INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS, VOL. 6, 507–527 (1986)

BOUNDARY CONFORMED CO-ORDINATE SYSTEMS FOR SELECTED TWO-DIMENSIONAL FLUID FLOW PROBLEMS. PART I: GENERATION OF BFGs

J. HÄUSER

Department of Electrical and Mechanical Engineering, College of Landshut, Am Lurtenhof 4, 8300 Landshut, Germany

H. G. PAAP

Physikalisches Institut, University of Bayreuth NW II, 8580 Bayreuth, Germany

D. EPPEL

Institute of Physics, GKSS-Research Center, 2054 Geesthacht, Germany

AND

S. SENGUPTA

Department of Mechanical Engineering, University of Miami, Miami, FL 33124, U.S.A.

SUMMARY

In this paper the generation of general curvilinear co-ordinate systems for use in selected two-dimensional fluid flow problems is presented. The curvilinear co-ordinate systems are obtained from the numerical solution of a system of Poisson equations. The computational grids obtained by this technique allow for curved grid lines such that the boundary of the solution domain coincides with a grid line. Hence, these meshes are called boundary fitted grids (BFG). The physical solution area is mapped onto a set of connected rectangles in the transformed (computational) plane which form a composite mesh. All numerical calculations are performed in the transformed plane. Since the computational domain is a rectangle and a uniform grid with mesh spacings $\Delta\xi = \Delta\eta = 1$ (in two-dimensions) is used, the computer programming is substantially facilitated. By means of control functions, which form the r.h.s. of the Poisson equations, the clustering of grid lines or grid points is governed. This allows a very fine resolution at certain specified locations and includes adaptive grid generation. The first two sections outline the general features of BFGs, and in section 3 the general transformation rules along with the necessary concepts of differential geometry are given. In section 4 the transformed grid generation equations are derived and control functions are specified. Expressions for grid adaptation are also presented. Section 5 briefly discusses the numerical solution of the transformed grid generation equations using successive overrelaxation and shows a sample calculation where the FAS (full approximation scheme) multigrid technique was employed. In the companion paper (Part II), the application of the BFG method to selected fluid flow problems is addressed.

KEY WORDS Computational Fluids Dynamics Numerical Grid Generation Two-dimensional Fluid Flow Problems

1. INTRODUCTION

Since BFGs are a novel approach to environmental flow computations, a basic introduction to their properties and their implementation will be presented. First, BFG methods will be put into historical context in relation to finite difference and finite element techniques.

Until recently, most of the problems in computational fluid dynamics (CFD) were solved by

0271–2091/86/080507–21$10.50

Received 5 January 1986
Revised 6 January 1986

finite difference methods using Cartesian co-ordinate systems, i.e. rectangular grids. With the introduction of finite elements in CFD,[1-3] problems of greater geometrical complexity could be solved. Finite elements, however, demand the construction of a table of nearest neighbours for the irregular grid, and the programming is therefore much more laborious. Furthermore, the matrices resulting from finite elements are in general not sparse and are more cumbersome to invert. Thacker[4] used finite differences on an irregular grid and reported that for the case of the SWEs (shallow water equations) finite differences were approximately an order of magnitude faster than Galerkin finite elements when the same accuracy for both techniques was demanded, though the number of grid points for the finite element technique was smaller. Hence, a method which is generally applicable and which retains the computational efficiency of finite differences as well as the geometrical flexibility of finite elements would be desirable. To a large extent, BFGs provide these desirable features. A BFG exactly matches curved boundaries where the solution domain may also be multiply connected. However, all calculations are performed on a regular grid; that is, a square grid with uniform grid spacings is used. Such a grid must have certain features in order to produce meaningful numerical results of the governing physical equations. In particular, many of the grids generated by so called finite element grid generators are not well suited for time-dependent problems in CFD.

BFGs are generated by co-ordinate transformations. The use of curvilinear co-ordinates is well known from general relativity and the concepts used there also apply to BFGs.[5-11] Two questions arise with the generation of computational grids. First, how to automatically generate grids which are well suited for CFD computations and, secondly, how to control grid point distribution.

If, for example, Laplace's equation had to be solved on an annular ring, the natural choice would be the use of polar co-ordinates. In this case the approximation of the solution area by rectangles or triangles would not be considered optimal. Furthermore, for calculations on a sphere, spherical co-ordinates would be used, whereas elliptical co-ordinates would be the natural choice if the solution area was an ellipse. All these co-ordinate systems have in common that boundaries of the solution area coincide with co-ordinate lines. The solution area of an annulus is determined by $r_1 \leqslant r \leqslant r_2$ and the azimuthal angle $0 < \psi < 2\pi$. In the transformed plane having r and ψ as co-ordinate axes, these co-ordinates define a rectangle. Figure 1 shows the mapping of a circle. Except for $r = 0$ the mapping is one-to-one. For the general application of BFGs to arbitrary solution domains, co-ordinate systems have to be found such that co-ordinate lines coincide with the boundaries of the solution area.

According to the transformation, all PDEs (partial differential equations) and their respective BCs (boundary conditions) have to be transformed. If the grid is generated by the numerical solution of PDEs, these equations must also be transformed. All transformed equations, then, are solved in the computational plane. In simple cases, the solution area is mapped onto a single rectangle and the calculations are performed on the square grid which spans this rectangle. Since

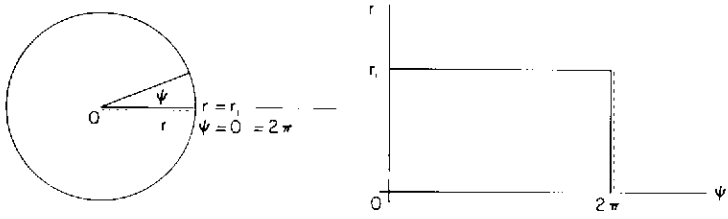

Figure 1. Physical and transformed planes, exemplified for a circle and polar co-ordinates

all computations are on a regular grid, finite differences or discrete elements (finite volume) are used.[12,13] Moreover, curved boundaries are accounted for and no loss of accuracy due to inaccurate modelling of the boundary occurs. Thirdly, grid point positions can be easily changed according to the needs of the physical solution. If a system of Poisson equations is employed for the calculation of grid point positions, the specification of appropriate control functions (see below) allows the desired clustering of grid points. Poisson equations have been used in the work of Thompson and coworkers,[14–18] where the r.h.s. governs grid line control. This can be visualized by an electrostatic problem where charges (r.h.s.) change the positions of the equipotential lines.

Therefore, the problem of solving the governing equations on a complex solution domain has been modified to that of solving the transformed equations. On a uniform grid of rectangular shape in the computational plane the transformation generates additional terms but does not change the type of the equation. For more complicated solution domains the mapping onto a set of connected rectangles is necessary. In this case the physical domain is subdivided into a set of segments where each segment is then mapped onto a rectangle. Such a grid is called a composite or patched grid.[19]

As has already been mentioned, the r.h.s. of Poisson's equation allows comprehensive grid line control (the same is true for algebraic grid generation[10]). Moreover, it is possible to consider problems with time dependent geometry. This will also be useful for inundating flow.

Grid point distribution can also be used to minimize numerical error[20] or to track the propagation of shock fronts. Again, this can also be achieved by algebraic grid generation.[21]

In addition to the work of Thompson *et al.*, parabolic and hyperbolic PDEs[22,23] have been used to generate grids. Instead of solving PDEs, complex variable methods and algebraic methods have also been used. Unfortunately complex variable methods are restricted to two dimensions. Algebraic techniques use algebraic expressions to cluster grid points near fixed boundaries. A simple example of the algebraic technique is a domain normalizing transformation to generate a mesh for solution of the flow in a diverging nozzle. Algebraic methods have been developed, for example, by Eiseman and Smith.[24,25] The proceedings published by Thompson,[26] Smith[27] and Ghia[28] also contain work on algebraic techniques. A short introduction to both algebraic and differential equation techniques is found in Reference 29. A concise review on grid generation techniques is found in Reference 30. There is a recent monograph on numerical grid generation by Thompson *et al.*[31]

2. FEATURES OF BOUNDARY FITTED GRIDS

In this section a survey of the features of BFGs is presented. All examples in this paper were obtained using a composite grid which is best suited for complex physical domains.

For a specified solution domain several co-ordinate systems are possible. For example, polar co-ordinates map a circle onto a rectangle via a branch cut (Figure 1). Owing to this cut, the east and west sides of the rectangle in the computational plane correspond to the same physical locations. Therefore no BCs must be specified along these re-entrant boundaries. Another co-ordinate system for the circle is given in Figure 2. The transformation depicted in Figure 2 maps only boundary points in the physical plane onto boundary points in the computational plane. This transformation has four singular points at the edges of the rectangle since in the physical plane ξ and η co-ordinate lines intersect at an angle π.

For a multiply connected area more choices exist; e.g. for a doubly connected area a branch cut can be used (Figure 3) or slits (Figure 5) or slabs (Figure 6) can be employed. The grid point distributions resulting from the corresponding composite grids are depicted in Figure 8–10. The

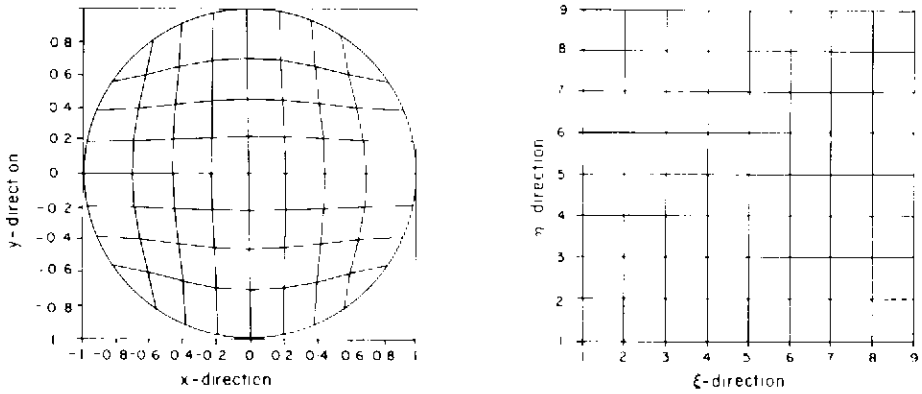

Figure 2. Transformation of a circle onto a rectangle without the use of polar co-ordinates

use of branch cuts corresponds to polar co-ordinates and gives a grid line pattern similar to the one obtained for an annular ring.

Since absolute values of grid spacings in the computational plane are not important, they are set to 1. In two-dimensions, where co-ordinate lines are denoted by ξ and η, we have $\Delta\xi = \Delta\eta = 1$. Lines with a fixed ξ-value are called ξ-lines, i.e. η varies from 1 to M, where M indicates the number of grid points in this direction. The same holds for η-lines, where ξ varies from 1 to N.

Figure 3 is an example for a region with two islands and the respective branch cuts. The corresponding grid line configuration is shown in Figure 4. If one visualizes the streamline pattern resulting from laminar flow past a cylinder, it can be easily shown that such a grid line pattern cannot be constructed via branch cuts. Consider for example a grid line which starts from the east side of the boundary in the physical plane and which ends at the west side (it is assumed that the shape of the solution area is such that east, west north and south sides exist). These two sides must be mapped onto opposite sides of the rectangle in the computational plane in order to obtain a grid line pattern like those in Figures 5 and 6. The same holds for the north and south sides in the physical plane. Hence, there is no possibility of mapping both the branch cut and the island onto the boundary of the rectangle in the computational plane. The island is therefore mapped onto a slit (Figure 5), which is double-valued, or onto a slab (Figure 6). This approach poses the problem that within the computational grid there are points which lie outside the computational area and makes additional book keeping necessary. For more complicated solution domains the transformed area must be different from a rectangle (Figure 7). To overcome this problem, a composite grid is used. It is, however, necessary to specify the segmentation and to prescribe how individual segments are connected to each other. Using the concepts of differential geometry, an atlas of compatible charts is constructed which covers the physical solution area. The composite grids corresponding to Figures 4–6 are shown in Figures 8–10, respectively.

As has already been mentioned, Laplace or Poisson equations are used to determine grid point positions. It is known that the solution of a Laplace equation obeys a maximum principle. In order to retain this feature, the r.h.s. of Poisson's equation must be properly chosen. The electrostatic problem mentioned in the introduction leads to the grid line generation system

$$\Delta\xi = P; \qquad \Delta\eta = Q + \text{BCs}, \tag{1}$$

where P and Q are functions of ξ and η or x and y. If P and Q depend on x and y, iteration

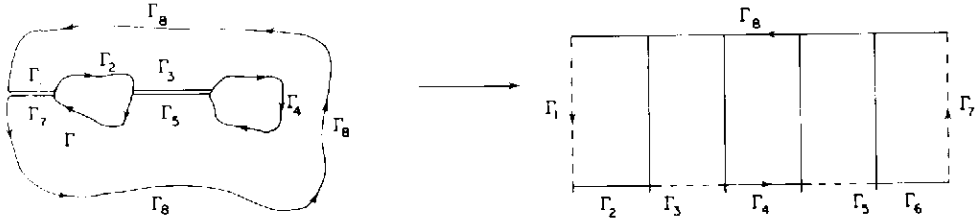

Figure 3. Transformation of a domain with two islands onto a single rectangle by use of two branch cuts. Re-entrant boundaries are depicted by dashed lines. Re-entrant boundaries corresponding to the same physical locations are connected by solid lines

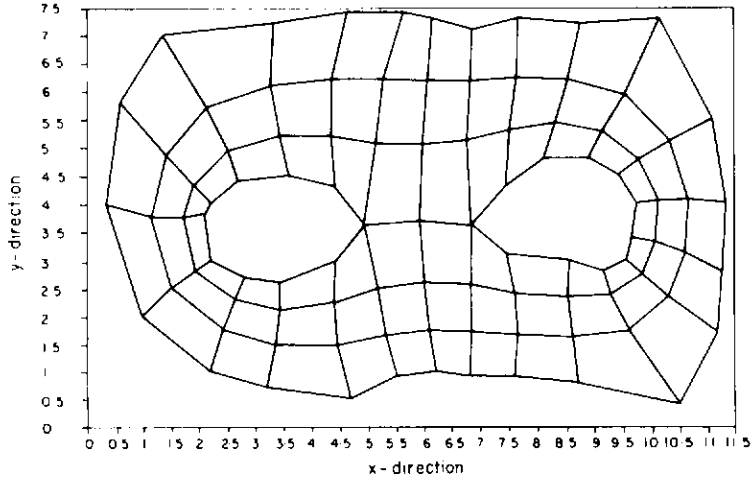

Figure 4. Grid line distribution as obtained by the use of branch cuts (see Figure 3)

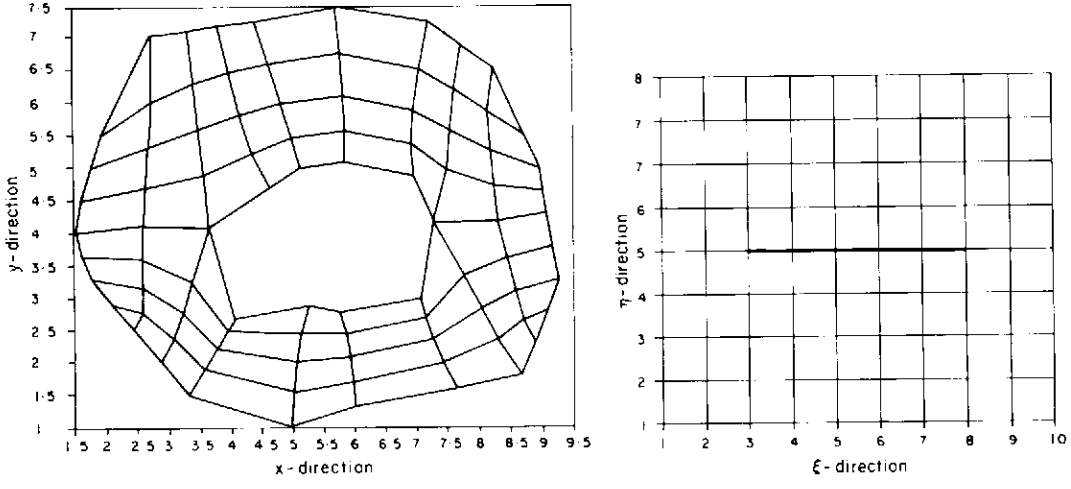

Figure 5. Transformation of an island onto a slit

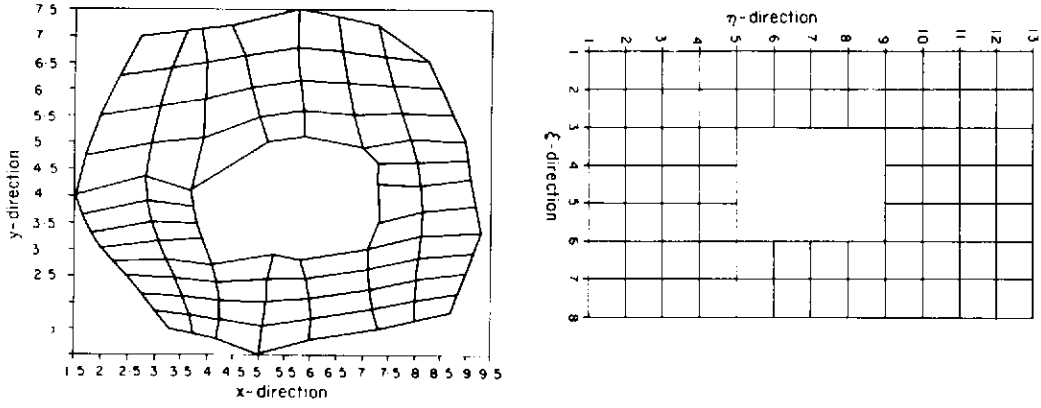

Figure 6. Transformation of a doubly-connected area. This transformation has the disadvantage that the transformed area is no longer a rectangle

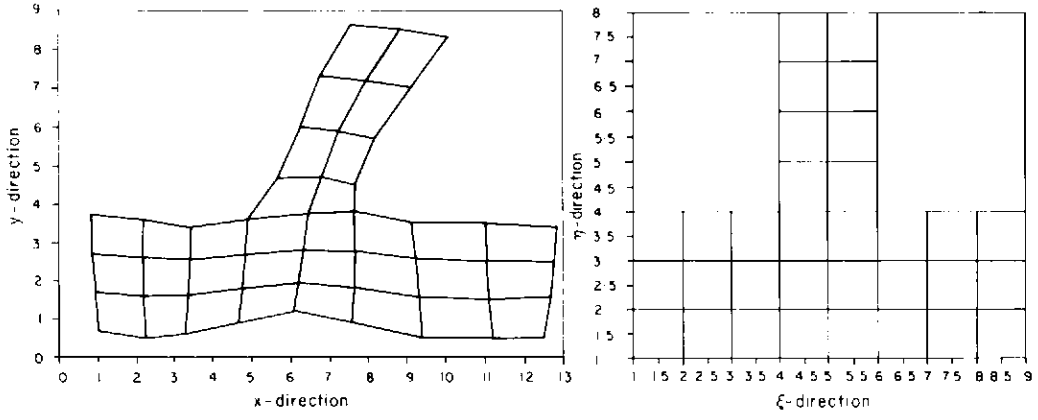

Figure 7. Example of the transformation of a river junction

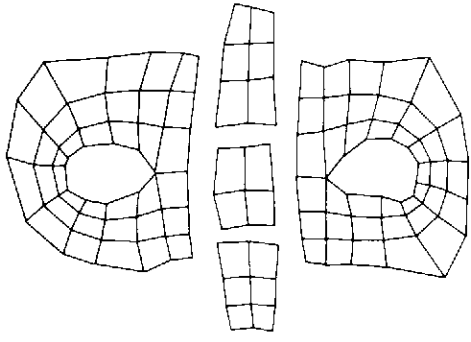

Figure 8. Segment structure for solution area shown in Figure 4

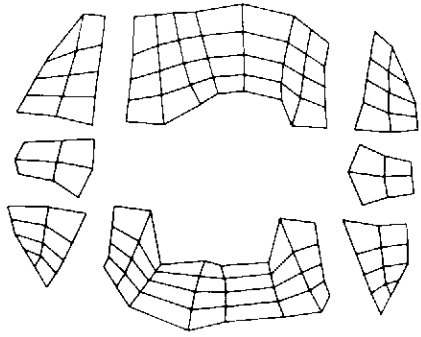

Figure 9. Segment structure for solution area shown in Figure 5

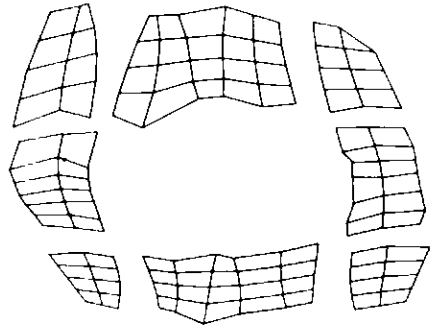

Figure 10. Segment structure for solution area shown in Figure 6

is necessary. It is sufficient to consider equations (1), since all regular co-ordinate systems satisfy these equations for appropriate functions P and Q. Since Dirichlet BCs are employed, that is values $\xi = 1(1)N$ are prescribed on the η-line boundaries $\eta = 1$ and $\eta = M$, and values $\eta = 1(1)M$ are used for the ξ-line boundaries $\xi = 1$ and $\xi = N$, the respective x- and y-values are employed in the transformed plane.

Many choices are possible for P and Q.[14–17] In this paper we use the method of Middlecoff and Thomas[32] where P and Q are calculated from the boundary point distribution and also give a method for grid adaptation. Grid points are clustered automatically in regions where boundary points are densely specified. In many cases only a few boundary points would suffice to describe the boundary. Since P and Q are calculated from the boundary point distribution, however, this requires the user to specify many more boundary points to achieve the desired grid point distribution. For this purpose an automatic grid doubling algorithm was written. Starting from a coarse initial grid, the grid is doubled each time the module is called. For problems concerned with composite meshes we refer to Reference 33.

Upon calculation of grid point positions, a further module calculates the metric coefficients. The output of this module is the input for the solution module of the transformed physical equations. The pipe concept is used for the connection of all modules. It is mandatory that the solution module reflects the same segment structure as the grid generation program. Before the physical equations are discretized, they are written in contravariant form.

In the numerical calculations it makes a difference whether covariant- or contravariant components are used. If, for example, the upwinding technique is used, contravariant components should be preferred since covariant base vectors are tangent to co-ordinate lines. Hence, numerical diffusion will be reduced by proper grid alignment.

3. DIFFERENTIAL GEOMETRICAL METHODS FOR NUMERICAL GRID GENERATION

In order to use BFGs, all equations must be expressed in curvilinear co-ordinates. Therefore the transformation rules between the different co-ordinate systems are necessary. To this end, we consider a physical solution domain $M \subset \mathbb{R}^n$ and a manifold $A \subset \mathbb{R}^m$ which denotes the transformed plane. It is assumed that a one-to-one and continuous mapping $A \to M$ exists with

$$\mathbf{x}(\xi^1, \ldots, \xi^m) := \begin{pmatrix} x^1(\xi^1, \ldots, \xi^m) \\ \cdots \\ \cdots \\ x^n(\xi^1, \ldots, \xi^m) \end{pmatrix}. \tag{2}$$

As an example, we consider the surface of a sphere where $\mathbb{R}^m = \mathbb{R}^2$ and $\xi^1 = \theta$, $-\pi/2 < \theta < \pi/2$; $\xi^2 = \psi$, $0 < \psi < 2\pi$. The co-ordinates x^1, x^2 and x^3 correspond to the usual Cartesian co-ordinates x, y and z. In the physical space, we have the surface of a sphere, whereas in the transformed space θ and ψ form a rectangle.

The tangent vectors or base vectors at a point $P \in M$ are defined by

$$\mathbf{e}_k := \frac{\partial \mathbf{x}}{\partial \xi^k}; \quad k = 1(1)m. \tag{3}$$

The tangent vector $\mathbf{e}_k$ points in the direction of the respective co-ordinate line. These base vectors are called covariant base vectors. A second set of base vectors is defined by

$$\mathbf{e}^i \cdot \mathbf{e}_j = \delta^i_j. \tag{4}$$

The $\mathbf{e}^i$ are called contravariant base vectors and are orthogonal to the respective covariant vectors for $i \neq j$. Covariant and contravariant vectors are related by the metric coefficients (see below).

For the above example the two tangent vectors $\mathbf{e}_\theta$ and $\mathbf{e}_\psi$ are obtained by differentiating each of the functions $x(\theta, \psi)$, $y(\theta, \psi)$ and $z(\theta, \psi)$ with respect to either θ or ψ.

A physical vector can either be represented by contravariant or covariant components

$$\mathbf{v} = v^i \mathbf{e}_i = v_j \mathbf{e}^j, \tag{5}$$

where the summation convention is employed. In two-dimensions, contravariant components of a vector are found by parallel projection onto the axes, whereas covariant components are obtained by orthogonal projection.

According to its transformation behaviour, a vector is called contravariant or covariant. Let $\underline{v}^i$ be the components of a vector in the co-ordinate system described by the x^i and let v^i be the components in the system ξ^i.

(α) A vector is a contravariant vector if its components transform in the same way as the co-ordinate differentials:

$$\mathrm{d}x^i = \frac{\partial x^i}{\partial \xi^j} \mathrm{d}\xi^j \text{ (chain rule)},$$

$$\underline{v}^i = \frac{\partial x^i}{\partial \xi^j} v^j. \tag{6}$$

(β) A vector is a covariant vector if it transforms in the same way as the gradient of a scalar function:

$$\underline{\phi}(x^1, \ldots, x^n) = \phi(\xi^1, \ldots, \xi^m),$$

$$\frac{\partial \underline{\phi}}{\partial x^i} = \frac{\partial \phi}{\partial \xi^j}\frac{\partial \xi^j}{\partial x^i} = \frac{\partial \xi^j}{\partial x^i}\frac{\partial \phi}{\partial \xi^j}, \tag{7}$$

$$\underline{v_i} = \frac{\partial \xi^j}{\partial x^i} v_j.$$

In order to measure the distance between neighbouring points, the first fundamental form is introduced

$$g_{ij} := \mathbf{e}_i \cdot \mathbf{e}_j. \tag{8}$$

The g_{ij} are also called components of the metric tensor. The components of the inverse matrix are found from

$$g_{ij} g^{ik} = \delta^k_j. \tag{9}$$

The distance ds between two neighbouring points is given by

$$\mathrm{ds} = \sqrt{(g_{ij}\, \mathrm{d}\xi^i\, \mathrm{d}\xi^j)}. \tag{10}$$

In Cartesian co-ordinates there is no difference between covariant and contravariant components since there is no difference between covariant and contravariant base vectors. Therefore the matrix $\hat{g}_{ij}$ (the $\hat{}$ denotes the Cartesian system) is the unit matrix. The components of g_{ij} in any other co-ordinate system can be directly calculated using the chain rule:

$$g_{ij} = \frac{\partial \hat{x}^k}{\partial x^i}\frac{\partial \hat{x}^l}{\partial x^j}\hat{g}_{kl}. \tag{11}$$

In order to find the transformation rules of derivatives of scalars, vectors and tensors, the Christoffel symbols of the first and second kinds are introduced. Suppose that co-ordinate ξ^j is changed by an amount $\mathrm{d}\xi^j$. This changes the base vector $\mathbf{e}_i$ by $\mathbf{de}_i$. Since $\mathbf{de}_i$ is a vector, it can be represented by the system of base vectors $\mathbf{e}_k$. Further, $\mathrm{d}\mathbf{e}_i$ is proportional to $\mathrm{d}\xi^j$. One can therefore write

$$\mathbf{de}_i = \Gamma^k_{ij}\, \mathrm{d}\xi^j \mathbf{e}_k, \tag{12}$$

where the symbols Γ^k_{ij} are only coefficients of proportionality. These symbols are also called Christoffel symbols of the second kind. Taking the scalar product with $\mathbf{e}^k$, one obtains directly from (12)

$$\Gamma^k_{ij} = \mathbf{e}_{i,j} \cdot \mathbf{e}^k, \tag{13}$$

where a comma denotes partial differentiation. The Christoffel symbols of the first kind are defined as

$$\Gamma_{ijk} := g_{il}\Gamma^l_{jk}. \tag{14}$$

If the base vectors are independent of position, the Christoffel symbols vanish. They are, however, not tensor components, which follows directly from their transformation behaviour.

The relationship between the Γ^k_{ij} and the g_{lm} is found in the following way. Insertion of the definition of the metric components g_{lm}, equation (8), into equation (13) and interchanging indices leads to

$$\Gamma^i_{jk} = \tfrac{1}{2} g^{ij}\left(\frac{\partial g_{jl}}{\partial \xi^k} + \frac{\partial g_{kl}}{\partial \xi^j} - \frac{\partial g_{kj}}{\partial \xi^l}\right). \tag{15}$$

If we use equation (3), the definition of the tangent vector, along with equation (13), one obtains the computational useful form

$$\Gamma^i_{jk} = \frac{\partial \xi^i}{\partial x^l} \frac{\partial^2 x^l}{\partial \xi^j \partial \xi^k}. \tag{16}$$

The knowledge of the grid point distribution, then, allows the numerical calculation of the Christoffel symbols. If we contract the Christoffel symbols, i.e. upper and lower indices are the same and are summed over, equation (15) yields

$$\Gamma^i_{ji} = \tfrac{1}{2} g^{im} \frac{\partial g_{mi}}{\partial \xi^j} = \frac{1}{\sqrt{g}} \frac{\partial \sqrt{g}}{\partial \xi^j} = (\ln \sqrt{g})_{,j}, \tag{17}$$

where g is the determinant of the metric tensor, that is $\sqrt{g}$ is the Jacobian of the transformation.

In two-dimensions with curvilinear co-ordinates ξ, η and Cartesian co-ordinates x, y, one finds

$$\begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix} = J^{-1} \begin{pmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{pmatrix}, \tag{18}$$

$$\sqrt{g} = J = (x_\xi y_\eta - y_\xi x_\eta).$$

In curved space, partial differentiation is replaced by covariant differentiation which takes into account the fact that base vectors themselves have non-vanishing spatial derivatives. In the following, the nabla operator

$$\nabla := \mathbf{e}^i \frac{\partial}{\partial \xi^i} \tag{19}$$

is used. For the calculation of the cross product the Levi-Civita tensor is introduced.

$$\hat{\varepsilon}_{ijk} := \begin{cases} 1, \text{ for } i = 1, j = 2, k = 3 \text{ and all even permutations,} \\ -1, \text{ for all odd permutations,} \\ 0, \text{ if any two indices are the same,} \end{cases} \tag{20}$$

where $\hat{}$ again denotes the Cartesian co-ordinate system. From equation (7) we know the transformation law for covariant components, and hence

$$\varepsilon_{ijk} = \frac{\partial \hat{x}^l}{\partial x^i} \frac{\partial \hat{x}^m}{\partial x^j} \frac{\partial \hat{x}^n}{\partial x^k} \hat{\varepsilon}_{lmn} = J \hat{\varepsilon}_{ijk}, \tag{21}$$

where J is the Jacobian of the transformation. Forming the determinant from the components of equation (11), results in $g = J^2$, and therefore

$$\varepsilon_{ijk} = \sqrt{g} \hat{\varepsilon}_{ijk}. \tag{22}$$

Raising the indices in equation (22) with the metric, one obtains

$$\varepsilon^{ijk} = \sqrt{g} \hat{\varepsilon}^{ijk}. \tag{23}$$

The cross product of two vectors $\mathbf{a}$ and $\mathbf{b}$ in any co-ordinate frame is then written as

$$\mathbf{a} \times \mathbf{b} = a^i b^j \mathbf{e}_i \times \mathbf{e}_j = a^i b^j \varepsilon_{ijk} \mathbf{e}^k = \sqrt{g} \hat{\varepsilon}_{ijk} a^i b^j \mathbf{e}^k. \tag{24}$$

It should be noted that equations (22) and (23) can also be derived by starting from the well-known relation in Cartesian co-ordinates

$$\hat{\mathbf{e}}_i \times \hat{\mathbf{e}}_j = \hat{\varepsilon}_{ijk} \hat{\mathbf{e}}_k. \tag{25}$$

Insertion of

$$\hat{\mathbf{e}}_i = \frac{\partial x^l}{\partial \hat{x}^i}\mathbf{e}_l; \quad \hat{\mathbf{e}}_j = \frac{\partial x^m}{\partial \hat{x}^j}\mathbf{e}_m; \quad \hat{\mathbf{e}}_k = \frac{\partial \hat{x}^k}{\partial x^n}\mathbf{e}^n$$

into equation (25) and multiplication of the resulting left and right hand sides by

$$\frac{\partial \hat{x}^i}{\partial x^p}\frac{\partial \hat{x}^j}{\partial x^r}$$

with summation over i and j finally leads to

$$\mathbf{e}_p \times \mathbf{e}_r = \sqrt{g}\hat{\varepsilon}_{prn}\mathbf{e}^n. \tag{26}$$

With the above equations it is now possible to derive the transformation rules (i) (viii) needed for the Euler or SWEs and for the grid generation equations themselves:

(i) Gradient of a scalar function h:

$$\nabla h = \mathbf{e}^i \frac{\partial}{\partial \xi^i} h = \mathbf{e}^i h_{,i} = g^{il} h_{,i}\mathbf{e}_l. \tag{27}$$

(ii) Gradient of a vector field $\mathbf{u}$:

$$\nabla \mathbf{u} = \mathbf{e}^i \frac{\partial}{\partial \xi^i}(u^j \mathbf{e}_j) = \left(\frac{\partial u^j}{\partial \xi^i} + u^m \Gamma^j_{im}\right)\mathbf{e}^i \mathbf{e}_j, \tag{28}$$

where equation (12) was used.

(iii) Divergence of a vector field $\mathbf{u}$:

$$\nabla \cdot \mathbf{u} = \mathbf{e}^k \cdot \frac{\partial}{\partial \xi^k}(u^i \mathbf{e}_i) = \frac{\partial u^i}{\partial \xi^i} + \mathbf{e}^k \cdot \mathbf{e}_m u^i \Gamma^m_{ik}$$

$$= \frac{\partial u^i}{\partial \xi^i} + u^i \Gamma^m_{im} = u^i_{,i} + \frac{1}{2g} g_{,j} u^i = \frac{1}{\sqrt{g}}(\sqrt{g} u^j)_{,j}, \tag{29}$$

where equation (17) was used.

(iv) Cross product of two vectors (here base vectors $\mathbf{e}_i$, $\mathbf{e}_j$ since they are most often needed):

$$\mathbf{e}_i \times \mathbf{e}_j = \varepsilon_{ijk}\mathbf{e}^k = \sqrt{g}\hat{\varepsilon}_{ijk}\mathbf{e}^k = J\hat{\varepsilon}_{ijk}\mathbf{e}^k. \tag{30}$$

It should be noted that this formula directly gives an expression for the area $\mathrm{d}A^k$ and the volume $\mathrm{d}V$.

$$\mathrm{d}\mathbf{A}^k := \mathrm{d}\mathbf{s}^i \times \mathrm{d}\mathbf{s}^j; \qquad \mathrm{d}V := \mathrm{d}\mathbf{s}^k \cdot (\mathrm{d}\mathbf{s}^i \times \mathrm{d}\mathbf{s}^j)$$

where i, j, k are all different. The vector $\mathrm{d}\mathbf{s}^i$ is of the form $\mathrm{d}\mathbf{s}^i := \mathbf{e}_i \mathrm{d}\xi^i$ (no summation over i). Thus we obtain

$$|\mathrm{d}\mathbf{A}^k| = J\,\mathrm{d}\xi^i\,\mathrm{d}\xi^j; \qquad \mathrm{d}V = J\,\mathrm{d}\xi^i\,\mathrm{d}\xi^j\,\mathrm{d}\xi^k.$$

(v) Delta operator applied to a scalar function ψ:

$$\Delta\psi = \nabla \cdot (\nabla\psi) = \mathbf{e}^k \cdot \frac{\partial}{\partial \xi^k}\left(\mathbf{e}_l g^{il}\frac{\partial \psi}{\partial \xi^i}\right)$$

$$= g^{il}\psi_{,il} + \psi_{,i}\frac{1}{\sqrt{g}}(\sqrt{g} g^{ik})_k \tag{31}$$

where the components of the gradient of equation (27) were inserted into equation (29).

(vi) For transient solution areas, i.e. the solution area moves in the physical plane but is fixed in the computational domain, time derivatives in the two planes are related by

$$\left(\frac{\partial f}{\partial t}\right)_\xi = \left(\frac{\partial f}{\partial t}\right)_x + \left(\frac{\partial f}{\partial x^i}\right)_t \left(\frac{\partial x^i}{\partial t}\right)_\xi, \tag{32}$$

where $(\partial x^i/\partial t)_\xi$ are the corresponding grid speeds, which can be calculated from two consecutive grid point distributions.

(vii) Relation between covariant and contravariant components of the metric tensor:

$$g^{ik} = \frac{\Delta^{ik}}{g}, \tag{33}$$

where Δ^{ik} denotes cofactor (i, k) of the matrix of the metric tensor. In two-dimensions we have

$$g^{11} = g^{-1} g_{22}; \quad g^{12} = g^{21} = -g^{-1} g_{12}; \quad g^{22} = g^{-1} g_{11}. \tag{34}$$

(viii) Relationship between grid point distribution and Christoffel symbols, equation (16):

$$\Gamma^i_{jk} = \frac{\partial \xi^i}{\partial x^l} \frac{\partial^2 x^l}{\partial \xi^j \partial \xi^k}. \tag{35}$$

There are 6 Christoffel symbols in two-dimensions and 18 in three-dimensions. Storage of Christoffel symbols for a staggered grid in three-dimensions will most likely not be possible since $(18 + 1) \times 4 \times N$ computer words (the 1 comes from the Jacobian) would be necessary where N is the total number of grid points (N could be the number of elevation points if SWEs were solved). If metric coefficients are used in three-dimensions, $(6 + 1) \times 4 \times N$ words were necessary. For a realistic value of some 20,000 grid points in three-dimensions, this requirement exceeds available storage for most machines. If interpolation of metric coefficients (dangerous) is used, the factor 4 can be eliminated. Even then, the storage needed is an order of magnitude higher than the number of grid points. Thus, one should consider the recalculation of the Christoffel symbols or the metric coefficients at each time step, using a vectorizable scheme.

4. NUMERICAL GRID GENERATION

We first consider the transformation of Poisson's equation. We start with a co-ordinate system $x = x(\xi, \eta)$, $y = y(\xi, \eta)$. The covariant base vectors are given by

$$\mathbf{e}_1 = (x_\xi, y_\xi)^{\mathrm{T}}; \qquad \mathbf{e}_2 = (x_\eta, y_\eta)^{\mathrm{T}} \tag{36}$$

that is, $\mathbf{e}_1$ is tangential to an η-line and $\mathbf{e}_2$ is tangential to a ξ-line. The superscript T means transposed, i.e. a vector is considered to be a one-column matrix. The components of the metric tensor are obtained from equation (8)

$$g_{11} = x_\xi^2 + y_\xi^2; \quad g_{12} = g_{21} = x_\xi x_\eta + y_\xi y_\eta; \quad g_{22} = x_\eta^2 + y_\eta^2. \tag{37}$$

The contravariant components are

$$g^{11} = \frac{g_{22}}{g}; \quad g^{12} = g^{21} = \frac{-g_{12}}{g}; \quad g^{22} = \frac{g_{11}}{g}. \tag{38}$$

The contravariant base vectors $\mathbf{e}^i$ are determined from equation (4). Using the matrix

$$\mathbf{E}:=\begin{pmatrix} x_\xi & x_\eta \\ y_\xi & y_\eta \end{pmatrix}=(\mathbf{e}_1,\mathbf{e}_2), \tag{39}$$

the inverse matrix is given by

$$\mathbf{E}^{-1}=\frac{1}{J}\begin{pmatrix} y_\eta & -x_\eta \\ -y_\xi & x_\xi \end{pmatrix}. \tag{40}$$

We find, since $\mathbf{e}^i$ are the row vectors,

$$\mathbf{e}^1=(y_\eta,-x_\eta)^{\mathrm{T}}/J; \qquad \mathbf{e}^2=(-y_\xi,x_\xi)^{\mathrm{T}}/J. \tag{41}$$

$\mathbf{e}^1$ is perpendicular to $\mathbf{e}_2$, $\mathbf{e}^2$ is perpendicular to $\mathbf{e}_1$ but $\mathbf{e}^1$ and $\mathbf{e}_1$ are in general not perpendicular. Since $\mathbf{E}^{-1}$ is also given by

$$\mathbf{E}^{-1}=\begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix}, \tag{42}$$

we have

$$\mathbf{e}^1=(\xi_x,\xi_y)^{\mathrm{T}}; \qquad \mathbf{e}^2=(\eta_x,\eta_y)^{\mathrm{T}}. \tag{43}$$

We now reconsider the grid generation equations

$$\Delta\xi=P; \qquad \Delta\eta=Q. \tag{44}$$

Using (v), the transformation equation of the Laplacian, and the fact that ξ and η are co-ordinates themselves, we find

$$\Delta\xi=g^{ik}(\xi_{,i,k}-\Gamma^j_{ik}\xi_{,j})=-g^{ik}\mathbf{e}_{i,k}\cdot\mathbf{e}^1, \tag{45a}$$

$$\Delta\xi=\frac{-1}{\sqrt{g}}[(y_\eta(g^{11}x_{\xi\xi}+2g^{12}x_{\xi\eta}+g^{22}x_{\eta\eta})-x_\eta(g^{11}y_{\xi\xi}+2g^{12}y_{\xi\eta}+g^{22}y_{\eta\eta})]=P. \tag{45b}$$

In a similar way we obtain for the η-co-ordinate

$$\Delta\eta=\frac{+1}{\sqrt{g}}[y_\xi(g^{11}x_{\xi\xi}+2g^{12}x_{\xi\eta}+g^{22}x_{\eta\eta})-x_\xi(g^{11}y_{\xi\xi}+2g^{12}y_{\xi\eta}+g^{22}y_{\eta\eta})]=Q. \tag{46}$$

By means of (vii), the g^{ij} can be expressed in terms of g_{ij}. For stability reasons in the numerical iterative solution of the above system of equations, the above equations are rewritten in the form

$$g_{22}x_{\xi\xi}-2g_{12}x_{\xi\eta}+g_{11}x_{\eta\eta}+g(x_\xi P+x_\eta Q)=0, \tag{47}$$

$$g_{22}y_{\xi\xi}-2g_{12}y_{\xi\eta}+g_{11}y_{\eta\eta}+g(y_\xi P+y_\eta Q)=0. \tag{48}$$

These equations are quasilinear, where the non-linearity appears in the expressions for the metric coefficients.

The grid generation can be extended to two-dimensional co-ordinates on parametrized surfaces. The derivation of these equations is found in Reference 34. The fact that boundary curves are grid lines does not uniquely determine the co-ordinate system. Further requirements of the grid point distribution can be demanded, such as equidistance of grid points or orthogonality of grid lines. For an orthogonal grid it is necessary that boundary curves intersect at right angles. For the construction of orthogonal grids we refer to References 34–37. For internal flow, however,

it seems that orthogonal grids are not well suited. Since certain features of a grid must be globally satisfied, grids can also be obtained from variational principles.[34,38]

If we set control functions P and Q equal to zero in equations (47) and (48), we obtain the transformed Laplace equation. To visualize the distribution of grid lines by solution of Laplace's equation, one can assume that each grid point is coupled to its four nearest neighbours by springs having equal spring constants (Figure 11). The equilibrium of that spring model gives the distribution of co-ordinate points. If the distribution of boundary points is equidistant, one observes the following fact (which can be directly shown by explicit solution of Laplace's equation): A convex boundary (Figure 12(a)), repels grid lines, whereas a concave boundary, (Figure 12(b)), attracts grid lines.

Very often, however, the opposite is desirable. For example, when a storm surge model is used for the calculation of the water level in a bay (convex area), high resolution near the shore is required. The simplest way to achieve this would be the change of the distribution of boundary points, i.e. to choose a dense distribution along the shoreline which forms the boundary of the bay. Since the Laplacian smoothes the effects coming from the boundary, the grid point distribution in the interior is not affected and remains nearly equidistant. For that reasons a Poisson equation is needed and control functions have to be determined.

Since a boundary is either a ξ- or an η-line (in two dimensions), all partial derivatives in the other direction vanish. At the moment we assume that ξ only depends on x, and η only depends on y. If the boundary is not a straight line, x and y must be replaced by arc length s. If we determine P from a boundary formed by an η-line (i.e. ξ varies) and Q from a boundary which is a ξ-line, we find from equations (47) and (48)

$$g^{11}x_{\xi\xi} = -x_\xi P; \qquad g^{22}y_{\eta\eta} = -y_\eta Q. \tag{49}$$

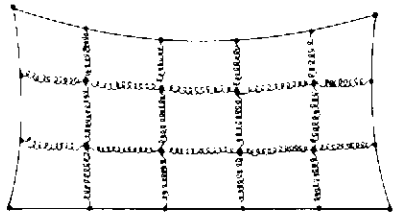

Figure 11. Grid point distribution obtained from Laplace's equation; exemplified by spring model

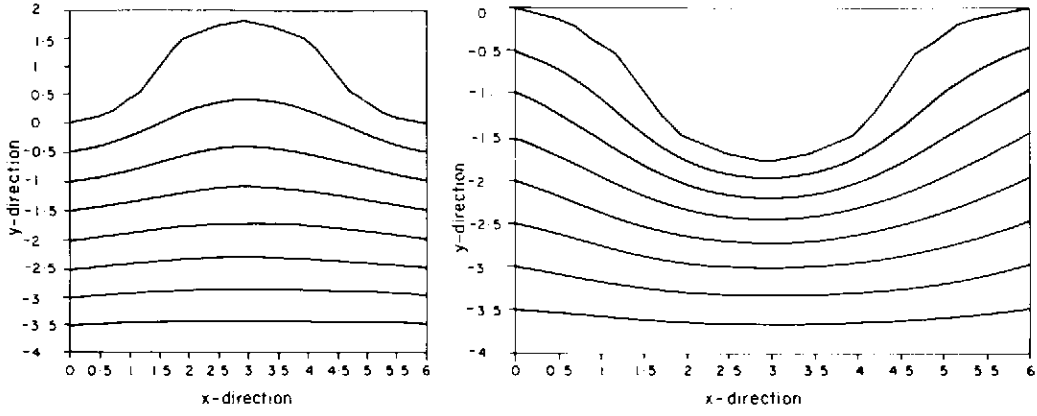

Figure 12. Grid line distribution for convex (a) and concave (b) solution areas

Since $x_{\xi\xi}$, x_ξ as well as $y_{\eta\eta}$, y_η can be calculated from the boundary point distribution, equations (49) can be solved for P and Q, yielding

$$P := -g^{11}\frac{x_{\xi\xi}}{x_\xi}; \qquad Q := -g^{22}\frac{y_{\eta\eta}}{y_\eta}. \tag{50}$$

The use of these control functions guarantees that the specified boundary point distribution is obtained. Moreover, if these control functions are employed in the interior of the solution domain, this yields the same grid point distribution along ξ- and η-lines as on the boundary. Figure 13 shows the grid line distribution resulting from a non-equidistant boundary point distribution with and without the use of control functions. If boundary point distributions on top and bottom (or left and right) are different, the values of the control functions in the interior can be determined by linear interpolation. For curvilinear boundaries arc length s replaces x and y.

According to what was said above, a higher grid line concentration near the inner boundary of an annular ring is expected. In order to have an equidistant distribution, control functions have to be specified. Without control functions, the Laplace operator expressed in polar co-ordinates takes the form

$$\Delta\psi = \psi_{,rr} + \frac{1}{r}\psi_{,r} + \psi_{,\varphi\varphi}. \tag{51}$$

For an equidistant distribution in radial direction ψ must be of the form $\psi = ar + b$. Let ψ satisfy the boundary condition $\psi = 1$ for $r = r_1$ and $\psi = N$ for $r = r_2$. The solution of equation (51) is then

$$\psi = 1 - \frac{1 - N}{r_1 - r_2}(r_1 \quad r). \tag{52}$$

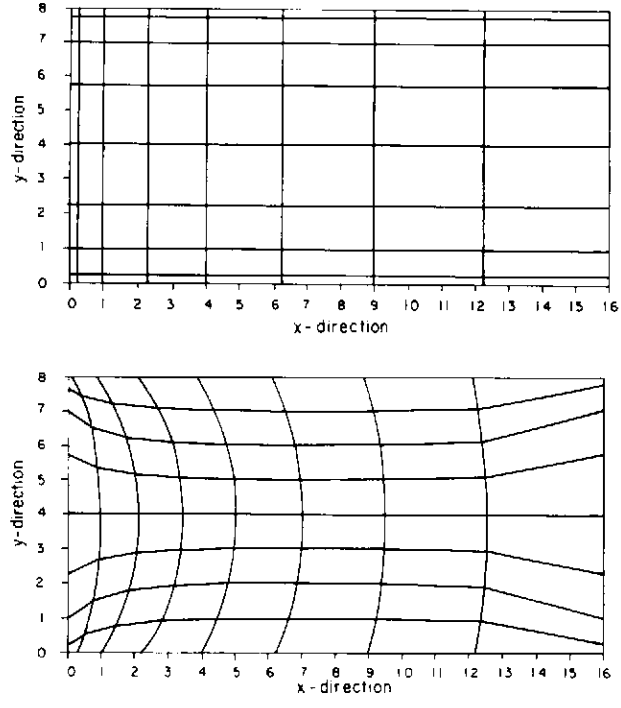

Figure 13. Grid line distribution for a rectangle with and without the use of control functions

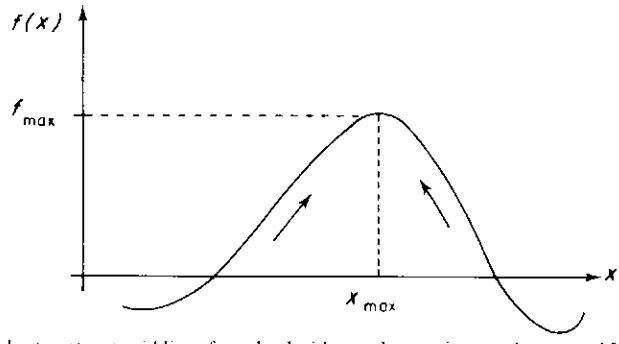

Figure 14. In order to attract grid lines from both sides to the maximum, the control function P must satisfy $P > 0$ at the left of the maximum and $P < 0$ at the right

Hence we obtain for the control function P

$$P = \Delta\psi = \frac{\psi_r}{r} = \frac{1}{r}\frac{N}{r_2 - r_1} = \frac{N}{\Delta r}\frac{1}{r}\kappa$$

where $\Delta r = r_2 - r_1$ and κ denotes curvature. Curvature is calculated on boundaries $\xi = 1$ and $\xi = N$ and determined from linear interpolation for the interior. In the same way control function Q is determined. In general, curvature is determined from

$$\kappa = (x'y'' - y'x'')/(x'^2 + y'^2)^{3/2}, \tag{53}$$

where any curve C is given by $(x(t), y(t))^T$ and $'$ denotes derivation with respect to t. In our case, boundary curves are parametrized by either ξ or η. Since the control functions are computed from a monotonic function, no singularities can arise.

Grid lines can also be attracted to extrema of the physical solution or to regions of high gradients. First, the one-dimensional case is considered. In order to attract grid lines, P must have the following properties (Figure 14). To the left of the maximum, grid lines must be moved in the direction of higher x values, that is $P > 0$ is necessary (this follows directly from solution of equation $\xi_{xx} = a$ for $a < 0$ and $a > 0$). To the right of the maximum, we have $P < 0$. A function which gives the right signs is the derivative f_x. Since $f_x = 0$ at the maximum, lines are moved only up to the maximum. Near f_{max}, f_x is small, which means that lines in the vicinity of f_{max} are only slightly moved. f_x is multiplied by f which serves as a weight function. For $P = ff_x$, we find from

$$\xi_{xx} = P = ff_x \tag{54}$$

that

$$\xi = \frac{1}{2}\int f^2\,dx, \tag{55}$$

which is a monotonic function in x. To concentrate grid lines near gradients, P has the form $P = f_x f_{xx}$. For the extension to two dimensions, f is differentiated in the direction of the co-ordinate lines ξ and η. f_x is replaced by $f_\xi/\sqrt{g^{11}}$ and $f_\eta/\sqrt{g^{22}}$. The factors $1/\sqrt{g^{11}}$ and $1/\sqrt{g^{22}}$ ensure that the slope is measured in physical units and not in grid units. If the second derivatives are approximated by $f_{\xi\xi}/g^{11}$ and $f_{\eta\eta}/g^{22}$, the control functions written in analogy to equation (54) assume the form

$$P = c(\xi, \eta) f_\xi f_{\xi\xi}/(g^{11})^{3/2} = c(\xi, \eta) f_\xi f_{\xi\xi} J^3/(g_{22})^{3/2}, \tag{56}$$

$$Q = d(\xi, \eta) f_\eta f_{\eta\eta}/(g^{22})^{3/2} = d(\xi, \eta) f_\eta f_{\eta\eta} J^3/(g_{11})^{3/2}, \tag{57}$$

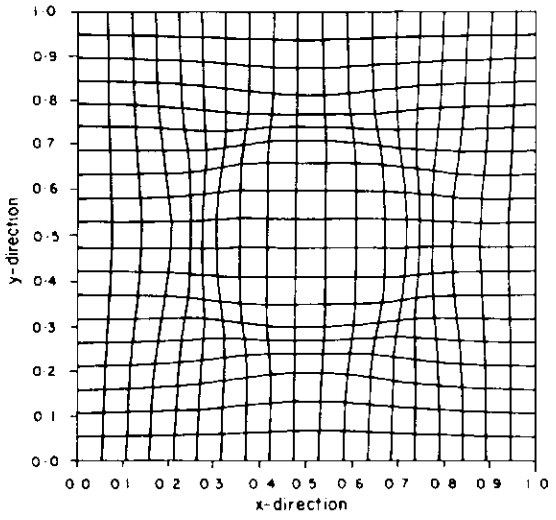

Figure 15. Concentration of grid lines to a circle. The initial grid is a square grid

where the functions c and d are additional normalizing functions. The term $1/J^2$ eliminates the factor which arises in the transformed equations (48). As an example of the control functions, Figure 15 depicts the concentration of grid lines to the extrema of the function

$$f = \alpha \tan h[D(r - \tfrac{1}{2}) - \tfrac{1}{4}]. \tag{58}$$

The initial grid is a square grid. The highest grid line density is for a circle of radius $\frac{1}{4}$ with the origin at $(\frac{1}{2}, \frac{1}{2})$, which is the location of the maximum of the gradient. The magnitude of the gradient is changed by α and D.

5. NUMERICAL SOLUTION OF GRID GENERATION EQUATIONS

In order to obtain the grid point distributions, equations (47) and (48) have to be solved numerically. To this end, the functions x, y, P and Q have to be discretized. Central differences are used for first derivatives:

$$(x_\xi)_{i,j} = \tfrac{1}{2}(x_{i+1,j} - x_{i-1,j}), \tag{59}$$

where $i \in [1, N]$, $j \in [1, M]$. First derivatives are needed for the components of the metric tensor. Second derivatives are discretized as follows:

$$\begin{aligned}
(x_{\xi\xi})_{i,j} &= x_{i+1,j} - 2x_{i,j} + x_{i-1,j}, \\
(x_{\xi\eta})_{i,j} &= \tfrac{1}{4}(x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}), \\
(x_{\eta\eta})_{i,j} &= x_{i,j+1} - 2x_{i,j} + x_{i,j-1}.
\end{aligned} \tag{60}$$

Using the denotations α, 2β, γ for the discretized forms of g_{22}, g_{12} and g_{11} and writing J^2 instead of g, equation (47) can be solved for $x_{i,j}$, resulting in the following system of equations:

$$\begin{aligned}
x_{i,j} = (&\alpha_{i,j}(x_{i-1,j} - x_{i-1,j}) \\
&- \beta_{i,j}(x_{i+1,j+1} - x_{i-1,j+1} - x_{i+1,j-1} + x_{i-1,j-1}) \\
&+ \gamma_{i,j}(x_{i,j+1} - x_{i,j-1}) + \tfrac{1}{2}J_{i,j}^2 P_{i,j}(x_{i+1,j} - x_{i-1,j}) \\
&+ \tfrac{1}{2}J_{i,j}^2 Q_{i,j}(x_{i,j+1} - x_{i,j-1}))/(2(\alpha_{i,j} + \gamma_{i,j})),
\end{aligned} \tag{61}$$

Table I. Variation with ω for the number of iterations for the free surface problem

ω	n	ω	n	ω	n	ω	n
1·45	83	1·70	42	1·75	36	1·80	41
1·50	74	1·71	41	1·76	37	1·85	64
1·55	66	1·72	39	1·77	38	1·90	92
1·60	58	1·73	38	1·78	39	1·95	150
1·65	50	1·74	36	1·79	40		

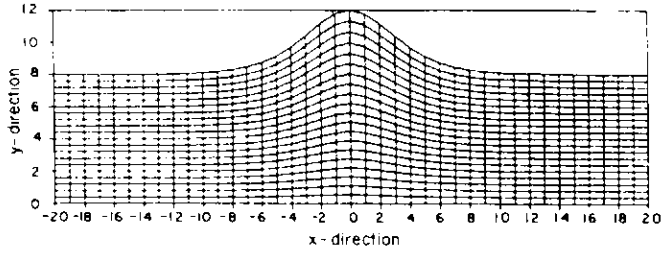

Figure 16. Time dependent solution domain for the free surface problem. The number of iterations for the generation of this grid strongly depends on ω, as depicted in Table I.

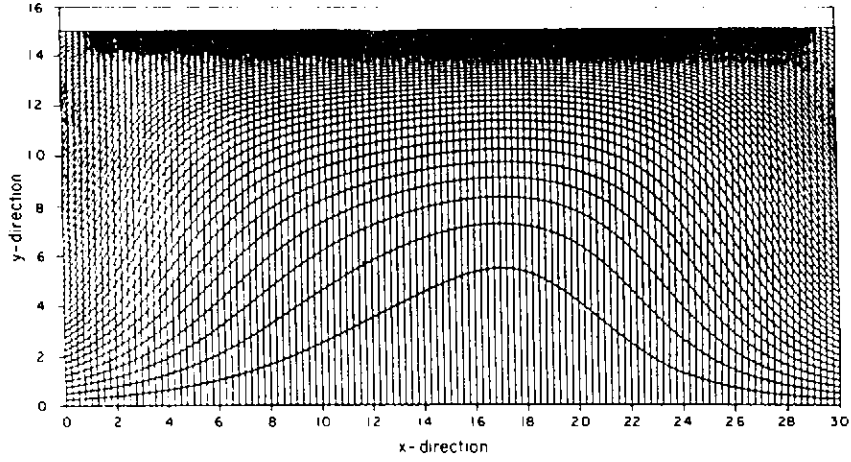

Figure 17. 'Converged' solution for grid generation equations on a 125 × 65 grid. As start values, the co-ordinate values of the north side were used

where i, j vary from 2 to $N-1$ and from 2 to $M-1$, respectively. A similar equation holds for $y_{i,j}$. This system of non-linear equations is solved by iteration. In order to increase convergence speed, SOR (successive overrelaxation) is used. Values from the previous iteration, x_{old}, are compared with those of the current iteration, x_{it}. The differences of these values are amplified by a factor ω to force the solution in this direction. The new values, x_{new}, are calculated from the formula

$$x_{\text{new}} = x_{\text{old}} + \omega(x_{it} - x_{\text{old}}); \qquad 1 \leqslant \omega \leqslant 2. \tag{62}$$

As is known from numerical experience, the convergence speed is susceptible to the value of ω, in particular for time-dependent domains. Table I shows the numbers of iterations for different

values of ω for the free surface problem (Figure 16). Although it is not necessary to solve equations (61) to a high degree of accuracy, it is necessary that the numerical error is sufficiently damped during each iteration. Since grid point i (one dimension considered for the moment) depends only on neighbouring points $i-1, i+1$, convergence at i is determined by the difference of the errors at these points. If the errors have the same sign, the error at i cannot vanish but is smoothed (averaged). Considering the Fourier components of the error function, one obtains from local mode analysis[19] that wave lengths of 2Δ (Δ is grid spacing) are highly damped while error components of long wave lengths are nearly unaffected, since many iterations are needed to propagate improved values through the grid because only direct neighbours are coupled.

Fourier transformation is not possible for the non-linear equations (61) but numerical experiments show that damping of numerical error may be so slow for large grids (see Figure 17) that the criterion for ending the iteration process is reached before a satisfactory solution is computed. Figure 17 shows an attempt at solving the transformed grid generation equations for a rectangle where a numerical error due to large wavelength components is present. An equidistant grid was the expected solution. If equations (61) are solved on grids with mesh spacings..., 8Δ, 4Δ, 2Δ, Δ, each of these grids eliminates the error in a certain frequency range. This method is called the multigrid technique.[39-42] Using the FAS (full approximation scheme[39]), the above problem was solved in 1 second on a Siemens 7875 computer.

It is felt that combination (where possible) of grid generation techniques and multigrid methods will substantially increase computational efficiency. It would be worth while to investigate the possible applications to transient fluid dynamics problems.

The following is a summary of the steps for the grid generation program. First, it is necessary to input the boundary co-ordinate values along with the chosen segment structure. If necessary, the program calculates the values of the control functions from the boundary point distribution and solves equations (61). Only one iteration sweep per segment is performed where the sequence of the segments is determined from the specified input order. Segments which have a side in common are connected by an overlap of either one row or column of grid points;[19] i.e. the boundary points of one segment are interior points in the neighbouring segment and vice versa. Hence boundary points on segment sides are automatically updated after each iteration sweep. The next module calculates the metric coefficients from the computed grid point positions. These values are then used in the solution module which will be discussed in the companion paper (Part II).

6. CONCLUSIONS AND OUTLOOK

In this paper a description of the basic ideas and properties of BFGs was given. Furthermore, the general transformation laws for grid generation and physical equations were presented. It was shown that for large grids convergence problems in the solution of the grid generation equations can occur which can be overcome by use of the FAS multigrid technique.

Since orthogonal grids do not seem to be advantageous for internal flow problems, their construction was not addressed.[43-44] It is believed that the use of composite grids can be directly extended to three-dimensional grids[45,46] which are important for environmental flow problems. In a companion paper the application of the generated grids to selected two-dimensional fluid flow problems will be discussed.[47]

ACKNOWLEDGEMENT

The authors are grateful for NATO research grant 681/84. The first author is also grateful to the GKSS-Research Centre for supporting this work.

REFERENCES

1. N. Praagman, 'Numerical solution of the shallow water equations by a finite element method', *Dissertation*, T. H. Delft, 1979.
2. C. Taylor, *Finite Element Programming of the Navier–Stokes Equations*, Pineridge Press, Swansea, 1981.
3. K. P. Holz *et al.*, 'Verifikation eines numerischen Tidemodells', *Wasserwirtschaft 71*, **10**, 289–294 (1981).
4. W. C. Thacker, 'Irregular grid finite difference techniques simulations of oscillations in shallow circular basins', *J. Phys. Ocean.*, **7**, 282–292 (1977).
5. R. D. Richtmyer, *Principles of Advanced Mathematical Physics, Vol. 2*, Springer-Verlag, 1981.
6. N. Prakash, *Differential Geometry: An Integrated Approach*, Tata McGraw-Hill, 1981.
7. R. U. Sexl and H. K. Urbantke, *Gravitation und Kosmologie*, Bibliographisches Institut, 1983.
8. J. A. Thorpe, *Elementary Topics in Differential Geometry*, Springer-Verlag, 1979.
9. E. Kreyszig, *Differential Geometry*, University of Toronto Press, 1970.
10. P. R. Eiseman, 'Geometric methods in computational fluid dynamics', *ICASE Report No-80-11*, 1980.
11. Z. U. A., Warsi, *Tensors and Differential Geometry Applied to Analytic and Numerical Coordinate Generation*, Mississippi State University, MSSU-EIRS 81–1, 1981.
12. J. Haeuser *et al.*, 'Analysis of thermal impact in tidal rivers and estuaries', *Water Research*, **14**, 1409–1419 (1980).
13. A. Jameson, in *Short Course in Numerical Grid Generation*., J. F. Thompson (Ed.) Mississippi State University, 1984.
14. J. F. Thompson and Z. U. A. Warsi, 'Three dimensional grid generation from elliptic systems', *AIAA-83-1905*, 1983.
15. J. F. Thompson, 'Wescor—Boundary-fitted coordinate code for general 2D regions with obstacles and boundary intrusions', *ARO Report 82–3*, 1982.
16. J. F. Thompson, 'General curvilinear coordinate systems', in J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982, 1–30.
17. Z. U. A. Warsi, 'Basic differential models for coordinate generation', in J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982, 1–30.
18. Z. U. A. Warsi, 'Generation of three-dimensional grids through elliptic differential equations', *Von Karman Institute of Fluid Mechanics, Lecture Series 1984–05*, 1984.
19. M. R. Coleman, 'Generation of boundary fitted coordinate systems using segmented computational regions', in J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982, 633–652.
20. C. W. Mastin, 'Error induced by coordinate systems', in J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982, 631–40.
21. P. R. Eiseman, 'Adaptive Grid Generation by mean value relaxation', *Advances in Grid Generation*, ASME, FED-Vol. 5, 1983, in U. Ghia and K. W. Ghia (eds).
22. S. Nakamura, 'Marching grid generation using parabolic differential equations', in J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982, 775–786.
23. H. A. Dwyer, in J. F. Thompson (ed.), *Short Course in Numerical Grid Generation*, Mississippi State University, 1984.
24. R. E. Smith, 'Introduction to algebraic grid generation', in J. F. Thompson (ed.), *Short Course in Numerical Grid Generation*, Mississippi State University, 1984.
25. R. E. Smith 'Algebraic grid generation', in J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982, 139–170.
26. J. F. Thompson (ed.), *Numerical Grid Generation*, North Holland, 1982.
27. R. E. Smith (ed.), *Numerical Grid Generation Techniques*, NASA Conference Publication 2166, NASA Langley Research Center, 1980.
28. K. Ghia and U. Ghia (eds), *Advances in Grid Generation*, ASME, FED-Vol. 5, 1983.
29. D. A. Anderson *et al.*, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corporation, Washington, 1984, Chapter 10.
30. W. C. Thacker, 'A brief review of techniques for generating irregular computational grids', *Int. j. numer. methods. eng.*, **15**, 1335–1341 (1980).
31. J. F. Thompson *et al.*, *Foundation of Numerical Grid Generation*, North Holland, 1985.
32. J. F. Middlecoff and P. D. Thomas, 'Direct control of the grid point distribution in meshes generated by elliptic equations', *AIAA 4th Comp. Fluid Dynamics Conference*, 79–1462, 1979, 175–179.
33. H. G. Paap and J. Hacuser, 'Automatic grid doubling for composite meshes', *Sixth Gamm Conference on Numerical Methods in Fluid Mechanics*, Goettingen, September 1985, 25–27.
34. H. G. Paap, 'Loesung der zwei-dimensionalen Navier–Stokes Gleichungen mit Hilfe randangepasster Koordinaten', *Diploma Thesis*, Universitet Hamburg, 1984.
35. Z. U. A. Warsi and J. F. Thompson, 'A noniterative method for generation of orthogonal coordinates in doubly-connected regions', *Mathematics of Computation*, **38**, (158), 501–516 (1982).
36. H. J. Haussling and R. M. Coleman, 'A method for generation of orthogonal and nearly orthogonal boundary fitted co-ordinate systems', *J. Comp. Phy.*, **43**, 373–381 (1981).
37. D. C. Mobley and R. J. Stewart, 'On the numerical generation of boundary-fitted orthogonal curvilinear coordinate systems', *J. Comp. Physics*, **34**, 124–135 (1980).
38. J. U. Brackbill, 'Coordinate system control: adaptive meshes', in J. F. Thompson (ed.) *Numerical Grid Generation*, North Holland, 1982, 277–294.

39. A. Brandt, *Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics*, Von Karman Institute of Fluid Dynamics, Lecture Series 1984 04, 1984.
40. K. Stueben and U. Trottenberg, 'Multigrid methods, fundamental algorithms, model problem analysis and applications', *Lecture Notes in Mathematics*, 760, Springer-Verlag, 1982, 1–176.
41. U. Trottenberg, *Schnelle Loesung Elliptischer Differentialgleichunen nach dem Mehrgitterprinzip*, Arbeitspapiere der GMD, 40, Bonn, 1983.
42. H. Rieger *et al.*, 'Fast iterative solution of Poisson equation with Neumann boundary conditions in nonorthogonal curvilinear coordinate systems by a multi-grid method', *Numerical Heat Transfer*, **6**, 1–15 (1983).
43. G. Ryskin and L. G. Leal, 'Orthogonal mapping', *J. Comp. Phys.*, **50**, 71–100 (1983).
44. W. C. Davies, 'An initial value approach to the production of discrete orthogonal coordinates', *J. Comp. Phy.*, **39**, 164–178 (1981).
45. K. Miki and T. Takagi, 'A domain decomposition and overlapping method for the generation of three-dimensional boundary fitted coordinate systems', *J. Comp. Phys.*, **53**, 319–330 (1984).
46. Z. U. A. Warsi, 'Generation of three-dimensional grids through elliptic differential equations', Von Karman Institute of Fluid Dynamics, *Lecture Series 1984–04*, 1984.
47. J. Haeuser *et al.*, 'Boundary conformed co-ordinate systems for selected two-dimensional fluid flow problems, Part II: Application of the BFG Method to Selected Problems in Two-dimensional Fluid Flow', *Int. j. numer. methods fluids*, **6**, 529–539 (1986).