we want to avoid. In this paper we describe a way of constructing composite grids, in which a-priori knowledge about the solution is used for the construction. This is realized with the help of so-called stretching functions. These functions contain parameters used to condense the grid lines in regions where the solution varies rapidly. For varying ε, the number of grid lines will be kept constant on all component grids in order to investigate if some practical ε-uniformity can be achieved.

Thus, the purpose of this paper is to study the behavior of the numerical solution for a range of parameters ε. We start with three component grids in order to fit the obvious features of the solution. Since, in principle, there are many free parameters in the numerical method, such as the priorities of the component grids (that should be specified by the user), we start with a simple case in order to make a first general selection and reduce the number of input parameters.

In this paper we only adapt the grids and we restrict ourselves to basic central difference discretisation. We refrain from stencil adaptation such as upwinding or exponential fitting. Other arguments, cf. [4, 13], show that it is unlikely that for the present non-trivial model problem such schemes could lead to essentially better results.

Our final composite grid will have a diffusion-parameter dependent grid spacing, and will take into account the direction of convection. The results will be calculated for the following values of the convection parameter: $\varepsilon = 0.01 \cdot 4^{-m}, m = -2, -1, 0, 1, 2$. Further we select two directions of convection: a direction aligned with the x-axes (i.e., aligned with the main grid, $\alpha = 0°$) and a direction at an arbitrary corner with the x-axes, for which we choose, $\alpha = 18°$.

2. The Mathematical Problem

In this paper we concentrate on the two-dimensional convection-diffusion equation [3],

$$\varepsilon \Delta u - \mathbf{v} \cdot \nabla u = 0, \quad \varepsilon > 0, \tag{1}$$

which is defined on the exterior of the unit disc, i.e. on $\Omega = \{(x, y) \in \mathbb{R}^2 | x^2 + y^2 \geq 1\}$, with boundary conditions:

$$\begin{aligned} u &= 1, \quad \text{if} \quad x^2 + y^2 = 1, \\ u &\to 0, \quad \text{for} \quad x^2 + y^2 \to \infty. \end{aligned} \tag{2}$$

The convection in the direction $\mathbf{v} = (\cos(\alpha), \sin(\alpha))$ makes the problem symmetric with respect to a line through the origin, at an angle α with the x-axis. Further the shape of the solution is independent of α.

Obvious properties of the solution are its formal smoothness, its monotonicity, and the fact that for smaller values of ε a sharp boundary layer appears at the upwind side of the circle, and a long 'shadow region' appears at the downwind side. In the limit there is a discontinuity between the shadow (where the limit solution $u = 1$), and the 'exposed part' of the solution (where $u \to 0$). At the boundary of the shadow region an interior layer appears.

3. The Discrete Problem

To numerically solve the problem by a discretization method, the infinite domain must be truncated to a finite region of interest. On the one hand the region should be large enough to contain the specific details of the solution and on the other hand not too large, to reduce the amount of computational work. To make the decision, our a-priori knowledge about the analytic solution is used.

3.1. Choice of the Finite Domain and Boundary Conditions

Because for $\alpha = 0$ equation (1) describes convection to the right, we select the finite domain

$$\Omega = [-3, 9] \times [-3, 3] \cap \{(x, y) | x^2 + y^2 \geq 1\}. \tag{3}$$

In this way the domain can show a significant part of the solution: the boundary and interior layers, and the shadow region of the solution.

Although on this finite domain the problem is not symmetric with respect to an arbitrary α, we use this fixed domain also for the case $\alpha \neq 0$. For the angles α we treat, the domain Ω is still the domain of interest for the solution of (1). For $\alpha \neq 0$ the typical features of the solution which do not align with the obvious grid, may cause numerical difficulties.

For the truncated, finite domain Ω we have to introduce artificial boundary conditions at the outer boundary. We choose these also according to our a-priori knowledge about the problem. We apply a homogeneous Neumann boundary condition $\mathbf{v} \cdot \nabla u(x,y) = 0$ at the outflow boundary $\partial\Omega$, where $\mathbf{v} \cdot \mathbf{n} > 0$, with $\mathbf{n}$ the outward pointing normal vector at $\partial\Omega$. Where $\mathbf{v} \cdot \mathbf{n} \leq 0$ we apply homogeneous Dirichlet boundary conditions $u = 0$. At the unit circle we always apply the Dirichlet boundary condition (2).

3.2. Over-Set Grids

Here we first give some definitions of the terms we need in relation with overset grids. We define the *open physical domain*

$$\Omega_o = (-3,9) \times (-3,3) \cap \{(x,y) | x^2 + y^2 > 1\}. \tag{4}$$

We cover the closure of this domain, $\Omega = \bar{\Omega}_o$, by a number of K (boundary-fitted) *component domains*, $\Omega^{(k)}$, so that $\bigcup_{k=1}^{K} \Omega^{(k)} \supset \Omega$, where we take care that different component domains have sufficient overlap. Some of the domains $\Omega^{(k)}$ are placed so that each part of the boundary of our domain of interest, $\partial\Omega = \Omega \backslash \Omega_o$, is covered by the boundary of some component domain. Other component domains may be placed so that they are in some sense (hopefully) aligned with features of the expected solution.

With each component domain $\Omega^{(k)}$ is related a smooth injective mapping $\mathcal{M}^{(k)} : [0,1]^2 \to \Omega^{(k)}$ from the unit square onto $\Omega^{(k)}$, and for each $\Omega^{(k)}$ a regular rectangular grid $\mathcal{E}^{(k)}$ is chosen on $[0,1]^2$, so that

$$\mathcal{E}^{(k)} = \left\{ (i/N^{(k)}, j/M^{(k)}) \middle| i = 0, \ldots, N^{(k)}, j = 0, \ldots, M^{(k)} \right\}. \tag{5}$$

Thus, by means of the mappings $\mathcal{M}^{(k)}$ we generate (topologically) regular $N^{(k)} \times M^{(k)}$ grids, $\mathcal{G}^{(k)} := \mathcal{M}(\mathcal{E}^{(k)}) \subset \Omega^{(k)} \subset \Omega$ in physical space. We say $\mathcal{E}^{(k)}$ lives in computational, and $\mathcal{G}^{(k)}$ in physical space.

The mappings $\mathcal{M}^{(k)}$ are most conveniently described by decomposing them into two mappings, by $\mathcal{M}^{(k)} = T^{(k)} \circ R^{(k)}$ where $R^{(k)} : [0,1]^2 \to [0,1]^2$ is a mapping that redistributes the gridlines over the unit square, and $T^{(k)}$ maps the unit square into the physical space. The mapping $R^{(k)}$, being responsible for the distribution of the gridlines, is taken of the form $R^{(k)}(x,y) = (R_1^{(k)}(x), R_2^{(k)}(y))$, where each

$R_i^{(k)} : [0,1] \to [0,1], i = 1,2$, is a continuous invertible function such that $R_i^{(k)}(0) = 0$ and $R_i^{(k)}(1) = 1$. These functions $R_i^{(k)}$ are called *stretching functions*, and the resulting

$$\mathcal{F}^{(k)} = \{(R_1^{(k)}(r), R_2^{(k)}(s)) | (r,s) \in \mathcal{E}^{(k)}\} \tag{6}$$

is a regular rectangular, stretched grid on the unit square. Obviously, if stretching is not needed in some direction, we take $R_i^{(k)}(x) = x$.

The component mapping $T^{(k)}$ is in general a (simple) smooth injection from the unit square into physical space:

$$T^{(k)} : [0,1]^2 \to \Omega^{(k)} \subset \mathbb{R}^2. \tag{7}$$

Figure 2 illustrates the above mappings, where $(r,s) \in \mathcal{E}^{(k)}$. The variables $(t,u) = ((R_1^{(k)}(r), (R_2^{(k)}(s)) \in \mathcal{F}^{(k)}$ note the stretched coordinates on the unit square and $(x,y) = T^{(k)}(t,u)$ denotes the physical coordinates.

The restriction of the *component mapping* $\mathcal{M}^{(k)}$ to the grid $\mathcal{E}^{(k)}$ is called the *component grid mapping* $\mathcal{M}_h^{(k)}$, and $\mathcal{M}_h^{(k)}(\mathcal{E}^{(k)}) := \mathcal{G}^{(k)}$ is the kth *component grid* in Ω. The *composite grid* is the union of all component grids:

$$\mathcal{G} = \bigcup_{k=1}^{K} \mathcal{G}^{(k)} = \bigcup_{k=1}^{K} (\mathcal{M}_h^{(k)}(\mathcal{E}^{(k)})).$$

We notice that in practical applications the mapping $\mathcal{M}^{(k)}$ may be unknown, but the mapping $\mathcal{M}_h^{(k)}$ should be available for the computation, possibly by means of a table.

By the choice of the component domains $\Omega^{(k)}$, the mappings $\mathcal{M}^{(k)}$ and the grids $\mathcal{E}^{(k)}$, it may appear that large parts of the physical domain are covered with large parts of overlapping $\Omega^{(k)}$. Computation on all these overlapping grids might result in a large, superfluous computational effort. Therefore we allow an *overlap algorithm* to cut away parts of the mutually overlapping component grids. Component grids, with overlapping parts cut out, we still call *component grids*.

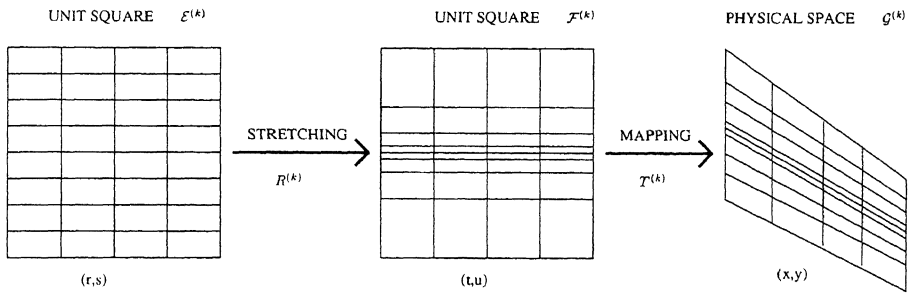

Figure 2. The decomposition of the mapping $\mathcal{M}^{(k)} = T^{(k)} \circ R^{(k)}$; the unit square with uniform spacing, the unit square with stretched grid lines and the component grid

To make a choice which parts of the overlapping component grids have to be cut, the overlap algorithm needs a priority ranking of the component grids. This ranking should be specified by the user. Based on this information and the geometry, the overlap algorithm determines what parts of $\mathcal{G}$ will be considered superfluous, and it distinguishes between two types of remaining grid points: *interpolation points*, for which the value is obtained by interpolation from another grid, and *discretization points*, for which the value can be determined by the discretized PDE (i.e., a point for which a discretization stencil should be constructed). In this way a different specification of the priority between the various component grids may essentially determine the shape (and the quality) of the final composite grid. Compare Figs. 1 and 3 showing the influence of the priority ranking.

The details of a real overlap algorithm are quite technical and we only mention some main features. The most recent overlap algorithm known to us, is implemented in Ogen [8], which is part of the Overture framework, see Section 4. In fact it is an improvement of an earlier algorithm that was implemented in the FORTRAN-code CMPGRD. In [2] detailed information about this overlap algorithm can be found. A similar algorithm, with a few nice additional features, is described in [15] and implemented in the C-code Xcog [14].

The overlap algorithm automatically constructs an optimal overlapping (composite) grid and classifies each point on the user-defined component grid. The points that are cut away from the component grid are also called *hole points* or *void points*, so that all points in the grids are classified as (i) discretization point, (ii) interpolation point, or (iii) void point. Interpolation points are further characterized by the component grid $\mathcal{G}^{(k)}$ from which they receive the interpolation information. The overlap algorithm will remove excessive interpolation points and optimize for the number of void points.

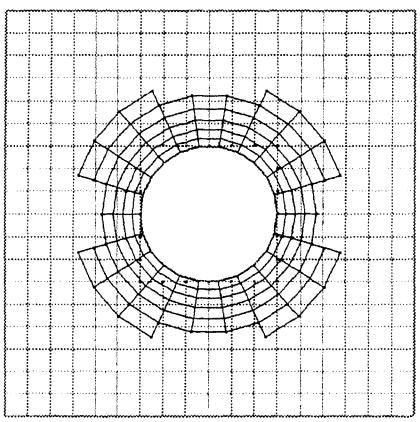

Figure 3. The square grid has the higher priority in this composite grid. This is in contrast with Fig. 1, where the annular grid has the higher priority

In order to improve on the flexibility of the construction, we also can allow a component grid to contain points that lie outside Ω, provided that each part of the boundary $\partial\Omega$ is captured by a gridline of a component grid.

Now, two interpolation techniques can be distinguished: (i) explicit interpolation, where the interpolation stencil for one grid contains only discretization points from the other grid (the *donor grid*), and (ii) implicit interpolation, where the interpolation stencil may contain both interpolation and discretization points from the donor grid.

At a discretization point one should construct the required discretization stencil for this point on its component grid. This stencil may contain other discretization points, (e.g., boundary points) or interpolation points. For the computations reported in this paper, we have chosen to use explicit interpolation only, which means that more overlap is required compared with the amount of overlap needed for implicit interpolation.

The *composite grid*, in a slightly more general sense than used above, is the collection of component grids, together with the information about the character of each grid point. Void points are not drawn in our figures. Thus, the resulting composite grid clearly depends on the priorities specified for the component grids and the width of the (interpolation and discretization) stencils.

4. Overture

Overture is a software system for the solution of two and three dimensional PDEs on complex domains by the use of over-set grid methods. It is under development at LLNL. It contains a comprehensive object-oriented software library written in C++ [1, 5]. The software, its documentation, a tutorial, etc. is freely available from the Overture web page http://www.llnl.gov/casc/Overture. The software system contains a number of major modules: an overlapping grid generator (Ogen) [8], a solver for PDEs on overlapping grids (Oges) [9], a view facility (plotStuff) [11] and much more. It is possible to use finite-difference or finite-volume methods in an object-oriented way [6]. In the documentation [12] clear examples are shown how Overture can be used for solving PDEs.

Our present experience is with Overture version 15. The results in this paper are obtained by using Ogen, Oges plotStuff and the show-file class [10], which is a data format developed for visualization. The composite grid data structure (output of the grid generator) is stored in the Hierarchical Data Format (HDF), which is an efficient standard for sharing scientific data. See http://hdf.ncsa.uiuc.edu.

5. The Use of Asymptotic Properties for Grid Generation

Some a-priori knowledge about the solution will be used for the selection of the component grids and for the construction of the stretching parameters needed to properly place the grid lines.

The problem (1)–(2) is a typical example of a singularly perturbed problem. For small values of the parameter ε we distinguish sharp layers in the solution [3]. The three main solution features are: the outer solution as approximated by the reduced equation, a boundary layer at the upwind side of the circle and two interior layers at the edge of the shadow. The outer solution is approximated by the solution of $\mathbf{v} \cdot \nabla u = 0$. It contains the 'shadow' part ($u \approx 1$) downwind from the circle and an 'exposed' part ($u \approx 0$) in the remainder of the domain.

At the upwind side of the circle ($x \leq -1$ if $\alpha = 0°$) the solution is approximated by $u(x, 0) = e^{(x+1)/\varepsilon}$, i.e., there the boundary layer is $\mathcal{O}(\varepsilon)$. The transition between the shadow and the exposed part of the solution runs in the $\mathbf{v}$ direction. It is described by a parabolic internal layer with an $\mathcal{O}(\varepsilon^{1/2})$ thickness, cf. [3]. We will handle this transition by a simple Cartesian grid.

To apply the over-set grids we first use a simple strategy, restricting ourself to three simple component grids corresponding with the three major features in the solution. Thus, we cover Ω with the following three component domains:

1. A *background domain*: $\{(x,y) \in \mathbb{R}^2 | (x,y) \in [-3,9] \times [-3,3]\}$. All boundaries of this domain are boundary-fitted and correspond with the outer boundary of the physical domain.
2. An *annulus*: $\{(x,y) \in \mathbb{R}^2 | 1 \leq x^2 + y^2 \leq 2\}$, with the inner side fitted to the boundary of the circle in the physical domain.
3. A *strip*: $\{(x,y) \in \mathbb{R}^2 | (x,y) \in [0,9] \times [-2,2]\}$ for $\alpha = 0$. For $\alpha \neq 0$ the strip is rotated around (0,0) over an angle α.

For the background grid we take a rectangular grid with a fixed mesh width, independent of the parameter ε. For the annular grid in one direction a periodic boundary condition is introduced. In the other direction we condense the grid lines in the radial direction with a maximum at the edge of the unit disk, where a mesh-width $\mathcal{O}(\varepsilon)$ is used. The third component grid, the strip, should take care of the interior layers. For this component grid the grid lines will be condensed with two clusters at the interior layers of the solution. Here we make the mesh-width of $\mathcal{O}(1)$ along the layer and $\mathcal{O}(\varepsilon^{1/2})$ perpendicular to the layer, as described in more detail below.

The function $R_i^{(k)}(x)$, used to control the grid refinement in the composite $\mathcal{G}^{(k)}$, is the 'inverse hyperbolic tangent stretching function' as introduced in [7]. It is a one-dimensional mapping $[0,1] \to [0,1]$ that is best described by its inverse function $\bar{R}_i^{(k)} : [0,1] \to [0,1]$ so that $R_i^{(k)}\left(\bar{R}_i^{(k)}(x)\right) = x$. The function $\bar{R}_i^{(k)}$ is of a simple nature, in which one can add a finite number of $m_i^{(k)}$ so-called *layer functions* $U_{i,j}^{(k)}$ to describe grid condensation in some neighborhood, whenever needed. As before, i denotes the axis and k the k-th component of the composite grid. We here choose the function $\bar{R}_i^{(k)}(x)$ to be of the form

$$\bar{R}_i^{(k)}(x) = \frac{x + \sum_{j=1}^{m_i^{(k)}} [U_{i,j}^{(k)}(x) - U_{i,j}^{(k)}(0)]}{1 + \sum_{j=1}^{m_i^{(k)}} [U_{i,j}^{(k)}(1) - U_{i,j}^{(k)}(0)]}, \tag{8}$$

with the layer function $U_{i,j}^{(k)}$ defined by

$$U_{i,j}^{(k)}(x) = \frac{1}{2} a_{i,j}^{(k)} \tanh[b_{i,j}^{(k)}(x - c_{i,j}^{(k)})]. \tag{9}$$

We notice that (8) satisfies the conditions of Theorem 1 in [16] and therefore would lead to uniform convergence for a simple one-dimensional model problem. We also see that, for $m_i^{(k)} = 0, i = 1, 2$, no stretching takes place and $\mathscr{F}^{(k)}$ will be a regular rectangular grid. For $m_{i,j}^{(k)} \neq 0$, local refinement takes place in the x_i-direction near $c_{i,j}^{(k)}$. The parameter $b_{i,j}^{(k)}$ determines the slope and $a_{i,j}^{(k)}$ the amount of the grid refinement. Thus $U_{i,j}^{(k)}$ determines each local condensation in a particular area by three parameters.

The parameter $b_{i,j}^{(k)}$ is proportional with the maximal derivative of the stretching function and hence determines the minimal grid spacing. The parameter $a_{i,j}^{(k)}$ is proportional with the number of grid lines that are stretched. In Table 1 we show the parameters used for a first computation with $\alpha = 0$. By the symmetry of the problem we can write $a_{i,0}^{(k)} = a_{i,1}^{(k)} := a_i^{(k)}$ and $b_{i,0}^{(k)} = b_{i,1}^{(k)} := b_i^{(k)}$. The parameter N determines the number of grid lines in each direction and will be fixed to a constant so that the number of grid lines will be *independent* of ε. I.e., a strategy is followed that is comparable with the one leading to Shishkin meshes. For each stretching we take the number of grid lines inside and outside the layer of the same order, so that–in this sense–an essential property of a Shishkin mesh [4] is mimicked. In the calculations shown we take $N = 5$. Further, the parameters are chosen such that, where possible, the grid spacing at overlapping regions is of equal order. On all grids central discretization is used, and bi-quadratic interpolation is used, which means that both discretization and interpolation stencils are 3×3 points wide. Thus the overall accuracy of the method is second order.

Overture constructs the composite grid by calling the grid generator 'Ogen' and solves the problem by using 'Oges'. The solution is stored in a 'show-file' [10]. Also the graphical results in this paper are produced by using this system. As a measure to judge the quality of the solution, the minimum and maximum function values (undershoot and overshoot) of the computed solution are determined (using the adage 'Don't kill the wiggle, it is telling you something').

6. First Results

As a first trial we calculated the solution for all six possible priority rankings of the three component grids. For 'large' parameter values, like $\varepsilon = 0.16$ these pri-

Table 1. Parameters used for stretching functions

Grid k	Axis i	Lines $N_i^{(k)}$	Layers $m_i^{(k)}$	$a_i^{(k)}$	$b_i^{(k)}$	$c_{i,0}^{(k)}$	$c_{i,1}^{(k)}$
0	0	$12N + 1$	0	–	–	–	–
0	1	$6N + 1$	0	–	–	–	–
1	0	$9N + 1$	0	–	–	–	–
1	1	$2N$	1	2	$\frac{0.16}{\varepsilon}$	0	–
2	0	$9N + 1$	0	–	–	–	–
2	1	$6N + 1$	2	$\frac{1}{4}$	$\frac{1.6}{\sqrt{\varepsilon}}$	$\frac{1}{4}$	$\frac{3}{4}$

orities are less relevant, because the solution is smooth and all component grids are reasonably well able to represent the solution. But for smaller values of ε the influence of the priority ranking becomes more clear. From our experiments we conclude that the best results are obtained if we give the background grid a lower priority than the annular grid. In all cases where this has not been done, significant numerical errors arise at the upwind side of the circle, a peak in the neighborhood of $(-1, 0)$ appears since, in that case, grid points of the annular grid are removed by the overlap algorithm. Having determined this, three possibilities remain. The best result follows if the priority of the annular grid is between the background and the 'strip'. This means that the strip should have highest priority. The criterion used is minimization of $|u_{\max} - 1| + |u_{\min}|$, where $u_{\min}$ and $u_{\max}$ denote the minimum (maximum) value of the computed solution over all grid points. Although our decision about the priority ranking is based on the $\varepsilon = 0.01$ case, we see that the highest priority of the 'strip' is consistent with the expected behavior of the solution for smaller values of ε. If the 'strip' were given a priority higher than the background grid, the overlap algorithm would remove almost all significant grid points. In this way we are also able to catch features of the $\alpha \neq 0$ case by rotating the strip.

With our priority ranking, the solution is calculated for $\alpha = 0$ and for our range of diffusion parameters. The composite grid used is shown in Fig. 4 and the results are shown in Table 2. The case $\alpha \neq 0$, i.e., rotation of the strip is presented in Section 7.

Figure 5 shows the two computed solutions $z = u(x, y)$, for $\varepsilon = 0.01$ and for $\varepsilon = 0.0025$. The solution for $\varepsilon = 0.01$ is smooth except for two small regions in the shadow of the circle where spurious peaks appear. For $\varepsilon = 0.0025$ these peaks become higher and other errors appear in the strip region. For $\varepsilon = 0.000625$ the solution becomes worse. In this case large peaks appear periodically at a distance of twice the cell width, as is typical for central discretization. Thus we see that numerical problems appear even if we catch the asymptotic behavior in the boundary and interior layers. For large values of ε (if $\varepsilon \geq 0.4$) the peaks are not present and the solution is smooth over the whole domain, as expected.

7. Difficulties and Remedy

If we study the above composite grid in more detail for small ε, we see that the interpolation points in the strip interpolate from points of the annular grid, behind the circle but relatively far from the circle. Since most annular grid lines are close to the circle, the strip interpolates from the coarse part of the annulus. This can be improved if we decrease the stretching of the annulus in that region. Indeed, we have found that the solution at the downwind region of the annular grid improves: the peaks disappear if the stretching parameter becomes smaller. The price to pay is that another peak arises at the upwind side of the circle (as expected for a coarser grid), because there a grid condensation is required corresponding with the $\mathcal{O}(\varepsilon)$ behavior of the boundary layer in the solution.

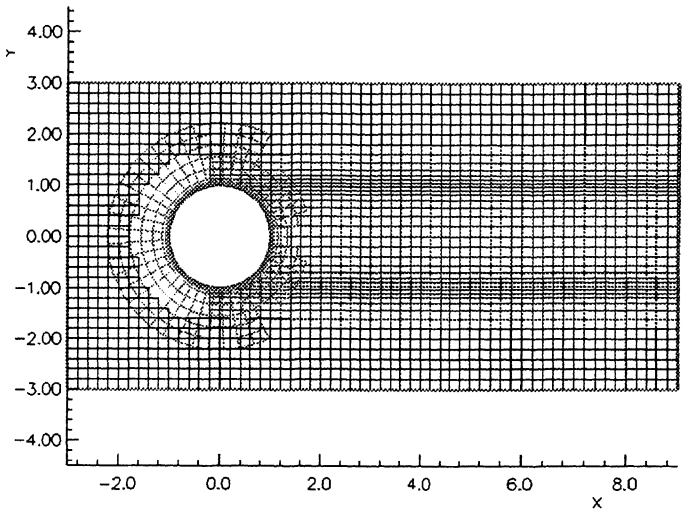

Figure 4. The composite grid with the priority ranking: 0: background grid; 1: annulus; 2: strip, constructed for $\varepsilon = 0.01$

Table 2. Minimum and maximum values for the computed solution, obtained with the best priority ranking: 0: background grid; 1: annulus; 2: strip (highest priority)

ε	$u_{\min}$	$u_{\max}$
0.16	0.0000000	1.0000000
0.04	−0.0008200	1.0000000
0.01	−0.0035112	1.1419837
0.0025	−0.0837866	1.2844765
0.000625	−1.0026414	1.7415791

The remedy is to divide the annular grid in two parts, one in the upstream and the other in the downstream direction. Technically we did this by adding a *half annular* grid at the upwind side of the circle. The overlap algorithm removes a part of the underlying *full annular* grid due to the given priorities. We will give the part at the upwind side of the circle a smaller grid spacing in the radial direction than the radial grid spacing at the downwind side of the circle, which is handled by the full annular grid. Corresponding with the asymptotic properties the latter will get $\mathcal{O}(\varepsilon^{2/3})$ thickness. Furthermore, taking into account more detailed asymptotics [3], an improvement will also follow from stretching all grids, except the background grid, in the direction *along* the layers. For this we introduce an $\mathcal{O}(\varepsilon^{1/3})$ behavior, which is also obtained from [3].

7.1. The New Patch

The new part, the *half annulus*, with a priority higher than the full annular grid, but lower than the 'strip' lies in the following region:

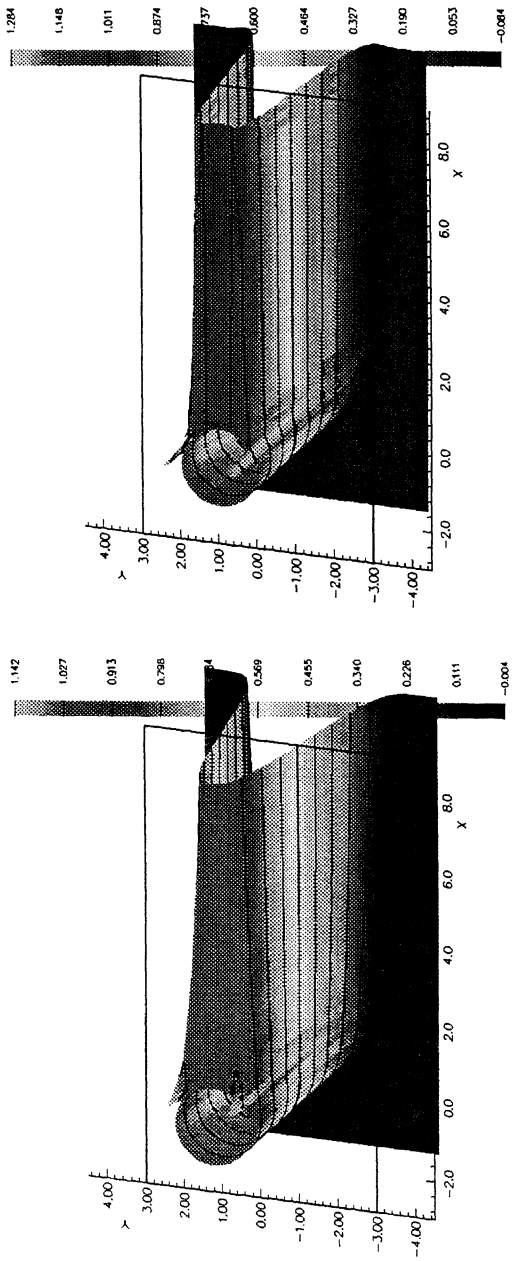

Figure 5. The solution for $\varepsilon = 0.01$ and $\varepsilon = 0.0025$ for the composite grid consisting of three components

$$\begin{cases} \alpha = 0: & \{(x,y) \in \mathbb{R}^2 | 1 \leq x^2 + y^2 \leq 2, x \geq 0\} \\ \alpha \neq 0: & \text{The same domain rotated around } (0,0) \text{ over an angle } \alpha. \end{cases}$$

To determine the parameters, we use Table 3, which shows the derivatives of the stretching function $\bar{R}_i^{(k)}(x)$ for large values of $b_i^{(k)}$ (i.e., the mesh-density for small values of ε). This determines the asymptotic mesh width in the inner and outer region. At the points of maximum clustering we couple $\bar{R}_i^{(k)}$ to the asymptotic behavior of the solution in the following way:

$$\bar{R}_i'^{(k)}(c_{i,0}^{(k)}) = \bar{R}_i'^{(k)}(c_{i,1}^{(k)}) = \frac{K_i^{(k)}}{\varepsilon^{\nu_{i,k}}}. \tag{10}$$

By this the relation between ε and b is determined. The constant $K_i^{(k)}$ denotes a chosen ratio between the maximum slope of the stretching function and the asymptotic behavior of the solution, which requires a layer thickness $\mathcal{O}(\varepsilon^{\nu_{i,k}})$.

In Section 5 we took $\nu_{i,k} = 1$ for the annular grid and a value $\frac{1}{2}$ for the 'strip'. Corresponding with the additional a-priori knowledge about the asymptotics that we want to use, we now choose some new parameters corresponding with the details shown in Fig. 6. Table 4 summarizes all parameters of the improved composite grid in order of the priority ranking. The rotation of the full annular grid is performed by changing the location of the two layers.

For the constants $K_i^{(k)}$ we take $K_0^{(1)} = 1$, $K_1^{(1)} = 0.005$, $K_0^{(2)} = 1$, $K_1^{(2)} = 0.05$, $K_0^{(3)} = 1$, $K_1^{(3)} = 0.1$. To prevent the value of b becoming too small (or even negative) for larger values of ε, we replace b by $\max(b, 1)$, which is allowed since in that case the precise stretching of the grid becomes irrelevant. Figure 7 gives the resulting composite grid for the case $\alpha = 0°$. As can be seen from this figure, only the background grid takes the outflow boundary conditions and the 'strip' interpolates from this grid and vice versa. Since the 'strip' will be rotated it is not possible to fit this grid with the outflow boundaries.

8. Results and Conclusion

Table 5 shows the results obtained with our composite grid. If we compare these results with those obtained with the previous grid (Table 2) we see they have much improved. There is less overshoot as we can see in the table or in Fig. 8. For $\varepsilon = 0.01$ the peaks almost completely disappeared. However, for smaller values of ε the results become less accurate again. For $\varepsilon = 0.000625$, for instance, several peaks appear in the shadow region of the solution, although the result is

Table 3. The derivatives of the function $\bar{R}'$, at the relevant points

Location of layer(s)	Derivative at layer(s)	Derivative far from layer(s)
$c = 0$	$\bar{R}'(0) = \frac{1+\frac{a}{2}}{1+\frac{a}{2}}$	$\bar{R}'(1) = \frac{1}{1+\frac{a}{2}}$
$c_1 = \frac{1}{4}, c_2 = \frac{3}{4}$	$\bar{R}'(c_{1,2}) = \frac{1+\frac{a}{2}}{1+2a}$	$\bar{R}'(0) = \bar{R}'(\frac{1}{2}) = \bar{R}'(1) = \frac{1}{1+2a}$
$c_1 = 0, c_2 = 1$	$\bar{R}'(0) = \bar{R}'(1) = \frac{1+\frac{a}{2}}{1+a}$	$\bar{R}'(\frac{1}{2}) = \frac{1}{1+a}$

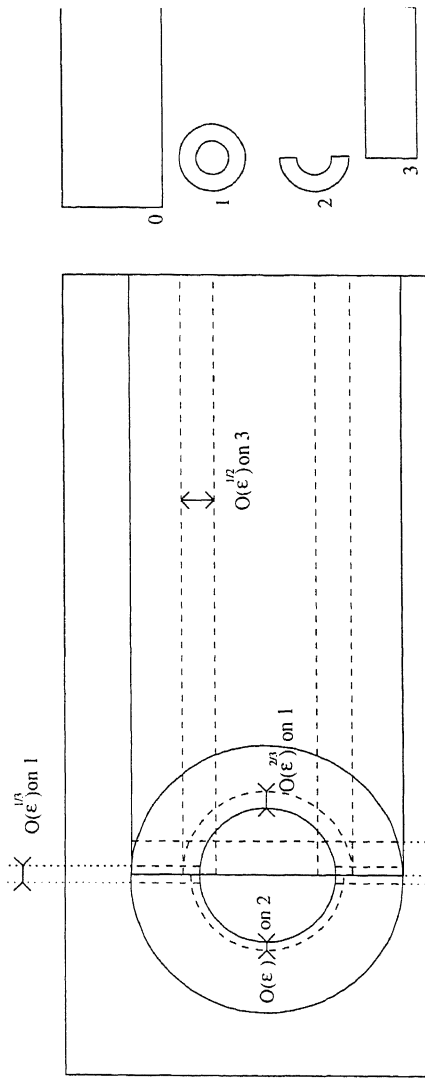

Figure 6. The layer thickness of each component grid is given in each direction. At the right the priorities of the various components are shown

Table 4. A summary of the parameters, here 0 is the background grid, 1 the full annular grid, 2 the half annular grid and 3 the 'strip'. For $\alpha \neq 0$ grids 1, 2 and 3 are rotated

Grid k	Axis i	Lines $N_i^{(k)}$	Layers $m_i^{(k)}$	$a_i^{(k)}$	$b_i^{(k)}$	$c_{i,0}^{(k)}$	$c_{i,1}^{(k)}$
0	0	$12N+1$	0	–	–	–	–
0	1	$6N+1$	0	–	–	–	–
1	0	$18N+1$	2	$\frac{1}{4}$	$\frac{K_0^{(1)}}{\varepsilon^{1/3}}\left(\frac{2+4a_0^{(1)}}{a_0^{(1)}}\right)-\frac{2}{a_0^{(1)}}$	$\frac{1}{4}$	$\frac{3}{4}$
1	1	$2N+1$	1	1	$\frac{K_1^{(1)}}{\varepsilon^{2/3}}\left(\frac{2+a_1^{(1)}}{a_1^{(1)}}\right)-\frac{2}{a_1^{(1)}}$	0	–
2	0	$12N+1$	2	$\frac{1}{4}$	$\frac{K_0^{(2)}}{\varepsilon^{1/3}}\left(\frac{2+2a_0^{(2)}}{a_0^{(2)}}\right)-\frac{2}{a_0^{(2)}}$	0	1
2	1	$2N+1$	1	1	$\frac{K_1^{(2)}}{\varepsilon}\left(\frac{2+a_1^{(2)}}{a_1^{(2)}}\right)-\frac{2}{a_1^{(2)}}$	0	–
3	0	$9N+1$	1	1	$\frac{K_0^{(3)}}{\varepsilon^{1/3}}\left(\frac{2+a_0^{(3)}}{a_0^{(3)}}\right)-\frac{2}{a_0^{(3)}}$	0	–
3	1	$6N+1$	2	$\frac{1}{4}$	$\frac{K_1^{(3)}}{\sqrt{\varepsilon}}\left(\frac{2+4a_1^{(3)}}{a_1^{(3)}}\right)-\frac{2}{a_1^{(3)}}$	$\frac{1}{4}$	$\frac{3}{4}$

much better than would be possible with a composite grid consisting of three simple components. We note that, for securing stability, a first-order Neumann boundary condition is applied at the right side of the grid (i.e., at $x = 9$) instead of the default second order accuracy Overture uses. Figure 9 shows the skew grid and the corresponding solution obtained. Since the standard second-order Neumann boundary condition is applied in this case, this gives rise to errors at the boundary as can be seen in this figure. For a first-order Neumann boundary condition this error will not occur, and the calculated solution will show a smooth outflow as we have seen for the $\alpha = 0°$ case (where the first-order condition was applied).

Nevertheless, these experiments clearly show that for our non-trivial model problem we do not get a numerical method that is ε-uniformly accurate with respect to the number of mesh points, even if we make use of more advanced asymptotics. With the solution method as described in this paper one can never get a perfect composite grid for values of ε that can become arbitrarily small. Because for any choice of components, the use of stretching for adapting each individual grid to the local asymptotic behavior without increasing the number of grid points, will always imply that next to overlapping regions with roughly equal grid spacing there will exist overlapping regions with grid spacing that differ in order of magnitude.

Table 5. Minimum and maximum values for the computed solution ($\alpha = 0$), obtained with the priority ranking shown in Fig. 6

ε	$u_{\min}$	$u_{\max}$
0.16	0.0000000	1.0000001
0.04	−0.0073076	1.0000011
0.01	−0.0060746	1.0063686
0.0025	−0.0074955	1.0435394
0.000625	−0.0226137	1.1181593

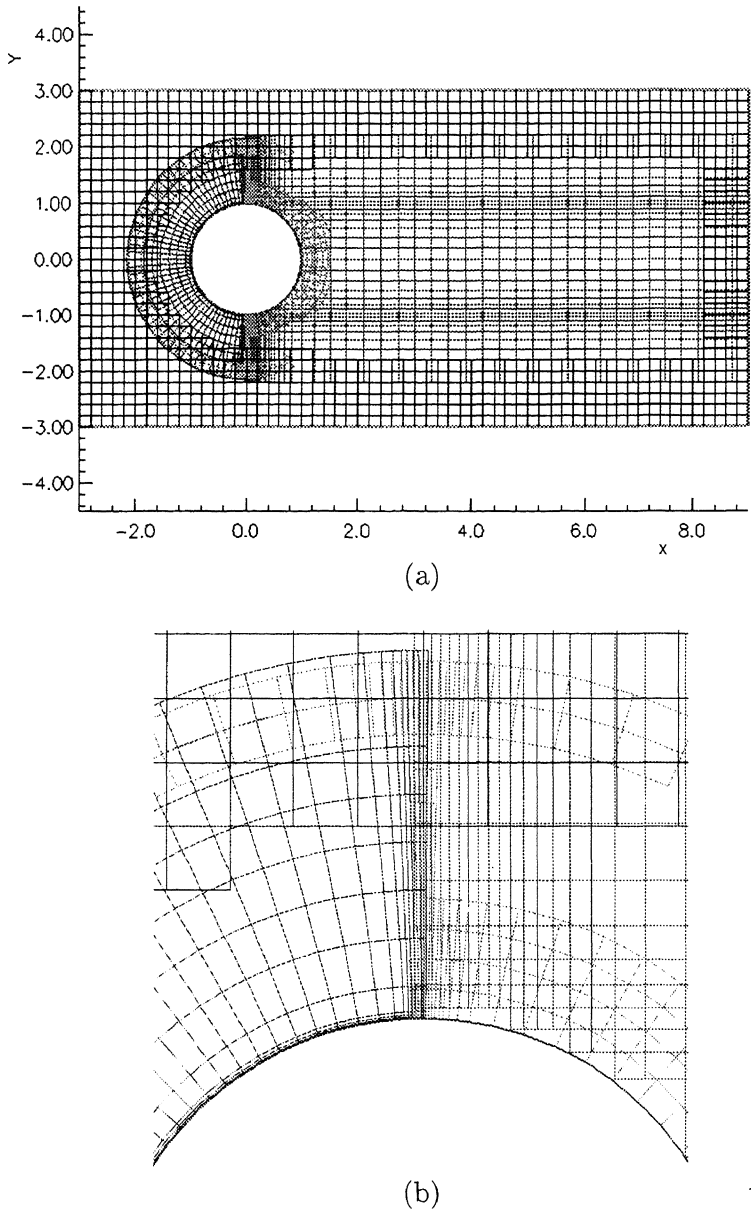

Figure 7. The resulting composite grid constructed for $\varepsilon = 0.0025$; **a** global view, **b** detail. Priority ranking as in Fig. 6 is used

Thus, it remains an open question whether it is possible to find ε-uniformly accurate methods for this model problem, and – seeing that the use of even detailed a-priori knowledge does not yield such result – one might raise the question if it makes sense pursuing such direction rather than using non-ε-uniform, but efficient, self-adaptive solution strategies.

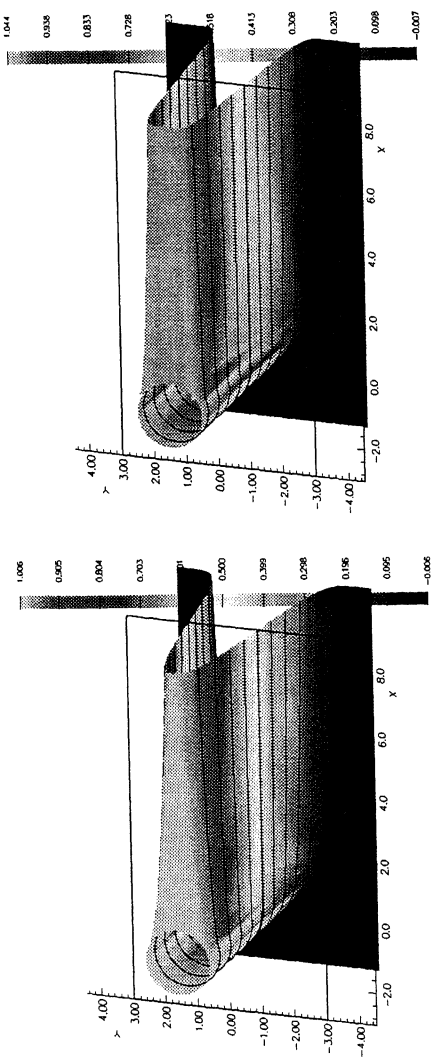

Figure 8. The solution for $\varepsilon = 0.01$ and $\varepsilon = 0.0025$. First-order accurate Neumann boundary condition applied to the right boundary

Figure 9. The rotated grid and the solution for $\varepsilon = 0.01$. A second-order accurate Neumann boundary condition applied at the outflow boundaries

Acknowledgement

We thank Bill Henshaw for helping us with some technical details of Overture.

References

[1] Brown, D. L., Chesshire, G. S., Henshaw, W. D., Quinlan, D. J.: OVERTURE: an object-oriented software system for solving partial differential equations in serial and parallel environments. Technical Report LA-UR-97-335, Los Alamos National Laboratory, 1997.
[2] Chesshire, G., Henshaw, W. D.: Composite overlapping meshes for the solution of partial differential equations. J. Comp. Phys. *90*, 1–64 (1990).
[3] Hemker, P. W.: A singularly perturbed model problem for numerical computation. J. Comp. Appl. Math. *76*, 277–285 (1996).
[4] Hemker, P. W., Shishkin, G. I.: On a class of singularly perturbed boundary value problems for which an adaptive mesh technique is necessary. In: Proceedings of the Second International Colloquium on Numerical Analysis (Bainov, D., Covachev, V., eds.), pp. 83–92. VSP, 1994.
[5] Henshaw, W. D.: Overture: an object-oriented system for solving PDEs in moving geometries on overlapping grids using C++. In: First AFOSR conference on dynamic motion CFD (Sakell, L., Knight, D. D., eds.) pp. 281–290, 1996.
[6] Henshaw, W. D.: Finite difference operators and boundary conditions for Overture, user guide, version 1.00. Technical Report LA-UR-96-3467, Los Alamos National Laboratory, 1998.
[7] Henshaw, W. D.: Mappings for Overture, A description of the mapping class and documentation for many useful mappings. Technical Report LA-UR-96-3469, Los Alamos National Laboratory, 1998.
[8] Henshaw, W. D.: Ogen: An overlapping grid generator for overture. Technical Report LA-UR-96-3466, Los Alamos National Laboratory, 1998.
[9] Henshaw, W. D.: Oges User guide, version 1.00, A solver for steady state boundary value problems on overlapping grids. Technical Report LA-UR-96-3468, Los Alamos National Laboratory, 1998.
[10] Henshaw, W. D.: Ogshow: Overlapping grid show file class, saving solutions to be displayed with plotstuff showfilereader: A class for reading solutions from a show file, user guide version 1.00. Technical Report LA-UR-96-3465, Los Alamos National Laboratory, 1998.
[11] Henshaw, W. D.: Plotstuff: A class for plotting stuff from Overture based on: GLGraphicsInterface: a graphics interface based on OpenGL based on: GenericGraphicsInterface: a generic graphics interface user guide, version 1.00. Technical Report LA-UR-96-3893, Los Alamos National Laboratory, 1998.
[12] Henshaw, W. D.: A primer for writing PDE solvers with overture. Technical Report LA-UR-96-3894, Los Alamos National Laboratory, 1998.
[13] Miller, J. J. H., O'Riordan, E., Shishkin, G. I.: Numerical methods for singular perturbation problems: error estimates in the maximum norm for linear problems in one and two dimensions. Singapore: World Scientific, 1996.
[14] Petersson, N. A.: User's guide to Xcog 2.2. Technical Report CHA/NAV/R-97/0048. Chalmers Univ. of Technology, 1997.
[15] Petersson, N. A.: An algorithm for assembling overlapping grid systems. SIAM J. Sci. Comput. *20*, 1995–2022 (1999).
[16] Roos, H-G., Linss, T.: Sufficient conditions for uniform convergence on layer adapted grids. Computing, *63*, 27–45 (1999).

E. D. Havik and W. Hoffmann
University of Amsterdam
Korteweg-de Vries Institute for Mathematics
Plantage Muidergracht 24
NL-1018 TV Amsterdam
The Netherlands
e-mail: walter@science.uva.nl

P. W. Hemker
Centrum voor Wiskunde en Informatica
P.O. Box 94079
NL-1090 GB Amsterdam
The Netherlands
e-mail: P.W.Hemker@cwi.nl