A SCHEME FOR AUTOMATIC GENERATION OF BOUNDARY-FITTED DEPTH- AND DEPTH-GRADIENT- DEPENDENT GRIDS IN ARBITRARY TWO-DIMENSIONAL REGIONS

PER NIELSEN AND OVE SKOVGAARD

Laboratory of Applied Mathematical Physics, Technical University of Denmark, Building 303, DK-2800 Lyngby, Denmark

SUMMARY

In this paper we present a scheme for the numerical generation of boundary-fitted grids that adapt to both water depth and depth gradient. The scheme can be used in arbitrary two-dimensional regions and is based on the application of the well-known control function approach to generate adaptive grids. The method includes the evaluation of water depths at the grid points from a known distribution of depth points and their associated depths plus a procedure for the numerical evaluation of depth gradients. It is demonstrated that the smoothness of the grid can be enhanced by introducing a suitable filtering technique.

KEY WORDS Boundary-fitted grids Adaptive grids Filtering

1. INTRODUCTION

The use of boundary-fitted grids for the solution of fluid dynamics problems using finite difference approximations has been given increased attention during the last few years.[1–4] It is commonly agreed that such grids perform much better than conventional Cartesian grids and preliminary results show that flow solutions obtained on these grids have an accuracy comparable to that of finite element methods,[5] the latter requiring much more computational effort. Furthermore, it is possible to write grid generation software that is applicable to arbitrary flow geometries, particularly if the region is divided into contiguous subregions.

The purpose of this work is to develop methods to generate boundary-fitted grids for use in computational hydraulics. In this field of study it is common practice to average the three-dimensional flow equations over the water depth, thus resulting in a two-dimensional system. An accurate numerical solution of such problems requires a grid that clusters in regions of large water depth and in regions of large depth gradient. This clustering is accomplished here by combining these two physical parameters into a so-called weight function. The advantage of this approach is that several methods exist for the generation of grids that concentrate according to some given weight function.

The first work on depth clustering using boundary-fitted grids was done by Johnson and Thompson.[6,7] They used a variational approach to generate grids that cluster according to the water depths but they introduced hypothetical depths in order to make the grid adapt properly. In a further study by Kim[8] it is concluded that this variational approach is less sensitive to weight

0271–2091/90/070741–12$06.00
© 1990 by John Wiley & Sons, Ltd.

Received May 1989
Revised August 1989

functions than the so-called control function approach (based on the solution of an elliptic partial differential equation system) because of constraints on smoothness and orthogonality. For these reasons the latter approach is adopted here.

In this paper we first describe the control function approach for generating adaptive grids. This approach includes the evaluation of control functions (used to control the distribution of grid lines) from the arc length distribution on the boundaries, the curvature of the boundaries and from the given weight function. However, a disadvantage of this approach is that rapid changes in boundary curvature are reflected in the flow region. To remove this undesired effect we introduce a lowpass filtering of the boundary curvature term. Next we describe how to evaluate the depths and the depth gradients at the grid points. When this step is completed it is an easy task to determine the weight functions needed to generate the adaptive grid. However, for computational reasons this procedure is only applied to evaluate the initial grid. During the iterative solution of the grid equations, the weight functions are updated. Finally, the capability of the scheme described above is demonstrated by showing examples of grids generated for a Danish fjord.

2. THE CONTROL-FUNCTION APPROACH FOR GENERATING ADAPTIVE GRIDS

One of the most well-known and well-established methods for generating boundary-fitted grids is to solve an elliptic partial differential equation system.[9] In two dimensions the co-ordinates of the grid points, (x, y), are generated by[9]

$$\begin{aligned} g_{22}(x_{\xi\xi} + Px_\xi) + g_{11}(x_{\eta\eta} + Qx_\eta) - 2g_{12}x_{\xi\eta} &= 0, \\ g_{22}(y_{\xi\xi} + Py_\xi) + g_{11}(y_{\eta\eta} + Qy_\eta) - 2g_{12}y_{\xi\eta} &= 0, \end{aligned} \tag{1}$$

where

$$\begin{aligned} g_{11} &= x_\xi^2 + y_\xi^2, \\ g_{12} &= x_\xi x_\eta + y_\xi y_\eta, \\ g_{22} &= x_\eta^2 + y_\eta^2. \end{aligned} \tag{2}$$

From these equations the co-ordinates of the grid points are determined from a specified distribution of curvilinear co-ordinates (ξ, η). P and Q are control functions used to control the quality of the grid. Originally these functions were evaluated in a manner that ensures that the grid lines generally follow the boundary point distribution. This yields control functions of the form[9, 10]

$$\begin{aligned} P_g &= A_1 + s_1\kappa_1, \\ Q_g &= A_2 + s_2\kappa_2. \end{aligned} \tag{3}$$

The terms A_1 and A_2 can be interpreted as contributions from the rate of change of arc length spacings in the ξ- and η-direction respectively. The terms s_1 and s_2 are the arc length spacings along the ξ- and η-co-ordinate lines. Finally, κ_1 and κ_2 are the curvatures along the η- and ξ-co-ordinate lines (the crossing grid line). Each of the six terms in (3) are evaluated individually on the appropriate boundaries and then interpolated into the field.[10]

All the derivatives in (1) are approximated by central differences. The discretized equations are then solved by point relaxation (SOR).

Most of the methods that have been used to generate adaptive boundary-fitted grids are based on an equidistribution statement, i.e. grid points are closely spaced in regions of large weight functions and vice versa. It can be shown that this equidistribution principle is equivalent to introducing control functions in (1) given by[11, 12]

$$P_W = \frac{(W_1)_\xi}{W_1} - \frac{g_{12}}{g_{22}} \frac{(W_1)_\eta}{W_1},$$
$$Q_W = \frac{(W_2)_\eta}{W_2} - \frac{g_{12}}{g_{11}} \frac{(W_2)_\xi}{W_2}, \tag{4}$$

where W_1 and W_2 are weight functions used to cluster points in the ξ- and η-direction respectively. The first terms in these equations evaluate the control functions in each direction from the variation of weight in that direction, while the additional terms take the transverse variation into account.

In practice it is convenient to use a linear combination of the geometric and the adaptive control functions. Thus

$$P = P_g + C_W P_W,$$
$$Q = Q_g + C_W Q_W, \tag{5}$$

where C_W is an arbitrary parameter is used in the generation equations (1).

In practice it is most convenient to divide the flow domain into a number of subregions and generate the grid independently for each subregion. In order to maintain the continuity of the entire grid it is necessary to transfer information over the interfaces during the iterative solution of (1).

3. FILTERING OF THE CURVATURE TERM

A disadvantage involved in using the grid generation system (1) is that a rapid variation in boundary curvature (such as a slope discontinuity) is propagated into the field (see Figure 1). The source of this unwanted effect, which particularly limits the quality of adaptive grids, can readily be identified as the curvature term in (3). In order to eliminate this problem a lowpass filtering is

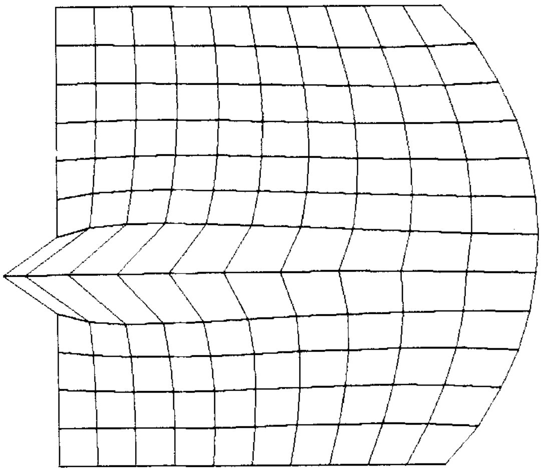

Figure 1. Grid generated from the elliptic generation system without filtering of the curvature term. It is clearly seen how the slope discontinuity occurring on the left boundary is propagated into the field

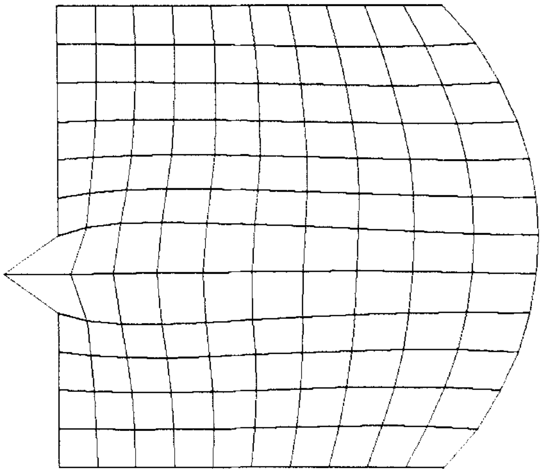

Figure 2. Grid generated for the same region as in Figure 1 with a lowpass-filtered curvature term

introduced.[13] If j denotes the jth grid point on the appropriate boundary, the filter is defined by

$$\kappa_j^{(1)} = \tfrac{1}{4}\kappa_{j-1} + \tfrac{1}{2}\kappa_j + \tfrac{1}{4}\kappa_{j+1}, \tag{6}$$

where $\kappa_j^{(1)}$ is the value of κ_j after filtering. Preliminary numerical experiments showed that the best results were obtained if the filter was used iteratively three or four times. Figure 2 illustrates the effect on the grid of the boundary curvature filtering. By comparison with Figure 1 it is clearly seen that the slope discontinuity is not reflected in the field after filtering.

Our results show that this filtering technique improves the smoothness of the grid. Furthermore, the numerical solution of the grid generation equations (1) converges faster. This is particularly true when generating adaptive grids.

The amount of filtering might be a bit excessive and might smooth out curvature variations even when they do not occur abruptly. However, in the present application no such problems were encountered. An alternative procedure would be to use a smaller amount of filtering, coupled with lower weight being placed on the curvature terms in (3). It is the authors' belief that the latter procedure will not produce significantly different grids.

4. EVALUATION OF DEPTH- AND DEPTH-GRADIENT-DEPENDENT WEIGHT FUNCTIONS

4.1. Choice of weight functions

The purpose of this work is to generate grids that cluster according to water depths and depth gradients. It is thus natural to choose the weight function as

$$W_1(x, y) = W_2(x, y) = W(x, y) = 1 + c_1 f + c_2 |\nabla f|, \tag{7}$$

where c_1 and c_2 are arbitrary constants. In (7), f is to be interpreted as a dimensionless water depth and $|\nabla f|$ as the absolute value of the corresponding (dimensionless) depth gradient. The '1' in (7) ensures that the weight function never becomes zero (or close to zero), which would produce values of the control functions (see (4)) that are infinite (or very large). Similarly, adaptation to shallow regions could be accomplished by choosing $W(x, y) = (1 + c_1 f)^{-1}$.

4.2. Evaluation of depths at grid points

In practical applications the water depths are only known at a number of randomly distributed points. The grid generation system (1), however, needs the values of the weight functions at the current grid points (the grid generation equations are solved by an iterative method). In order to accomplish this we have somehow to evaluate the water depths at the current grid points from the depths at the points where it is known. In the present work the evaluation is made by introducing a bilinear interpolation scheme that locally approximates the depth profile as a plane surface. The scheme is given by

$$f(x, y) = ax + by + c, \tag{8}$$

where the coefficients a, b and c are evaluated form three depth points close to the grid point in question. The interpolation error is reduced if these points generally surround the grid point in question. In this work the surrounding depth points are chosen in a manner that ensures that the grid point lies inside the triangle formed by these depth points[5] (see Figure 3). In the present application no problems were encountered in finding the three forming points of the triangle, except when the grid point and points #1 and #2 were lying on a straight line (the boundary points are added to the depth points). However, other examples exist where the forming points cannot be found. In these cases the three depth points closest to the grid point should be used in the bilinear interpolation.

As the search for surrounding depth points is very CPU-time-consuming, the procedure described above is only used to evaluate the depths at the initial and final grid points. During

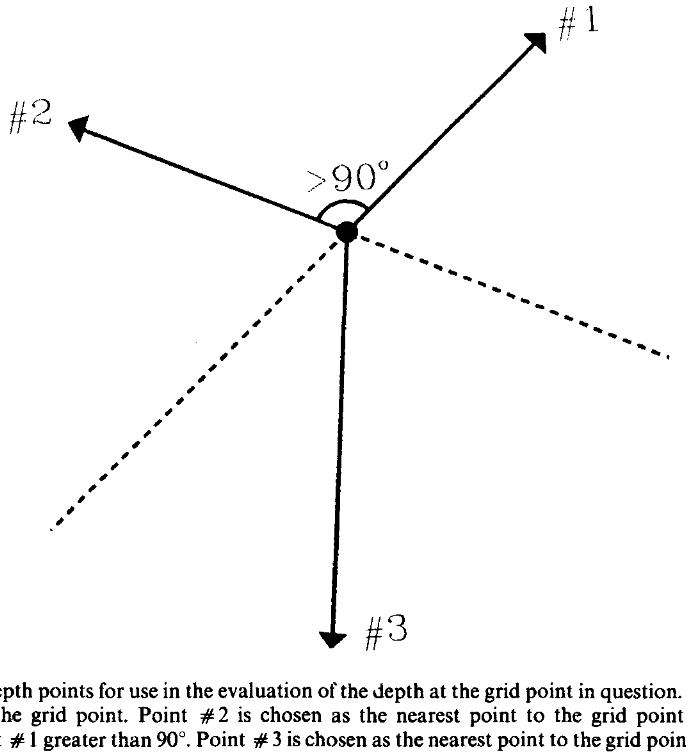

Figure 3. Choice of depth points for use in the evaluation of the depth at the grid point in question. Point #1 is chosen as the nearest point to the grid point. Point #2 is chosen as the nearest point to the grid point that has an angle of intersection with point #1 greater than 90°. Point #3 is chosen as the nearest point to the grid point that lies in the region between the backward extensions of the vectors from the grid point to the points #1 and #2. This procedure ensures that the grid point lies inside a triangle formed by points #1, #2 and #3 (The method and figure are outlined in Reference 6.)

the iterative solution of the grid generation equations the weight functions are updated by application of the chain rule as explained in Section 4.4.

4.3. Evaluation of the depth gradient

Once the depths at the grid points are known it is straightforward to evaluate the depth gradient. In Cartesian co-ordinates it is given by

$$\nabla f(x, y) = f_x \mathbf{x} + f_y \mathbf{y}, \tag{9}$$

where **x** and **y** denote unit vectors in the x- and y-direction respectively. The derivatives f_x and f_y are easily determined from the basic transformation relations,[9] i.e.

$$\begin{aligned} f_x &= \frac{1}{J}(y_\eta f_\xi - y_\xi f_\eta), \\ f_y &= \frac{1}{J}(- x_\eta f_\xi + x_\xi f_\eta), \end{aligned} \tag{10}$$

where J denotes the Jacobian of the transformation from curvilinear to physical co-ordinates and is given by

$$J = x_\xi y_\eta - x_\eta y_\xi. \tag{11}$$

It should be noted that it is the absolute value of the gradient that is used in the weight function (see (7)).

4.4. Updating of the weight function

Since the grid generation equations are solved by an iterative method, the grid moves during the solution of the grid system. For this reason it is necessary to update the weight function at the

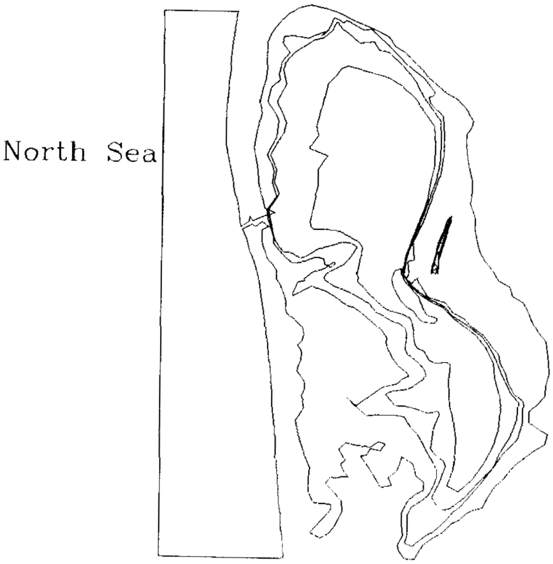

Figure 4. Boundary geometry and depth contours for Ringkoebing Fjord. Contour curves correspond to depths of 0, 1, 2, 3 and 4 m respectively. In the present test case no depth contours are included for the North Sea region

grid points after each iteration. Since the procedure described in Section 4.2 requires too much computer time, it is only used to determine the depths at the initial grid points. Instead, the chain rule is applied to update the weight function at each grid point. This yields

$$\Delta W = W_x \Delta x + W_y \Delta y, \tag{12}$$

where ΔW is the change in value of the weight function and Δx and Δy account for the movement of the grid point between two successive iterations. The derivatives in (12) are determined by the transformation relations

$$\begin{aligned} W_x &= \frac{1}{J}(y_\eta W_\xi - y_\xi W_\eta), \\ W_y &= \frac{1}{J}(-x_\eta W_\xi + x_\xi W_\eta). \end{aligned} \tag{13}$$

To solve the hydraulic flow equations, the grid generator should supply both the grid points and the associated water depths. Since preliminary results showed that the updating procedure described by (12) resulted in rather inaccurate depths, the procedure described in Section 4.2 is used again after the grid equations have converged in order to produce more accurate depths.

5. APPLICATION TO RINGKOEBING FJORD

In order to demonstrate the capability of the scheme described above, grids were generated for Ringkoebing Fjord, which is located on the Danish North Sea coast. The geometry and the depth contour of the fjord are illustrated in Figure 4. The geometry is complicated by a small, long and

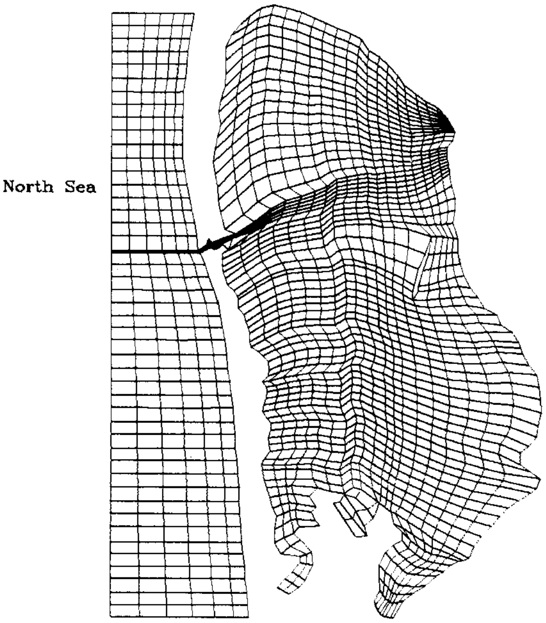

Figure 5. Non-adaptive grid for Ringkoebing Fjord generated without filtering of the curvature term ($C_W = 0$) (geometric control functions only)

narrow island in the fjord. The depth varies from zero at the boundaries to a maximum depth of 4·7 m in the middle of the fjord.

Figures 5 and 6 show grids generated without and with lowpass filtering of the curvature term respectively. It is clearly seen that the filtering greatly enhances the smoothness of the grid.

The adaptive grids were generated using 544 randomly distributed depth points with known depths (read from a chart) using the non-adaptive grid generated by (1) ($C_W = 0$) as an initial grid (see Figure 6). The computer code automatically adds the boundary points to the depth points with a depth equal to zero at the former.

The capability of generating depth-clustered grids is illustrated in Figures 6–8 ($c_1 = 1{\cdot}0$, $c_2 = 0{\cdot}0$, $C_W = 0{\cdot}0$, $1{\cdot}5$ and $2{\cdot}0$ respectively). It is clearly seen how the grid lines move away from the boundaries towards regions of large depths and that the amount of depth clustering can be controlled through the parameter C_W.

Figures 6, 9 and 10 ($c_1 = 0{\cdot}0$, $c_2 = 1{\cdot}0$, $c_W = 0{\cdot}0$, $1{\cdot}5$ and $2{\cdot}0$ respectively) demonstrate that the method is capable of generating grids that cluster according to the depth gradient. The grids, however, seem to suffer from a lack of smoothness. A more thorough examination showed that some peaks occurred in the numerically evaluated depth gradient profile. These peaks can be eliminated by introducing an appropriate smoothing of the weight function. This smoothing enhances the smoothness of the grid.

When a valley approximately aligned with one family of grid lines exists in the geometry, a proper grid will be one with cells that are relatively long in the direction along the valley compared with that across. This cannot be accomplished with the use of a single weight function, since this will tend to produce cells of unit aspect ratio. However, the proper adaptation can be accomplished by using separate weight functions (see (4)) along the two families of grid lines, each reacting to the depth gradient in that direction.

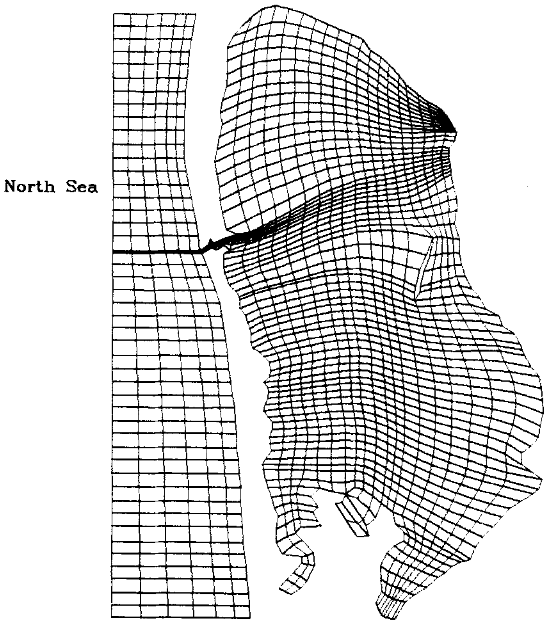

Figure 6. Non-adaptive grid for Ringkoebing Fjord generated with lowpass filtering of the curvature term (geometric control functions only)

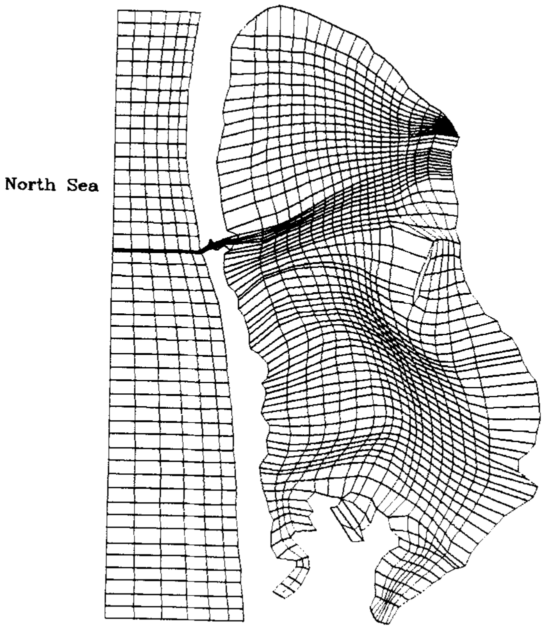

Figure 7. Adaptive grid generated for Ringkoebing Fjord ($c_1 = 1{\cdot}0$, $c_2 = 0{\cdot}0$, $C_W = 1{\cdot}5$) (adaptation to depth)

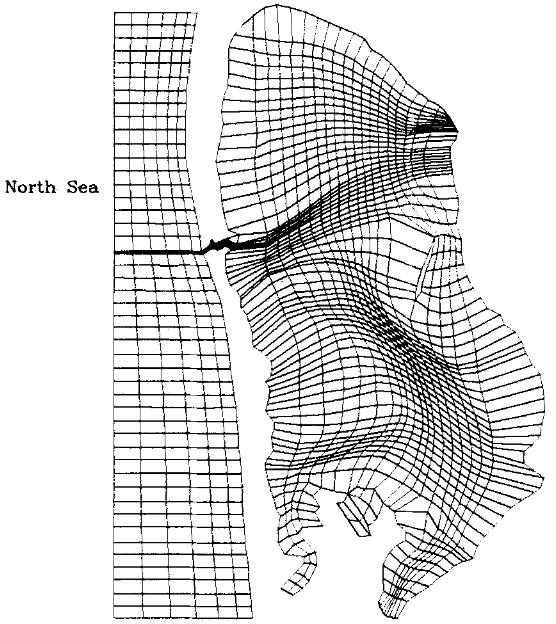

Figure 8. Adaptive grid generated for Ringkoebing Fjord ($c_1 = 1{\cdot}0$, $c_2 = 0{\cdot}0$, $C_W = 2{\cdot}0$) (adaptation to depth)

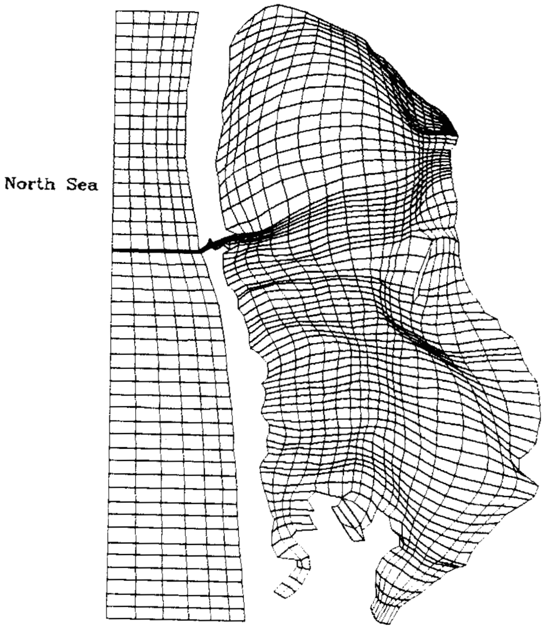

Figure 9. Adaptive grid generated for Ringkoebing Fjord ($c_1 = 0{\cdot}0$, $c_2 = 1{\cdot}0$, $C_W = 1{\cdot}5$) (adaptation to depth gradient)

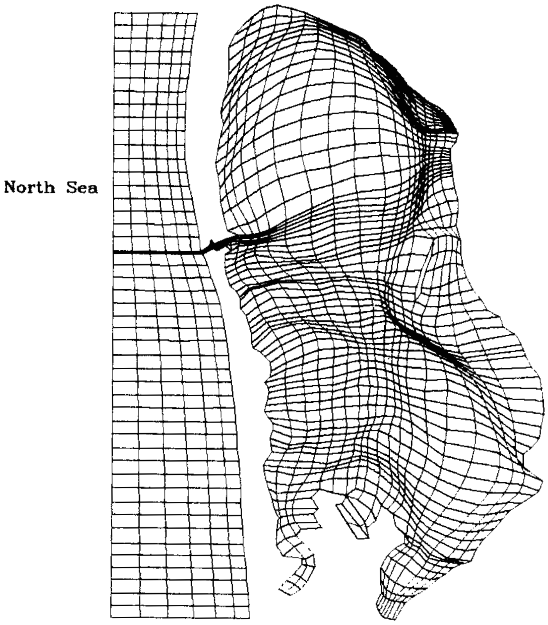

Figure 10. Adaptive grid generated for Ringkoebing Fjord ($c_1 = 0{\cdot}0$, $c_2 = 1{\cdot}0$, $C_W = 2{\cdot}0$) (adaptation to depth gradient)

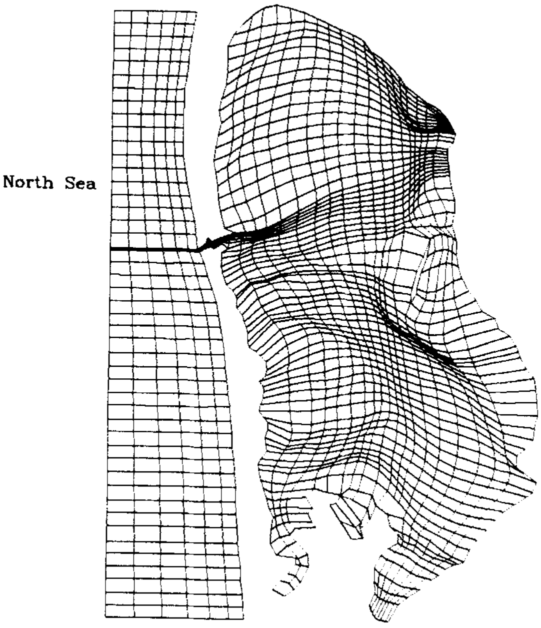

Figure 11. Adaptive grid generated for Ringkoebing Fjord ($c_1 = 1{\cdot}0$, $c_2 = 3{\cdot}0$, $C_W = 1{\cdot}5$) (adaptation to both depth and depth gradient)

Finally, Figure 11 ($c_1 = 1{\cdot}0$, $c_2 = 3{\cdot}0$, $C_W = 1{\cdot}5$) shows that it is possible to generate grids that cluster both in regions of large water depths and in regions of large depth gradients.

6. CONCLUSIONS

A scheme for generating grids that cluster according to water depths and depth gradients has been introduced. The scheme is applicable to arbitrary two-dimensional flow regions and is based on the well-known control function approach for generating adaptive grids. This approach includes the evaluation of control functions from the arc length distribution on the boundaries, the boundary curvature and from depth- and depth-gradient-dependent weight functions. As a novelty, a lowpass filter is introduced which enhances the smoothness of the grid by eliminating the rapid fluctuations of the curvature term.

Future research is directed towards the adaptation to a valley approximately aligned with one family of grid lines such that the cells are relatively long in the direction along the valley compared with that across.

ACKNOWLEDGEMENT

The financial support of the Danish Technical Research Council (through contract no. 5.17.3.6.12 as a part of the FTU programme) is gratefully acknowledged.

REFERENCES

1. J. Hauser and C. Taylor (eds), *Numerical Grid Generation in Computational Fluid Dynamics*, Pineridge Press, Swansea, 1986.

2. S. Sengupta, J. Hauser, P. R. Eiseman and J. F. Thompson (eds), *Numerical Grid Generation in Computational Fluid Mechanics, '88*, Pineridge Press, Swansea, 1988.
3. J. Hauser, H. G. Paap, D. Eppel and S. Sengupta, 'Boundary conformed co-ordinate systems for selected two-dimensional fluid flow problems. Part I: Generation of BFGs', *Int. j. numer. methods fluids*, **6**, 507–527 (1986).
4. J. Hauser, H. G. Paap, D. Eppel and S. Sengupta, 'Boundary conformed co-ordinate systems for selected two-dimensional fluid flow problems. Part II: Application of the BFG method', *Int. j. numer. methods fluids*, **6**, 529–539 (1986).
5. T.-J. Liu, H.-M. Lin and C.-N. Hong, 'Comparison of two numerical methods for the solution of non-Newtonian flow in ducts', *Int. j. numer. methods fluids*, **8**, 845–861 (1988).
6. J. F. Thompson and B. H. Johnson, 'Development of an adaptive boundary-fitted coordinate code for use in coastal and estuarine areas', *Miscellaneous Paper HL-85-5*, U.S. Army Engineer Waterways Experiment Station, Vicksburg, Mississippi, 1985.
7. B. H. Johnson and J. F. Thompson, 'Discussion of a depth-dependent adaptive grid generator for use in computational hydraulics', in J. Hauser and C. Taylor (eds), *Numerical Grid Generation in Computational Fluid Dynamics*, Pineridge Press, Swansea, 1986, pp. 629–640.
8. H. J. Kim, 'Three-dimensional adaptive grid generation on a composite structure', *Ph.D. Dissertation*, Mississippi State University, 1987.
9. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, *Numerical Grid Generation—Foundations and Applications*, Elsevier, New York, 1985.
10. J. F. Thompson, 'A general three-dimensional elliptic grid generation system on a composite structure', *Comput. Methods Appl. Mech. Eng.*, **64**, 377–411 (1987).
11. P. R. Eiseman, 'Adaptive grid generation', *Comput. Methods Appl. Mech. Eng.*, **64**, 321–376 (1987).
12. H. J. Kim and J. F. Thompson, 'Three dimensional adaptive grid generation on a composite block grid', *AIAA 26th Aerospace Sciences Meeting*, Reno, Nevada, 1988.
13. R. Shapiro, 'Linear filtering', *Math. Comput.*, **29**, 1094–1097 (1975).