INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUID, VOL. 6, 291–304 (1986)

AN ALGEBRAIC GRID GENERATION TECHNIQUE FOR TIME-VARYING TWO-DIMENSIONAL SPATIAL DOMAINS

S.-L. YANG AND T. I-P. SHIH

Department of Mechanical Engineering, University of Florida, Gainesville, FL 32611, U.S.A.

SUMMARY

An efficient and versatile algebraic grid generation technique is presented for generating grid points in irregularly shaped and time-varying spatial domains. The method presented is based on the 'two-boundary technique' of Smith. The usefulness and the feasibility of the grid generation technique were demonstrated by (1) generating grid points inside one of the combustion chambers of a motored two-dimensional rotary engine and (2) obtaining numerical solutions for the flow field inside one of those combustion chambers.

KEY WORDS Grid Generation Numerical Methods

INTRODUCTION

In order to obtain finite-difference solutions to partial differential equations (PDEs) that described fluid flow problems, the spatial domain of the problem must be replaced by a system of grid points. For computational efficiency, the number of grid points used should be kept to the minimum that is required to resolve spatially all significant features of the flow. For spatial domains with deforming and/or irregularly shaped boundaries and for flow problems with disparate length scales in different parts of the flow, the ideal distribution of grid points is usually non-uniform and changes with time.

To facilitate the finite-difference method of solution (in particular, implementation of boundary conditions, algorithm development and programming), the moving and non-uniformly distributed grid points in the spatial domain are generally mapped onto a transformed domain in which they are stationary and uniformly distributed. In addition, grid lines as well as the boundaries of the spatial domain are mapped so that they correspond to co-ordinate lines in the transformed domain. As a result the transformed domain is either rectangular in shape or comprised of rectangular subregions.

In practice, since grid points are stationary and uniformly distributed in the transformed domain (i.e. the locations of the grid points are known precisely), the mapping is from the transformed domain to the 'physical' spatial domain. For this reason, the mapping of grid points is known as grid generation, i.e. the generation of grid points in the 'physical' spatial domain.

The mapping of grid points, or grid generation, is achieved by what are known as grid generation techniques. There are many different grid generation techniques and they are classified as either differential equation methods or algebraic methods.[1–6] Differential equation methods require PDEs to be solved. Algebraic methods do not require the solutions of PDEs. As a result, algebraic methods are computationally much more efficient than differential equation methods. This is especially true for unsteady problems involving grid points that move in the spatial domain.

0271–2091/86/050291–14$07.00

Received 1 September 1985

the 'two-boundary technique'. This technique can generate non-uniformly distributed, but stationary, grid points in two- or three-dimensional spatial domains of fairly arbitrary shapes. In this paper, we extend the 'two-boundary technique' of Smith so that in addition to non-uniform distribution, the grid points can also move in the spatial domains. This extension is important to problems involving time-varying or deforming spatial domains and to problems in which regions of sharp gradients move about.

To demonstrate the usefulness and the feasibility of the extended 'two-boundary technique' developed here, (1) grid points were generated inside the combustion chambers of a motored two-dimensional rotary engine and (2) numerical solutions were obtained for the flow inside one of those combustion chambers. Since the spatial domain represented by the combustion chambers of a motored rotary engine deforms considerably both in shape and in size, this domain and the flow that occurs in it serve as a good test of the grid generation technique presented here.

Here, we note that the extended 'two-boundary technique' is demonstrated via a two-dimensional, time-varying spatial domain. However, the ideas presented here can readily be extended to time-varying, three-dimensional spatial domains.

In the next section, the equations describing the combustion chambers of a motored two-dimensional rotary engine are given. Afterwards, the grid generation technique developed here is described in detail. Finally, the grid points generated and the solutions obtained are presented.

A TIME-VARYING SPATIAL DOMAIN

A motoring two-dimensional rotary engine is shown in Figure 1. It consists of a housing, a rotor, a shaft and three apexes. Between the rotor, the housing and the apexes are three voids called combustion chambers. As the rotor rotates inside the housing, the combustion chambers move as well as deform.

The Cartesian co-ordinates X_1 and Y_1 of the inner surface of the housing (denoted as surface 1 in

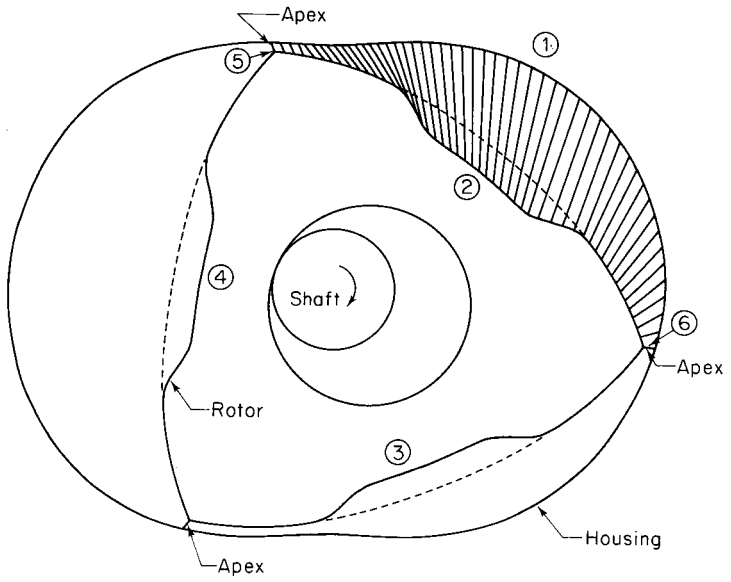

Figure 1. The rotary engine geometry

Figure 1) are described by the following parametric equations involving the parameter A:[8,9]

$$X_1 = E\cos(3A) + (R + C)\cos(A), \tag{1}$$

$$Y_1 = E\sin(3A) + (R + C)\sin(A), \tag{2}$$

where A varies between 0 and 2π. E, R and C determine the shape of the rotary engine. Here E, R and C were set equal to 0·015 m, 0·1045 m and 0·005 m, respectively.

The Cartesian co-ordinates X_2 and Y_2 of the outer surface of the rotor (denoted as surfaces 2, 3 and 4 in Figure 1) are described by another set of parametric equations involving the parameter V:[8,9]

$$X_2 = E\sin(\theta) + X_{R0}\cos(\theta/3) + Y_{R0}\sin(\theta/3), \tag{3}$$

$$Y_2 = E\cos(\theta) + Y_{R0}\cos(\theta/3) - X_{R0}\sin(\theta/3), \tag{4}$$

where

$$X_{R0} = X_R\cos(\pi/6) + Y_R\sin(\pi/6), \tag{5}$$

$$Y_{R0} = Y_R\cos(\pi/6) + X_R\sin(\pi/6), \tag{6}$$

$$X_R = R\cos(2V) - (3E^2/R)\sin(6V)\sin(2V) + 2E[1 - (9E^2/R^2)\sin^2(3V)]^{1/2}\cos(3V)\cos(2V) - P_x, \tag{7}$$

$$Y_R = R\sin(2V) - (3E^2/R)\sin(6V)\cos(2V) + 2E[1 - (9E^2/R^2)\sin^2(3V)]^{1/2}\cos(3V)\sin(2V) - P_y, \tag{8}$$

$$P_x = P\cos(2V), \tag{9}$$

$$P_y = P\sin(2V), \tag{10}$$

$$P = \begin{cases} 0 & V_1 \leqslant V \leqslant V_T, \\ \frac{P_1}{2}\{1 - \cos[\pi(V_1 - V)/(V_1 - V_2)]\} & V_2 \leqslant V \leqslant V_1, \\ P_2 + (P_1 - P_2)(V - V_3)/(V_2 - V_3) & V_3 \leqslant V \leqslant V_2, \\ \frac{P_2}{2}\{1 - \cos[\pi(V - V_4)/(V_3 - V_4)]\} & V_4 \leqslant V \leqslant V_3, \\ 0 & V_L \leqslant V \leqslant V_4. \end{cases} \tag{11}$$

The above equations represent surface 2 when the parameter V is between $V_L = \pi/6$ and $V_T = \pi/2$. When V is between $V_L = 5\pi/6$ and $V_T = 7\pi/6$, the above equations represent surface 3. Finally, when V is between $V_L = 3\pi/2$ and $V_T = 11\pi/6$, the above equations represent surface 4. In the above equations, E and R take on the same values as those used in equations (1) and (2). P_1, P_2, V_1, V_2, V_3 and V_4 describe the shape of the rotor pocket. Their definitions are given in Figure 2. Here $P_1 = P_2 = 0{\cdot}005$, $V_1 = V_L + 0{\cdot}75(V_T - V_L)$, $V_2 = V_L + 0{\cdot}625(V_T - V_L)$, $V_3 = V_L + 0{\cdot}375(V_T - V_L)$ and $V_4 = V_L + 0{\cdot}25(V_T - V_L)$. The crank angle, θ, is a function of time (t) and is given by

$$\theta = \int_0^t \Omega\,dt = \Omega t, \tag{12}$$

where Ω is the angular speed of the shaft. Here Ω was set equal to 5000 revolutions per minute.

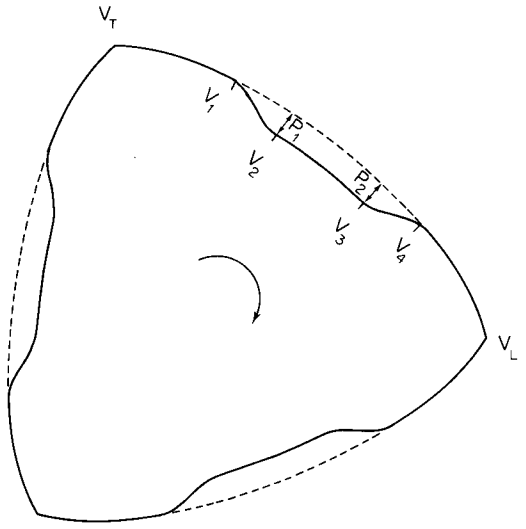

Figure 2. The rotor pocket geometry

Because of the symmetry of this problem, we only wish to generate grid points inside one of the three combustion chambers as it moves inside the rotary engine. The combustion chamber chosen is the shaded region shown in Figure 1.

GRID GENERATION TECHNIQUE

For two-dimensional time-varying spatial domains, the mapping of grid points, or grid generation, involves the following co-ordinate transformation:

$$(X, Y, t) \rightleftarrows (\xi, \eta, \tau), \tag{13}$$

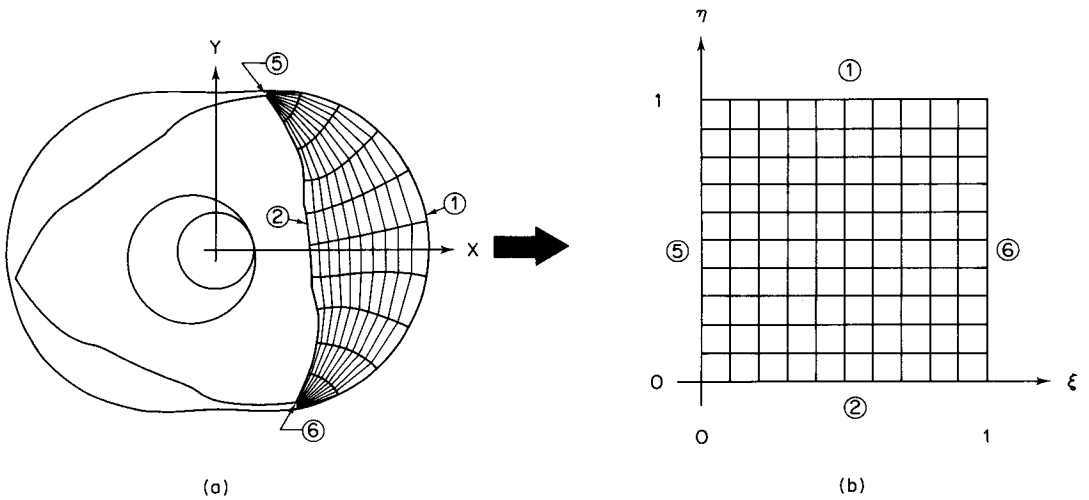

Figure 3. (a) The 'physical' spatial domain. (b) The transformed domain

or

$$t = t(\tau), \tag{14}$$

$$X = X(\xi, \eta, \tau), \tag{15}$$

$$Y = Y(\xi, \eta, \tau), \tag{16}$$

where X, Y and t represent the 'physical' domain and ξ, η and τ represent the transformed domain (see Figure 3).

Similar to other grid generation techniques, the goal of the extended 'two-boundary technique' is to determine the co-ordinate transformation (equations (14)–(16)) so that co-ordinate lines of the transformed domain correspond to grid lines and boundaries of the 'physical' spatial domain.

The extended 'two-boundary technique' developed here for generating grid points involves six major steps:

1. Select a time stretching function, i.e. a relationship between t and τ.
2. Select two boundaries of the spatial domain that do not touch each other at any point. These two boundaries will coincide with two co-ordinate lines, $\eta = 0$ and $\eta = 1$, in the transformed domain.
3. Describe the two boundaries selected above by parameter equations using the parameters ξ and τ, co-ordinates of the transformed domain.
4. Define curves that connect the two boundaries.
5. Control distribution of grid points by stretching functions.
6. Calculate the metric coefficients needed to obtain finite-difference solutions to fluid flow problems.

In the following, the above six steps are described in detail by applying them to generate grid points in the combustion chamber geometry described in the preceding section.

Step 1—select a time stretching function

Here the relationship between t and τ is taken to be

$$t = \tau, \tag{17}$$

i.e. there is no stretching in time. For problems involving disparate time scales at different periods in time, time stretching may be useful.

Step 2—select two boundaries

The next step is to select two boundaries of the spatial domain that do not touch each other at any point. For the spatial domain represented by the shaded area in Figure 1, we selected the two boundaries to be surface 1 and surface 2.

The boundaries chosen determine the correspondence between the boundary of the time-varying spatial domain and the boundary of the transformed domain. Here we chose surfaces 2 and 1 to correspond to co-ordinate lines $\eta = 0$ and $\eta = 1$, respectively (see Figure 3), i.e.

$$X_2 = X(\xi, \eta = 0, \tau) = X_2(\xi, \tau), \tag{18}$$

$$Y_2 = Y(\xi, \eta = 0, \tau) = Y_2(\xi, \tau), \tag{19}$$

$$X_1 = X(\xi, \eta = 1, \tau) = X_1(\xi, \tau), \tag{20}$$

$$Y_1 = Y(\xi, \eta = 1, \tau) = Y_1(\xi, \tau). \tag{21}$$

The above equations indicate that surfaces 1 and 2 must be expressed in parametric form in terms of the parameters ξ and τ.

In general, the 'two-boundary technique' can only correctly map two boundaries. However, if the remaining two boundaries are straight lines as in the present problem, then they can also be mapped correctly. As will be shown in Step 4, the remaining two surfaces, surfaces 5 and 6 in Figure 1, were mapped to co-ordinate lines $\xi = 0$ and $\xi = 1$, respectively (see Figure 3).

Here it is noted that many spatial domains of practical interest are characterized by only two boundaries that do not touch each other at any points. An example is the domain between an aerofoil and the inflow–outflow boundary.

When one does encounter a spatial domain more complex than that discussed here, the 'four-surface technique' developed by Vinokur and Lombard[10] can be used. Vinokur and Lombard extended the 'two-boundary technique' of Smith so that all four boundaries can be mapped correctly. The technique presented in this paper can also be used in conjunction with the 'four-surface technique' of Vinokur and Lombard for generating grid points that move in the 'physical' spatial domain.

Step 3—obtain parametric representations

Once the two boundaries have been selected, the next step is to represent these two boundaries by parametric equations in terms of the parameters ξ and τ. It is noted that since surfaces 5 and 6 were mapped to co-ordinate lines $\xi = 0$ and $\xi = 1$, respectively, the parameter ξ can only vary between 0 and 1.

A parametric representation of surface 1 is given by equations (1) and (2) in terms of the parameter A and a parametric representation of surface 2 is given by equations (3)–(12) in terms of the parameters V and t. t in equations (3)–(12) can be replaced by τ because of equation (17). We now must relate A and V to ξ. Since ξ can only vary between 0 and 1, we relate A and V to ξ as follows:

$$\xi = 1 - \frac{A - A_L}{A_T - A_L}, \tag{22}$$

$$\xi = 1 - \frac{V - V_L}{V_T - V_L}, \tag{23}$$

where $A_L = (\pi/6) - (\theta/3)$, $A_T = A_L + 2\pi/3$, $V_L = \pi/6$ and $V_T = \pi/2$. θ in the expression for A_L is given by equation (12) with t replaced by τ. By substituting equation (22) into equations (1) and (2) and by substituting equation (23) into equations (3)–(12), we obtain the desired parametric equations.

Step 4—define connecting curves

A number of interpolation techniques can be used to derive the curves for connecting the two boundaries selected in Step 2. Here we shall investigate only two methods, Lagrange interpolation and Hermite interpolation.

Method 1—Lagrange interpolation. By using Lagrange interpolation to interpolate between the two boundaries selected in Step 2, we can readily obtain the following functional form for the connecting curves:[11]

$$X(\xi, \eta, \tau) = X_2(\xi, \tau) l_1(\eta) + X_1(\xi, \eta) l_2(\eta), \tag{24}$$

$$Y(\xi, \eta, \tau) = Y_2(\xi, \tau) l_1(\eta) + Y_1(\xi, \eta) l_2(\eta). \tag{25}$$

In the above equations, X_1, Y_1, X_2 and Y_2 are the co-ordinates of surfaces 1 and 2. These co-ordinates were determined in Step 3. The functions $l_1(\eta)$ and $l_2(\eta)$ are constrained by

$$l_1(\eta=0)=1, \qquad l_1(\eta=1)=0,$$
$$l_2(\eta=0)=0, \qquad l_2(\eta=1)=1.$$

These constraints give

$$l_1(\eta)=\frac{(\eta-1)}{(0-1)}=1-\eta,$$

$$l_2(\eta)=\frac{(\eta=0)}{(1-0)}=\eta.$$

Substitution of the above equations into equations (24) and (25) yields the desired functional form of the connecting curves based on Lagrange interpolation:

$$X(\xi,\eta,\tau)=(1-\eta)X_2(\xi,\tau)+\eta X_1(\xi,\tau), \tag{26}$$

$$Y(\xi,\eta,\tau)=(1-\eta)Y_2(\xi,\tau)+\eta Y_1(\xi,\tau). \tag{27}$$

The connecting curves derived above using Lagrange interpolation have two disadvantages:

1. In general, the connecting curves will not intersect the two boundaries perpendicularly. This makes it difficult to implement derivative boundary conditions.
2. The two boundaries selected in Step 2 cannot be S-shaped. This is because connecting curves given by equations (26) and (27) are straight lines and hence will intersect an S-shaped boundary twice. This disadvantage severly limits the usefulness of equations (26) and (27).

The disadvantages cited above are not insurmountable. Both disadvantages can be removed by defining a number of intermediate boundaries between the two boundaries selected. Such a procedure would be bery similar to the 'multi-surface technique' of Eiseman.[12-14] A simpler procedure for overcoming the aformentioned disadvantages is to use Hermite interpolation, which is described below.

Method 2—Hermit interpolation. By using Hermite interpolation to interpolate between the two boundaries selected in Step 2, we can readily obtain the following functional form for the connecting curves:[11]

$$X(\xi,\eta,\tau)=X_2(\xi,\tau)h_1(\eta)+X_1(\xi,\tau)h_2(\eta)+\frac{\partial X(\xi,\eta=0,\tau)}{\partial\eta}h_3(\eta)+\frac{\partial X(\xi,\eta=1,\tau)}{\partial\eta}h_4(\eta), \tag{28}$$

$$Y(\xi,\eta,\tau)=Y_2(\xi,\tau)h_1(\eta)+Y_1(\xi,\tau)h_2(\eta)+\frac{\partial Y(\xi,\eta=0,\tau)}{\partial\eta}h_3(\eta)+\frac{\partial Y(\xi,\eta=1,\tau)}{\partial\eta}h_4(\eta). \tag{29}$$

X_1, Y_1, X_2 and Y_2 are the co-ordinates of surfaces 1 and 2 and were determined in Step 3. The functions $h_1(\eta)$, $h_2(\eta)$, $h_3(\eta)$, $h_4(\eta)$, $\partial X(\xi,\eta=0,\tau)/\partial\eta$, $\partial X(\xi,\eta=1,\tau)/\partial\eta$, $\partial Y(\xi,\eta=0,\tau)/\partial\eta$ and $\partial Y(\xi,\eta=1,\tau)/\partial\eta$ still need to be determined.

$h_1(\eta)$, $h_2(\eta)$, $h_3(\eta)$ and $h_4(\eta)$ are determined from the following constrains:

$$h_1(\eta=0)=1, \quad h_1(\eta=1)=0, \quad \frac{\mathrm{d}h_1(\eta=0)}{\mathrm{d}\eta}=\frac{\mathrm{d}h_1(\eta=1)}{\mathrm{d}\eta}=0,$$

$$h_2(\eta=0)=0, \quad h_2(\eta=1)=1, \quad \frac{\mathrm{d}h_2(\eta=0)}{\mathrm{d}\eta}=\frac{\mathrm{d}h_2(\eta=1)}{\mathrm{d}\eta}=0,$$

$$h_3(\eta=0)=h_3(\eta=1)=0, \quad \frac{\mathrm{d}h_3(\eta=0)}{\mathrm{d}\eta}=1, \quad \frac{\mathrm{d}h_3(\eta=1)}{\mathrm{d}\eta}=0,$$

$$h_4(\eta=0)=h_4(\eta=1)=0, \quad \frac{\mathrm{d}h_4(\eta=0)}{\mathrm{d}\eta}=0, \quad \frac{\mathrm{d}h_4(\eta=1)}{\mathrm{d}\eta}=1,$$

giving

$$h_1(\eta)=2\eta^3-3\eta^2+1, \tag{30}$$

$$h_2(\eta)=-2\eta^3+3\eta^2, \tag{31}$$

$$h_3(\eta)=\eta^3-2\eta^2+\eta, \tag{32}$$

$$h_4(\eta)=\eta^3-\eta^2. \tag{33}$$

The remaining functions $\partial X(\xi,\eta=0,\tau)/\partial\eta$, $\partial X(\xi,\eta=1,\tau)/\partial\eta$, $\partial Y(\xi,\eta=0,\tau)/\partial\eta$ and $\partial Y(\xi,\eta=1,\tau/\partial\eta$ are chosen so that the connecting curves (equations (28) and (29)) perpendicularly intersect the two boundaries selected in Step 2. In order for the connecting curves given by equations (28) and (29) to intersect boundary 1 perpendicularly in the 'physical' spatial domain, the dot product of $\mathbf{e}_\xi$ (the vector tangent to boundary 1) and $\mathbf{e}_\eta$ (the vector tangent to the connecting curve) at any point on boundary 1 must be zero, i.e.

$$\mathbf{e}_\xi\cdot\mathbf{e}_\eta=0,$$

or

$$\left(\frac{\partial X_2(\xi,\tau)}{\partial\xi}\mathbf{i}+\frac{\partial Y_2(\xi,\tau)}{\partial\xi}\mathbf{j}\right)\cdot\left(\frac{\partial X(\xi,\eta=0,\tau)}{\partial\eta}\mathbf{i}+\frac{\partial Y(\xi,\eta=0,\tau)}{\partial\eta}\mathbf{j}\right)=0,$$

$$\frac{\partial X_2(\xi,\tau)}{\partial\xi}\frac{\partial X(\xi,\eta=0,\tau)}{\partial\eta}+\frac{\partial Y_2(\xi,\tau)}{\partial\xi}\frac{\partial Y(\xi,\eta=0,\tau)}{\partial\eta}=0. \tag{34}$$

From equation (34), it can be seen that orthogonality is guaranteed if the following two conditions are satisfied:

$$\frac{\partial X(\xi,\eta=0,\tau)}{\partial\eta}=-\frac{\partial Y_2(\xi,\tau)}{\partial\xi} \quad \text{or} \quad \frac{\partial X(\xi,\eta=0,\tau)}{\partial\eta}=-K_2(\xi)\frac{\partial Y_2(\xi,\tau)}{\partial\xi}, \tag{35}$$

$$\frac{\partial Y(\xi,\eta=0,\tau)}{\partial\eta}=+\frac{\partial X_2(\xi,\tau)}{\partial\xi} \quad \text{or} \quad \frac{\partial Y(\xi,\eta=0,\tau)}{\partial\eta}=+K_2(\xi)\frac{\partial X_2(\xi,\tau)}{\partial\xi}. \tag{36}$$

In a similar manner, we can show that for orthogonality at boundary 2, the following conditions must be satisfied:

$$\frac{\partial X(\xi,\eta=1,\tau)}{\partial\eta}=-K_1(\xi)\frac{\partial Y_1(\xi,\tau)}{\partial\xi}, \tag{37}$$

$$\frac{\partial Y(\xi,\eta=1,\tau)}{\partial\eta}=+K_1(\xi)\frac{\partial X_1(\xi,\tau)}{\partial\xi}. \tag{38}$$

$K_1(\xi)$ and $K_2(\xi)$ in equations (35)–(38) are chosen by trial and error to ensure that grid lines do not overlap each other at the interior of the 'physical' spatial domain. For the present problem

$$K_1(\xi)=K_2(\xi)=2\{[X_1(\xi,\tau)-X_2(\xi,\tau)]^2+[Y_1(\xi,\tau)-Y_2(\xi,\tau)]^2\}^{1/2}. \tag{39}$$

Since $\partial X_1(\xi,\tau)/\partial\xi$, $\partial X_2(\xi,\tau)/\partial\xi$, $\partial Y_1(\xi,\tau)/\partial\xi$ and $\partial Y_2(\xi,\tau)/\partial\xi$ can be evaluated

analytically from the parametric equations derived in Step 3, $\partial X(\xi, \eta = 0, \tau)/\partial \eta$, $\partial X(\xi, \eta = 1, \tau)/\partial \eta$, $\partial Y(\xi, \eta = 0, \tau)/\partial \eta$ and $\partial Y(\xi, \eta = 1, \tau)/\partial \eta$ are calculated by using equations (35)–(39).

Substitution of equations (30)–(33) and equations (35)–(39) into equations (28) and (29) gives the desired connecting curves based on Hermite interpolation.

In this paper, grid points inside the cumbustion engine geometry described in the preceding section were generated by using equations (28) and (29) except for surfaces 5 and 6. Surfaces 5 and 6 are straight lines in the 'physical' spatial domain and were mapped by using equations (26) and (27).

Step 5—control distribution of grid points

By following Steps 1 to 4 described above, we can map a system of moving and non-uniformly distributed grid points in the 'physical' spatial domain onto a transformed domain where all of the grid points are stationary and uniformly distributed. The question now is: are we satisfied with the distribution of grid points in the 'physical' spatial domain? For example, are there enough grid points in regions containing sharp gradients? If the answer is no, then stretching functions can be used to redistribute the system of grid points. Several useful stretching functions are given in References 15 and 16.

For the combustion chamber geometry described in the preceding section, a large number of grid points are needed near the walls of the combustion chamber to resolve the boundary layer flow there. To concentrate more grid points near surfaces 5 and 6 (see Figure 3), we replaced equations (22) and (23) by the following equations:

$$1 - \frac{A - A_L}{A_T - A_L} = 0{\cdot}5(B_\xi + 1) - B_\xi\{1 + [(B_\xi + 1)/(B_\xi - 1)]^{(2\xi - 1)}\}^{-1}, \tag{40}$$

$$1 - \frac{V - V_L}{V_T - V_L} = 0{\cdot}5(B_\xi + 1) - B_\xi\{1 + [(B_\xi + 1)/(B_\xi - 1)]^{(2\xi - 1)}\}^{-1}. \tag{41}$$

To concentrate more grid points near surfaces 1 and 2 (see Figure 3), we replaced η in equations (26)–(29) by the following expression:

$$0{\cdot}5(B_\eta + 1) - B_\eta\{1 + [(B_\eta + 1)/(B_\eta - 1)]^{(2\eta - 1)}\}^{-1}. \tag{42}$$

Equation (42) and the right-hand sides of equations (40) and (41) are stretching functions. By varying the parameters B_ξ and B_η between 0 and 1, different concentrations of grid points can be obtained. Here B_ξ and B_η were set equal to 0·1 and 0·3, respectively.

Step 6—calculate the metric coefficients

Once we have obtained a satisfactory distribution of grid points in the 'physical' spatial domain, we can proceed with the determination of the metric coefficients. The metric coefficients ($\xi_x = \partial\xi/\partial X$, $\xi_y = \partial\xi/\partial Y$, $\xi_t = \partial\xi/\partial t$, $\eta_x = \partial\eta/\partial X$, $\eta_y = \partial\eta/\partial Y$ and $\eta_t = \partial\eta/\partial t$) needed to obtain solutions to partial differential equations (PDEs) on the grid system generated here can be calculated from the following equations (pp. 252–255 of Reference 16, and Reference 17):

$$\xi_x = JY_\eta, \tag{43}$$

$$\xi_y = -JX_\eta, \tag{44}$$

$$\xi_t = J(Y_\tau X_\eta - X_\tau Y_\eta), \tag{45}$$

$$\eta_x = -JY_\xi, \tag{46}$$

$$\eta_y = JX_\xi, \tag{47}$$

$$\eta_t = J(X_\tau Y_\xi - Y_\tau X_\xi), \tag{48}$$

where J is the Jacobian and is given by

$$J = 1/(X_\xi Y_\eta - X_\eta Y_\xi). \tag{49}$$

Even though $X_\xi = \partial X/\partial\xi$, $X_\eta = \partial X/\partial\eta$, $X_\tau = \partial X/\partial\tau$, $Y_\xi = \partial Y/\partial\xi$, $Y_\eta = \partial Y/\partial\eta$ and $Y_\tau = \partial Y/\partial\tau$ in equations (43)–(49) can be evaluated analytically, these terms should be evaluated by finite-difference methods.[3,6] Depending on how the PDEs are cast (e.g. in strong conservation law form or in chain rule conservation law form), X_ξ, X_η, X_τ, Y_ξ, Y_η and Y_τ may be required to be finite-differenced in certain prescribed manners to avoid geometric induced errors.[17,18]

If the PDEs to be solved are cast in chain rule conservation law form.[18] X_ξ, X_η, Y_ξ and Y_η can be calculated by using second-order accurate in space central-difference formulae. X_τ and Y_τ can be calculated from the following generalized time-difference formula:[19]

$$U^{n+1} - \frac{\gamma_1 \Delta\tau}{1+\gamma_2}\left(\frac{\partial U}{\partial z}\right)^{n+1} = (1-\gamma_1)\frac{\Delta\tau}{1+\gamma_2}\left(\frac{\partial U}{\partial\tau}\right)^{n} + U^n + \frac{\gamma_2}{1+\gamma_2}(U^n - U^{n-1}). \tag{50}$$

In the above equation, $\Delta\tau$ denotes the time step size; γ_1 and γ_2 are constants; and U denotes either X and Y. Equation (50) can reproduce most commonly used time-difference formulae by varying γ_1 and γ_2. Here γ_1 and γ_2 were set equal to 0·5 and 0, respectively.

RESULTS

In order to demonstrate the usefulness and the feasibility of the technique presented in the previous section for generating grid points in deforming spatial domains,

1. Grid points were generated inside one of the combustion chambers of a motored two-dimensional rotary engine.
2. Numerical solutions were obtained for the flow field inside one of the combustion chambers of a motored two-dimensional rotary engine with the intake and exhaust ports closed.

Figures 4–9 show grid points generated inside one combustion chamber at several different crank angles. The grid systems shown in Figures 4–6 were generated by following Steps 1 to 4 described in the preceding section, i.e. no stretching functions were used. The grid systems shown in Figures 7–9 were generated by following Steps 1 to 5, i.e. stretching functions were used. Comparing Figures 4–9 indicates that the stretching functions employed can concentrate grid points near the walls of the combustion chamber. Figures 4–9 show that the technique presented here can be used to generate grid points in time-varying or deforming spatial domains.

To further demonstrate the usefulness as well as the feasibility of the grid generation technique presented here, numerical solutions were obtained for the flow field inside the combustion chambers of a motored two-dimensional rotary engine with the intake and exhaust ports closed. The description of this rotary engine problem as well as the details of the governing equations and the numerical method of solution are given elsewhere.[20,21] Here only a brief description is given to facilitate interpretation of the results.

The geometry of the problem studied is shown in Figure 1. Initially, the combustion chamber is assumed to be filled with stagnant air at constant temperature and pressure. Suddenly, the shaft starts to rotate at an angular speed of 5000 revolutions per minute. The rotation of the

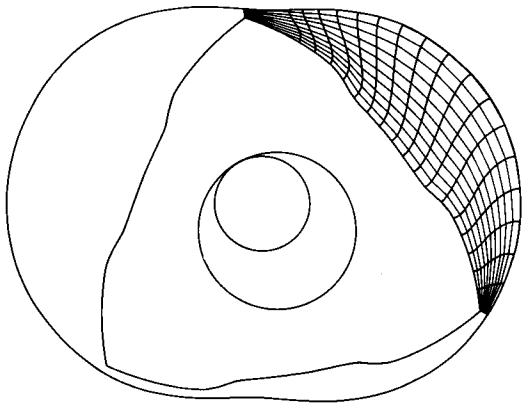

Figure 4. Grid points generated by following Steps 1 to 4. No stretching functions used. The crank angle is 150°

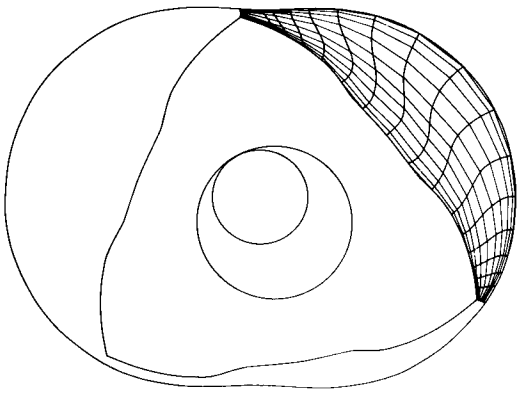

Figure 7. Grid points generated by following Steps 1 to 5. Stretching functions used. The crank angle is 150°

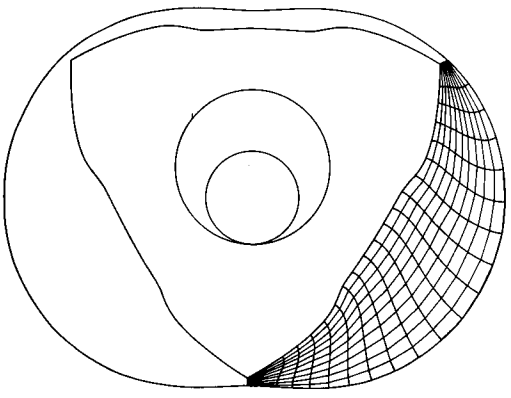

Figure 5. Grid points generated by following Steps 1 to 4. No stretching functions used. The crank angle is 360°

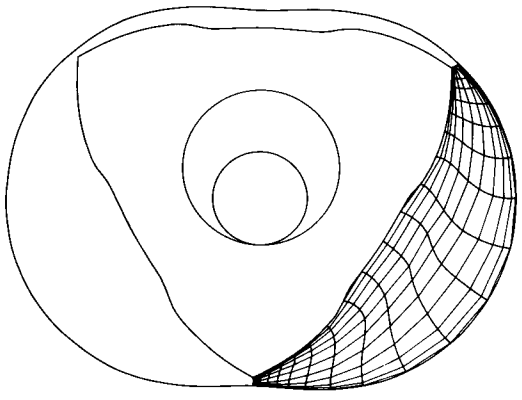

Figure 8. Grid points generated by following Steps 1 to 5. Stretching functions used. The crank angle is 360°

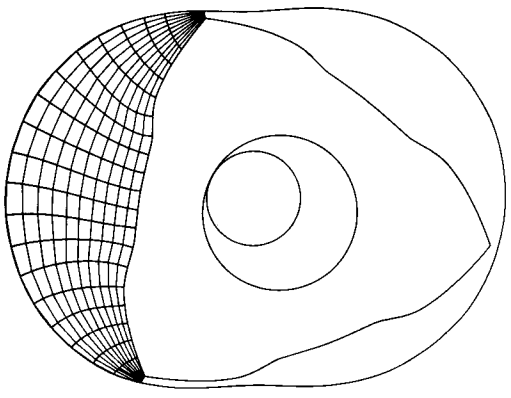

Figure 6. Grid points generated by following Steps 1 to 4. No stretching functions used. The crank angle is 840°

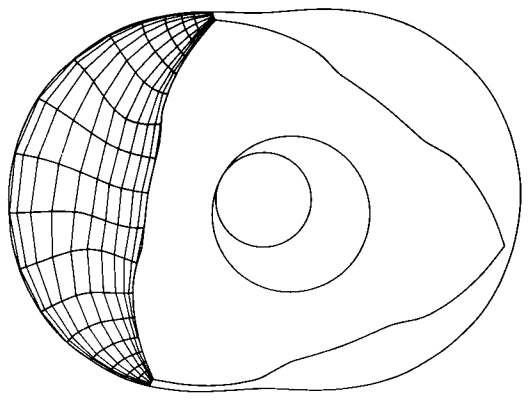

Figure 9. Grid points generated by following Steps 1 to 5. Stretching functions used. The crank angle is 840°

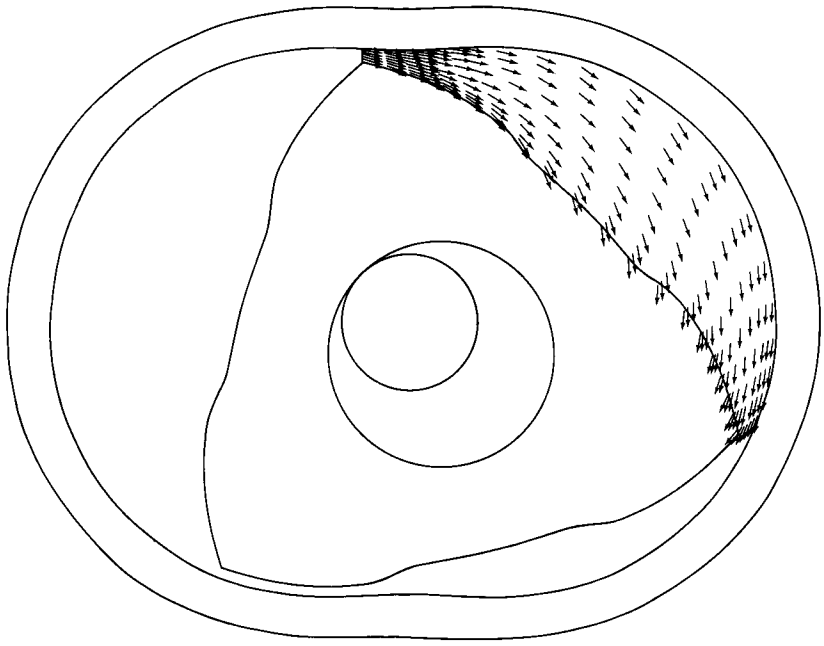

Figure 10. Flow pattern inside the combustion chamber at crank angle 136°

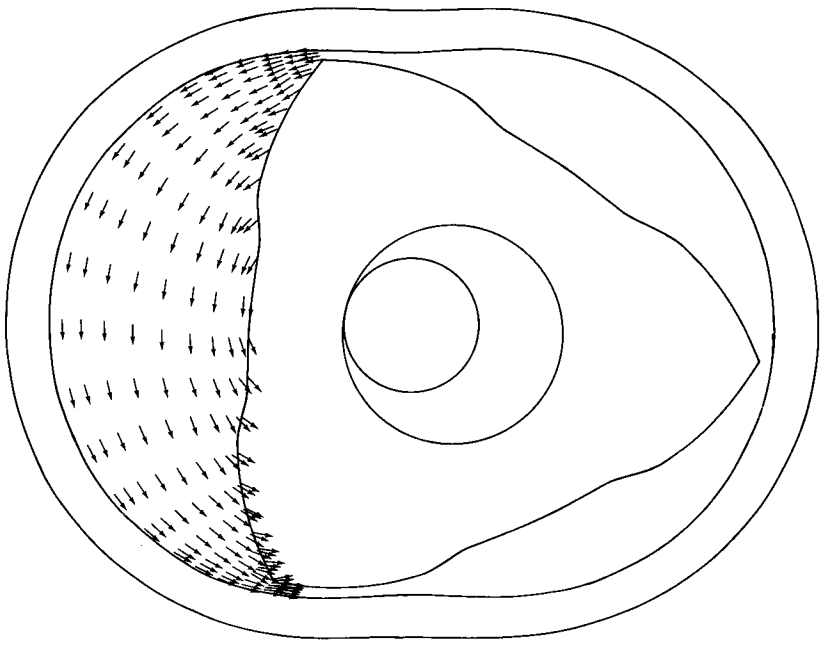

Figure 11. Flow pattern inside the combustion chamber at crank angle 254°

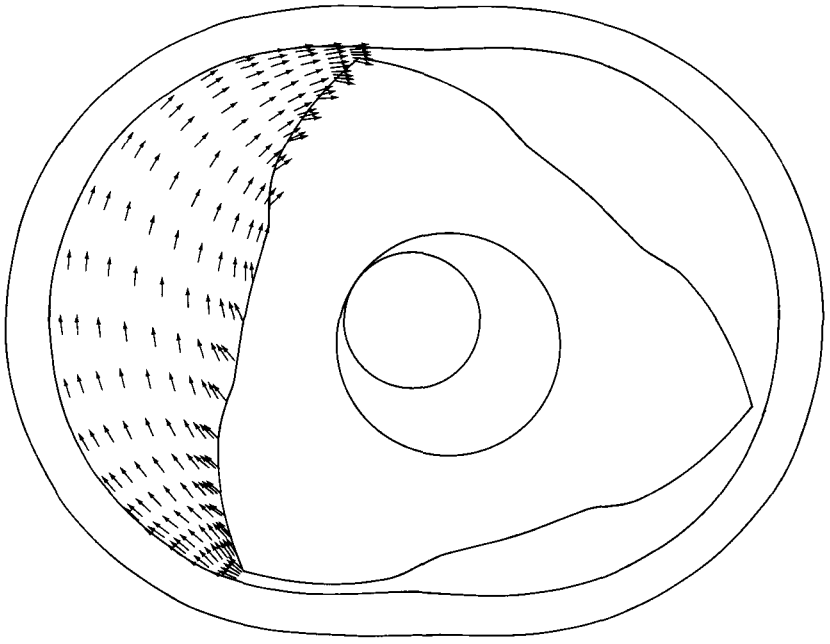

Figure 12. Flow pattern inside the combustion chamber at crank angle 844°

shaft causes the rotor to rotate which in turns causes the air inside the combustion chamber to move and to be compressed or expanded.

The deforming spatial domain of this problem was replaced with grid points generated by following Steps 1 to 6 described in the preceding section with 51 grid points spanning the ξ direction and 21 grid points spanning the η direction.

The governing equations used to obtain solutions were the conservation equations of mass, momentum and energy valid for unsteady, two-dimensional, compressible, laminar flows of a viscous and thermally conducting ideal gas. The increased mixing due to turbulence was modelled by appropriately chosen effective transport properties. The thermal conductivity was selected by taking the turbulent Prandtl number to be equal to unity. These governing equations were first written in Cartesian co-ordinates (X, Y and t) and then transformed to the ξ–η–τ co-ordinate system in chain rule conservation law form.[18]

Numerical solutions to the governing equations described above were obtained by the implicit factored method of Beam and Warming.[19,20] Boundary conditions needed to obtain solutions were implemented implicitly.[22]

Numerical solutions were generated for three revolutions of the shaft (or one complete revolution of the rotor). In Figures 10–12, results for the velocity are presented in graphical form illustrating the flow pattern at several different crank angles. As described in References 20 and 21, the results obtained are reasonable.

CONCLUSION

In this paper we have presented an efficient and versatile algebraic grid generation technique for generating grid points inside time-varying or deforming two-dimensional spatial domains. The procedure for implementing this grid generation technique was described in detail by

generating grid points inside one of the combustion chambers of a motored two-dimensional rotary engine. The usefulness and the feasibility of the grid generation technique presented were demonstrated by the successful generation of grid points inside that combustion chamber and by obtaining reasonable numerical solutions to the flows that take place inside that combustion chamber.

REFERENCES

1. R. E. Smith (ed.), *Numerical Grid Generation*, NASA Conference Publication 2166, 1980.
2. J. F. Thompson (ed.), *Numerical Grid Generation*, Elsevier Publishing Co., Inc., New York, 1982.
3. J. F. Thompson, Z. U. A. Warsi and C. W. Mastin, 'Boundary-fitted coordinate systems for numerical solution of partial differential equations—a review', *Journal of Computational Physics*, **47**, 1–108 (1982).
4. I. Babuska, J. Chandra and J. E. Flaherty, *Adaptive Computational Methods for Partial Differential Equations*, SIAM, Philadelphia, 1983.
5. K. N. Ghia and U. Ghia (eds), *Advances in Grid Generation*, FED—Vol. 5, ASME, New York, 1983.
6. J. F. Thompson, 'Grid generation techniques in computational fluid dynamics', *AIAA Journal*, **22**, 1505–1523 (1984).
7. R. E. Smith, 'Two-boundary grid generation for the solution of the three-dimensional compressible Navier–Stokes equations', *NASA Technical Memorandum 83123*, 1981.
8. R. F. Ansdale, *The Wankel RC Engine—Design and Performance*, Iliffe Books Limited, London, 1968, pp. 131–139.
9. K. Yamamoto, *Rotary Engine*, Sankaido Co., Ltd., Tokyo, Japan, 1981, pp. 11–13.
10. M. Vinokur and C. K. Lombard, 'Algebraic grid generation with corner singularities', in Reference 5, pp. 99–106.
11. A. Ralston and P. Rabinowtiz, *A First Course in Numerical Analysis*, Second Edition, McGraw-Hill Book Co., New York, 1978.
12. P. R. Eiseman, 'A multi-surface method of coordinate generation', *Journal of Computational Physics*, **33**, 118–150 (1979).
13. P. R. Eiseman, 'Coordinate generation with precise controls over mesh properties', *Journal of Computational Physics*, **47**, 331–351 (1982).
14. P. R. Eiseman, 'High level continuity for coordinate generation with precise controls', *Journal of Computational Physics*, **47**, 352–374 (1982).
15. M. Vinokur, 'On one-dimensional stretching functions for finite-difference calculations', *Journal of Computational Physics*, **50**, 215–234 (1983).
16. D. A. Anderson, J. C. Tannehil and R. H. Pletcher, *Computational Fluid Mechanics and Heat Transfer*, Hemisphere Publishing Corp., Washington, 1984, pp. 247–252.
17. P. D. Thomas and C. K. Lombård, 'Geometric conservation law and its application to flow computations on moving grids', *AIAA Journal*, **17**, 1030–1037 (1979).
18. R. G. Hindman, 'Generalized coordinate forms of governing fluid equations and associated geometrically induced errors', *AIAA Journal*, **20**, 1359–1367 (1982).
19. R. M. Beam and R. F. Warming, 'An implicit factored scheme for the compressible Navier–Stokes equations', *AIAA Journal*, **16**, 393–402 (1978).
20. S.-L. Yang, T. I-P. Shih and H. J. Schock, 'Application of the implicit factored method to arbitrary two-dimensional unsteady flow problems', to be submitted.
21. T. I-P. Shih, S. L. Yang and H. J. Schock, 'A two-dimensional numerical study of the flow inside the combustion chambers of a motored rotary engine', to be submitted.
22. T. I-P. Shih, G. E. Smith, G. S. Springer and Y. Rimon, 'Boundary conditions for the solution of the compressible Navier–Stokes equations by an implicit factored method', *Journal of Computational Physics*, **52**, 54–79 (1983).