Arch. Mech., 32, 5, pp. 655–661, Warszawa 1980

An accurate algorithm for Dirichlet boundary conditions in hyperbolic flows

K. FÖRSTER (STUTTGART)

THE USUAL explicit finite difference approach for the computation of boundary values is critically reviewed; an improved philosophy is suggested and tested using Ringleb's solution as a test rig.

Przeanalizowano krytycznie standardowe podejście do obliczania wartości brzegowych za pomocą różnic skończonych sugerując pewne ulepszenia, które sprawdzono traktując rozwiązanie Ringleba jako sprawdzian metody.

Критически проанализирован стандартный подход к расчету граничных значений при помощи конечных разностей, предполагая некоторые улучшения, которые проверены трактуя решение Ринглеба как проверку метода.

1. Introduction

THE ACCURACY of explicit finite difference solving procedures for the hyperbolic flow equations is usually bound by the quality of the boundary algorithm. Whereas the accuracy potential of a common second-order field-point algorithm is, by and large, about 10^{-4} for a reasonably well-behaved problem (and thus surprisingly close to that of a similar scheme for ordinary differential equations) most commonly used boundary algorithms depress this figure by a factor well over 10. Thus it was felt necessary to review critically the usual procedure and eventually look for a better one.

2. The usual computation procedure

To set forth the philosophy we use the problem of two-dimensional, steady, supersonic potential flow between rigid walls, and we assume the very important basic step of introducing contour-aligned coordinates ξ, η to be already performed. Then for the Cartesian velocity components u, v we have the differential equations

$$u_\xi = b_{11}u_\eta + b_{12}v_\eta,$$
$$v_\xi = b_{21}u_\eta + b_{22}v_\eta,$$

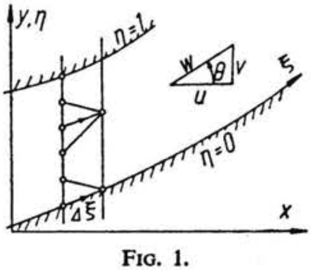

FIG. 1.

where

$$b_{11} = -\eta_x - \frac{2uv}{u^2-c^2}\eta_y, \quad b_{12} = -\frac{v^2-c^2}{u^2-c^2}\eta_y,$$

$$b_{21} = \eta_x, \quad b_{22} = -\eta_y;$$

c speed of sound.

We call u, v von Neumann variables because at the boundaries there exists for them a von Neumann type boundary condition

$$v_b = t_b u_b$$

($t = \tan\theta$, θ direction of streamline, index b means boundary). The above form of the differential equations lends itself directly to an explicit solving procedure which we sketch here only in its simplest first-order form: inside the domain, truncated Taylor series are used to advance the solution from one line of grid points ξ = const (denoted by an upper index n) to a new line $\xi+\Delta\xi$ = const (upper index $n+1$):

$$u_i^{n+1} = u_i^n + (u_\xi)_i^n \Delta\xi,$$
$$v_i^{n+1} = v_i^n + (v_\xi)_i^n \Delta\xi,$$

the ξ-derivatives are calculated from some discretized form of the pde's right hand sides.

At the boundaries, however, there is the boundary condition as a third equation, and the whole set would be overdetermined. The simplest and widely used way out is to drop one differential equation, say that for v, calculate u as inside the field (with obvious modifications in discretizing the η-derivatives) and use the boundary condition for the computation of v:

$$u_b^{n+1} = u_b^n + (u_\xi)_b^n \Delta\xi,$$
$$v_b^{n+1} = t_b^{n+1} u_b^{n+1}.$$

This procedure is quite unsatisfactory. First, there is the arbitrariness of which equation to drop. Second, and worse: the shape of the boundary downstream of ξ, especially its curvature t_ξ, does not enter the calculation but post festum; it is ignored completely in the calculation of u_b^{n+1}. This is not only true for the sketched one-step scheme but also for the particular steps in multistep schemes of higher order of accuracy.

This situation is not altered basically if one introduces other dependent variables. A widely used set is P, t ($P = \ln p$, p pressure)whose differential equations read

$$P_\xi = a_{11}P_\eta + a_{12}t_\eta \quad \text{with} \quad a_{11} = a_{22} = -\eta_x - \frac{u^2 t}{u^2-c^2}\eta_y,$$

$$t_\xi = a_{21}P_\eta + a_{22}t_\eta, \qquad a_{12} = -\frac{\varkappa u^2}{u^2-c^2}\eta_y, \quad a_{21} = -\frac{w^2-c^2}{u^2-c^2}\frac{c^2}{\varkappa u^2}\eta_y,$$

($w^2 = u^2+v^2$, $\varkappa$ ratio of specific heats), and we call P, t Dirichlet variables because the boundary conditions are of the Dirichlet type: $t = t(\xi)$ respectively $P = P(\xi)$ are prescribed functions along rigid walls, respectively free-jet boundaries.

The explicit solving procedure is quite similar to the one above: inside the domain, P and t are advanced as described for u and v and at the boundaries only the relevant differential equation is retained, so in our case we have

$$P_b^{n+1} = P_b^n + (P_\xi)_b^n \Delta\xi,$$
$$t_b^{n+1} = t(\xi + \Delta\xi).$$

Though here the arbitrariness is removed, there still remains the fact that P_b^{n+1} is computed ignoring completely t_ξ.

Of course in this case the theoretically sound method of characteristics or one of its many variants could be used (which in fact is frequently done) but experience shows that mixing two such different methods very often results in unexpectedly low accuracy and, furthermore, the computer program is considerably lengthened by the extra code needed.

3. A new approach

We will once more consider the differential equations

$$P_\xi = a_{11} P_\eta + a_{12} t_\eta,$$
$$t_\xi = a_{21} P_\eta + a_{11} t_\eta.$$

At a rigid wall only $(P_\xi)_b$ is needed whereas $(t_\xi)_b$ is known, so why not use the second equation to eliminate one of the η-derivatives from the first one? By linear combination we easily get

$$\text{either} \quad P_\xi = \frac{a_{12}}{a_{11}} t_\xi + \left(a_{11} - \frac{a_{12}a_{21}}{a_{11}}\right) P_\eta$$

but because of $(a_{11})_b \sim t_b$ this form is singular for $t_b = 0$ (a very likely value to occur)

$$\text{or} \quad P_\xi = \frac{a_{11}}{a_{21}} t_\xi + \left(a_{12} - \frac{a_{11}^2}{a_{21}}\right) t_\eta .$$

The latter equation is very well suited for advancing P along the boundary (so we may call it the differential boundary equation) and it contains t_ξ!

Before proceeding we should remark that one could treat the case of the von Neumann variables in a very similar manner: differentiating the boundary conditions for a rigid wall gives $v_\xi = tu_\xi + ut_\xi$ which can be used to yield the differential equations

$$u_\xi = b_{11} u_\eta + b_{12} v_\eta,$$
$$tu_\xi = b_{21} u_\eta + b_{22} v_\eta - ut_\xi .$$

Linear combination again gives one equation which is singular for $t_b = 0$ and one useful boundary equation. This procedure bears a faint relationship to the implicit BVLR-scheme [1]. As it is felt, however, that Dirichlet variables are more convenient to use, we return to these.

The boundary equation is closely related to the domain equations and so we can discretize it in quite the same manner as has been chosen for the latter. For example, in the case of the Lax-Wendroff-Richtmyer scheme we extrapolate linearly $P_b^{n+1/2}$ from inside the field, issue a CALL for a_{11}, a_{12}, a_{21} and calculate P_ξ after the boundary equation approximating t_η by a one-sided quadratic with $t_b^{n+1/2}$ and the already (for the first field point) computed values $t_{1/2}^{n+1/2}$ and $t_{3/2}^{n+1/2}$. Thus the coding expense is very low and the computation time for a boundary point becomes distinctly smaller than for a field point.

4. A numerical test

As a test rig we have chosen the well-known Ringleb flow [2] which is an exact solution of Tchapligin's hodograph equation and can be computed very conveniently with the help of some previously published subprograms [3]. From the first Riemann surface of this solution we picked the two channels indicated by heavier lines in Fig. 2 between the

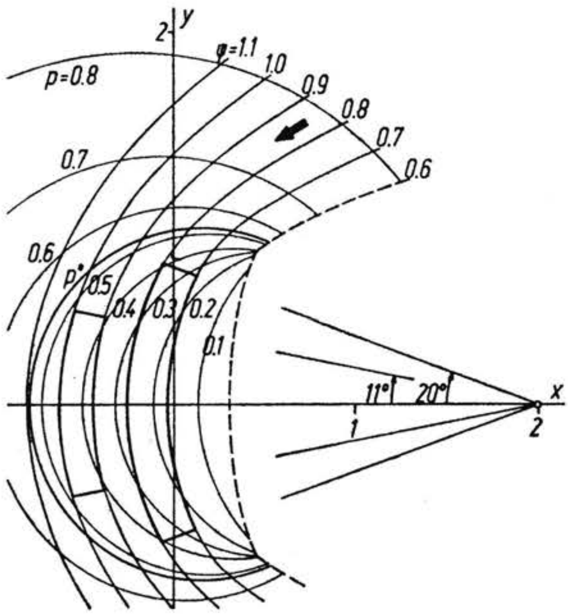

FIG. 2. Streamlines and isobars of Ringleb Flow (part of first Riemann Surface). Chosen examples of channel flow shown by heavy lines.

streamlines $0.7 \leqq \psi \leqq 0.8$ (high speed channel) and $0.9 \leqq \psi \leqq 1.0$ (lower speed channel). They are located symmetrically to the symmetry line of the flow pattern and therefore feature first an expansion and then a recompression which is physically and computationally more sensitive than the former. The computations presented below were based on the polar frame indicated in Fig. 2; the Cartesian frame yielded practically identical results. The following diagrams show lines of constant error-in-pressure $\cdot 10^4$, the grid had 10 meshes across the width of the channel.

The first case, Fig. 3a, is a null-test using the conventional Lax-Wendroff-Richtmyer scheme for the field points and the exact values in boundary points. The result confirms what has been said in the introduction about the accuracy level of a second-order scheme.

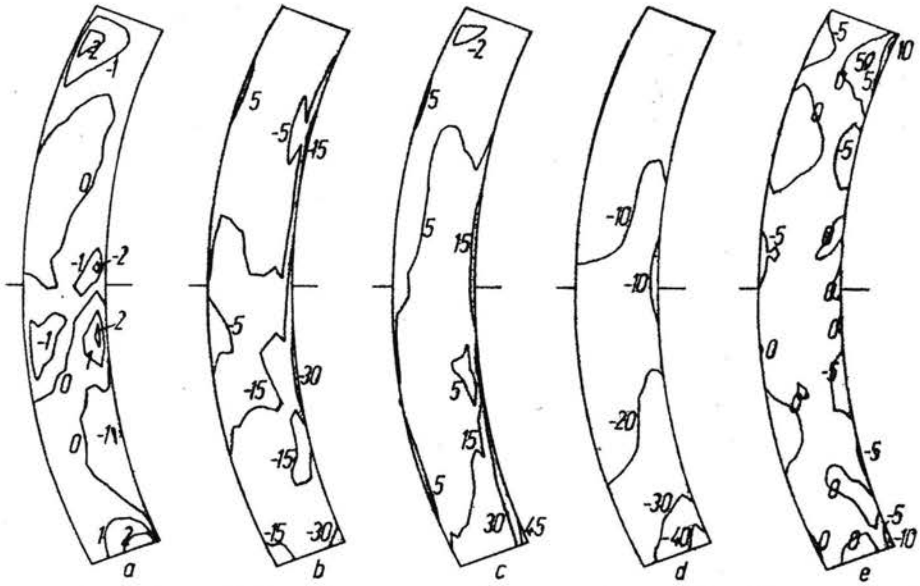

FIG. 3. High speed channel; lines of constant error-in-pressure times 10^4 for different computational methods: a) LWR +exact boundary values, b) LWR+extrapolation of P_ξ, c) same as b)+correction after de Neef, d) Pandolfi and Zannetti, e) LWR+boundary equation algorithm.

The second experiment, Fig. 3b, was done using the crude method outlined above: we just extrapolated $(P_\xi)^{n+1/2}$ from inside the domain to the wall. The result clearly indicates the large increase of the errors mainly in the recompression zone.

In the next run, Fig. 3c, we tried to improve the calculation by adding a correction after de NEEF's paper [4] which in a theoretically very clever way uses the characteristic

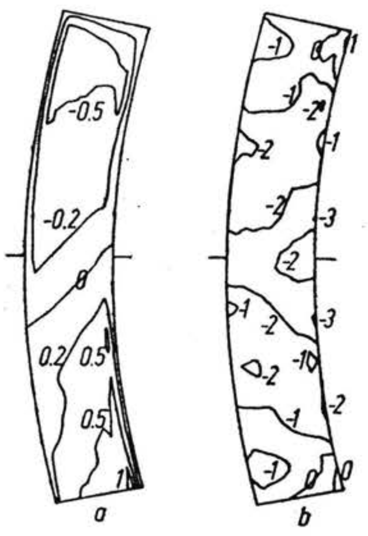

FIG. 4. Lower speed channel; lines of constant error-in-pressure times 10^4 for a) LWR+exact boundary values, b) LWR+boundary equation algorithm.

compatibility equation to compute and correct the error in P. The results are somewhat disappointing and thus provide a good example for what has been said about mixing different computing philosophies: the error in the expansion flow is a bit overcompensated for and increases sharply towards the end of the recompression.

The fourth example, Fig. 3d, has been done by PANDOLFI and ZANNETTI [5] using MacCormack's scheme in the field and a boundary algorithm based on the compatibility equation but expressing the characteristic derivatives in derivatives after ξ and η. The accumulation of errors in the recompression zone is quite clearly marked in this case.

In contrast to all these examples the computation using the boundary equation, Fig. 3e, exhibits not only a markedly lower error level but also no accumulation of errors at all which can be seen from a certain "touch" of symmetry between expansion and recompression. Both features are still more marked in the case of the lower speed channel, Fig. 4b. Figure 4a is the null-test for this case.

5. Further remarks

During the actual computations the new algorithm was found to be remarkably stable. By accident, one boundary value t_b was grossly miscalculated and thus a large error impulse introduced. The boundary algorithm swallowed this impulse wiggleless within three steps $\Delta\xi$, though of course the error was propagated through the field and reflected at the opposite wall due to the hyperbolic nature of the problem.

Further, we found it necessary to code sensitive parts of the program in double precision since a common minicomputer's accuracy (nominally 24 bits of mantissa, depressed by hexadecimal normalization to 21 bits) proved to be too low. Even on a UNIVAC 1108 the results could be improved sligthly by partly double precision coding.

6. Conclusion

An algorithm for the computation of boundary values in two-dimensional hyperbolic flow has been developed using as a basis the differential boundary equation. Thus the new procedure is theoretically satisfying, its discretization easily made consistent with that chosen for the field point algorithm, simple to code, reasonably fast, and computationally stable. It yields results superior to other widely used methods especially in recompression regions. The principle can be extended to free jet boundaries and, so we hope, to multi-dimensional problems.

References

1. K. I. BABENKO, G. P. VOSKRESENSKIY, A. N. LYUBIMOV and V. V. RUSANOV, *Three-dimensional flow of ideal gas past smooth bodies*, NASA TT F-380, April 1966, (Translat. of Russian original paper from 1964).
2. F. RINGLEB, *Exakte Lösungen der Differentialgleichungen einer adiabatischen Gasströmung*, ZAMM, **20**, 185–198, 1940.

3. K. FÖRSTER (ed.), *Boundary algorithms for multidimensional inviscid hyperbolic flows*, Notes on Numerical Fluid Mechanics, **1**, 4–6, Vieweg, Wiesbaden 1978.
4. T. DE NEEF, *Treatment of boundaries in unsteady inviscid flow computations*, Delft Univ. of Technol. Dept. Aerospace Eng. Report LR-262, Febr. 1978.
5. M. PANDOLFI and L. ZANNETTI, *Some tests on finite difference algorithms for computing boundaries in hyperbolic flows*, Contained in Ref. [3], 68–88.

INSTITUT FÜR AERODYNAMIK UND GASDYNAMIK
DER UNIVERSITÄT STUTTGART, BRD.

Received October 25, 1979.