JOURNAL OF COMPUTATIONAL PHYSICS 101, 368–374 (1992)

A Fast Conformal Mapping Algorithm with No FFT

P. LUCHINI AND F. MANZO

Istituto di Gasdinamica, Facoltà di Ingegneria, University of Naples, 80125 Naples, Italy

Received October 16, 1990; revised July 8, 1991

An algorithm is presented for the computation of a conformal mapping discretized on a non-uniformly spaced point set, useful for the numerical solution of many problems of fluid dynamics. Most existing iterative techniques, both those having a linear and those having a quadratic type of convergence, rely on the fast Fourier transform (FFT) algorithm for calculating a convolution integral which represents the most time-consuming phase of the computation. The FFT, however, definitely cannot be applied to a non-uniform spacing. The algorithm presented in this paper has been made possible by the construction of a calculation method for convolution integrals which, despite not using an FFT, maintains a computation time of the same order as that of the FFT. The new technique is successfully applied to the problem of conformally mapping a closely spaced cascade of airfoils onto a circle, which requires an exceedingly large number of points if it is solved with uniform spacing.

1. INTRODUCTION

The numerical calculation of a conformal mapping is the basic step in the solution of a large number of problems of fluid dynamics, including the determination of inviscid flow fields around arbitrary profiles and the setup of orthogonal calculation grids within curved boundaries. One of the most frequently used techniques for numerical conformal mapping of airfoil profiles first maps the given, simply or multiply connected, profile onto a smooth, cornerless, shape by means of analytical pretransformations and then iteratively calculates, in discretized form, the final transformation of the intermediate shape into a circle or annulus.

Several methods have been developed for the numerical calculation of the final mapping of a smooth curve onto a circle, starting with the iterative algorithm devised by Theodorsen and Garrick in 1933 in order to study potential flow around arbitrary wing sections [1]. The Theodorsen–Garrick algorithm was extended by Garrick himself to the cases of biplane wings [2] and cascades of airfoils [3]. Since the Theodorsen–Garrick method involves repeated Fourier transforms, it benefited greatly from the advent of electronic computers and the introduction of the fast Fourier transform (FFT) algorithm, as described in the review paper by Ives [4]. During more recent years, several variants of the Theodorsen–Garrick algorithm were introduced by Melentiev and Kulisch, Timman, Friberg, Bergstroem, Menikoff and Zemach, Chakvararthy and Anderson, and Challis and Burley, all of which are described in the review paper by Gutknecht [5], as well as in other papers collected in Ref. [6] and in the book by Henrici [7].

Among the techniques for numerically mapping an intermediate smooth curve onto a circle, a breakthrough was represented by the idea, hinted at in special situations by Vertgeim [8] and Huebner [9] and independently set forth in general by Wegmann [10] and Luchini and Manzo [11], that a quadratic-convergence Newton-style iterative algorithm for conformal mapping can be constructed using for each iteration the solution of a Riemann–Hilbert problem which requires only twice as much computing time as an iteration of the previous, linearly converging, algorithms. This method requires a much smaller number of iterations to converge than Theodorsen and Garrick's and can handle more general shapes of the initial profile. It has also been extended to cascades and to twin airfoils, by Wegmann [12] and in a faster form by Luchini and Manzo [11].

The most time-consuming step performed, at one or more intermediate stages, in calculating the conformal mapping of a smooth shape onto a circle numerically is the solution of a problem of function conjugation on the circumference, that is, the determination of the imaginary part of a complex analytic function whose real part is given. In Theodorsen and Garrick's method and most of its followers, the solution of this problem is first expressed, according to Schwartz' formula, by a convolution integral between the given real part and a suitable Green function, and then this convolution integral is calculated by Fourier transforming the given function, multiplying its transform by the Fourier transform of the Green function, and reverse transforming the result. With the aid of the FFT algorithm, the convolution integral can thus be computed in a time $O(N \log N)$, N being the number of points of the discretization. The discretization, however, has to be uniform.

Every time the profile whose conformal mapping is

0021-9991/92 $5.00

sought presents multiple scales, for instance, having small bumps or a rather sharp leading edge, a non-uniform point spacing can afford a precise representation of the profile, and of its conformal mapping, with a much lower number of points than required by a uniform discretization. A similar situation arises, even if the initial profile is smooth and regular, when considering closely spaced periodic cascades; for the pretransformation of a cascade into a single profile maps most of the profile's contour onto two very small regions near the leading and trailing edges (the phenomenon of "crowding"), and these two regions are smaller the closer that the spacing is between the successive blades. As will be seen in the examples, calculation of the conformal mapping of a closely spaced cascade using a uniform discretization is very uneconomical.

However, a non-uniform discretization definitely prevents use of the FFT. On the other hand, a straightforward calculation of the convolution integral as N successive numerical quadratures requires a time $O(N^2)$, much longer than $O(N \log N)$. If it is agreed that, even in the present day of ever faster computers, any algorithm is worth being accelerated no matter how small its running time may already be in absolute terms, because there will always be applications in which the algorithm has to be run a large number of times, then the need exists for a method that may afford the calculation of a convolution integral on a non-uniform discretization in a time comparable to that required for uniform spacing. We shall present such an algorithm in the following section.

2. CALCULATION OF A CONVOLUTION INTEGRAL OVER A NON-UNIFORMLY SPACED POINT SET

Let us consider a general convolution integral of the form

$$I(\vartheta') = \oint f(\vartheta)\, g(\vartheta' - \vartheta)\, d\vartheta, \tag{1}$$

where $f(\vartheta)$, which will be called the source function, is known in discretized form for several values of the angle variable ϑ, $g(\vartheta)$ is an analytically known kernel, which for the conjugation problem on the circle is given by $g(\vartheta) = (2\pi)^{-1} \operatorname{cotg}(\vartheta/2)$, and $I(\vartheta')$ is the resulting function. The integral is extended over the whole circumference $0 \leqslant \vartheta \leqslant 2\pi$. Let us name ϑ "source point" and ϑ' "target point," and let us assume that the resulting function $I(\vartheta')$ is sought to be calculated on the same, non-uniformly spaced set of points on which the values of the source function are given.

A direct calculation of Eq. (1) as N separate integrals over N points requires a time $O(N^2)$, much larger than that required for the same number of points by the FFT algorithm (which, however, only works for uniformly spaced points). A faster computation may be obtained if the discretization points can be grouped in a way that reflects the existence of multiple scales in the problem. Let us start from the observation that the Green function of a conjugation problem is generally singular at the target point but regular everywhere else, and, one could say, increasingly smooth with increasing distance from the target point. This may be directly verified to be true for the function $g(\vartheta) = (2\pi)^{-1} \operatorname{cotg}(\vartheta/2)$ but is not at all an exclusive property of the circle.

This observation hints at the fact that small-scale details of the source function $f(\vartheta)$ are important near the target point, but quite irrelevant far from it. More generally, it may be expected that in the neighbourhood of any source point ϑ details on a scale that is small compared to the distance of ϑ from ϑ' may be neglected. Following this idea, we envision an algorithm in which the circumference is divided up into intervals of varying size, each having a size proportional to its distance from the target point, and a functional representation of some kind is adopted for $f(\vartheta)$ and $g(\vartheta)$ such that only a fixed number M of parameters, independent of the number of discretization points, is sufficient to represent them inside each interval.

To be more precise, let us rewrite Eq. (1) as

$$I(\vartheta') \simeq \sum_{k=1}^{K} \int_{\vartheta_{k-1}}^{\vartheta_k} f(\vartheta) \sum_{m=0}^{M} G_{mk}\, \varphi_{mk}(\vartheta' - \vartheta)\, d\vartheta, \tag{2}$$

where the K intervals $(\vartheta_{k-1}, \vartheta_k)$, with $\vartheta_K = \vartheta_0 + 2\pi$, span the whole circumference, and G_{mk} are the coefficients of an expansion of the kernel $g(\vartheta' - \vartheta)$ with respect to the set of functions φ_{mk}. Equation (2) can, of course, be rewritten as

$$I(\vartheta') = \sum_{k=1}^{K} \sum_{m=0}^{M} G_{mk} M_{mk}, \tag{3}$$

where

$$M_{mk} = \int_{\vartheta_{k-1}}^{\vartheta_k} f(\vartheta)\, \varphi_{mk}(\vartheta' - \vartheta)\, d\vartheta. \tag{4}$$

The time required for the calculation of Eq. (3) for each target point is proportional to KM, and for all target points to NKM. If we now assume it to be true, as will be shown to be in the particular case of the circumference, that a given precision in the representation of $g(\vartheta' - \vartheta)$ may be achieved with a fixed number M of terms in the expansion, provided the intervals $(\vartheta_{k-1}, \vartheta_k)$ have a size proportional to their distance from the target point, it will be seen that this time is roughly proportional to $N \log N$. In fact, the requirements that the intervals should be contiguous and the size of each interval approximately proportional to its distance from

the target point imply that the interval sizes form an approximate geometric progression, i.e., that $\vartheta_k - \vartheta_{k-1} \simeq Aq^k$ for some A and q with $q < 1$ (actually, two geometric progressions, one on each side of the target point). If we carry on this progression up to the point where about M discretization points are left between ϑ_K and the target point, so that the integral from ϑ_K to the target point may be computed in a time $O(M)$ anyhow, the resulting number of intervals K will turn out to be proportional to the logarithm of the size of the interval that contains M points, which means, at least for uniform discretization, to $\log(N/M)$. (We shall assume this estimate to be representative of the case of non-uniform spacing too. Whether such an assumption is true or not in the limit for $N \to \infty$ depends on the particular non-uniform distribution chosen and on the particular way in which it is changed in order to let N go to infinity. However, there is always a sequence of distributions for which it is true.) If M is considered to be fixed, this is enough to show that the computation time NKM of Eq. (3) is $O(N \log N)$. In order that the exact value of the integral of Eq. (1) be approached for $N \to \infty$, however, M should be allowed to grow together with N in such a way that the error of the expansion (2) may remain comparable to the error caused by the discretization of the source function f. An asymptotic estimation of these errors for N and M tending to infinity is complicated and would lead us too far from the scope of this paper. In practice, since the error decreases rapidly with increasing M, the effect of M can change the above estimate only very slightly.

Of course, the argument just exposed is valid provided that the time necessary to evaluate the coefficients G_{mk} and the integrals M_{mk} is negligible or, at most, comparable with respect to that necessary for Eq. (3). Indeed, a choice of the functions φ_{km} may be found for which this is so. The key to a rapid calculation of the integrals M_{mk} is to calculate explicitly the integrals extended over the smallest intervals only, simultaneously for all target points, and then to find a way to express the integrals over the larger intervals in terms of those over the smaller ones. This is possible if the basis functions φ_{mk} relating to the larger intervals are chosen so that they are linear combinations of those relating to the smaller ones.

In order to have this property, the functions φ_{mk} must be scale-invariant; i.e., they must remain similar to themselves upon a scale transformation. This requirement uniquely identifies the set of powers of ϑ, if each basis function must remain similar to itself, independently of the others, or more generally the set of polynomials, once linear combinations are allowed.

In addition, since the same basis is to serve for different target points, the basis also may need to be translation-invariant. The property of remaining similar to itself after a change of origin in ϑ uniquely identifies exponentials, real or complex, of ϑ, which, unfortunately, are not scale-invariant. Once linear combinations are allowed, however, the same property is also shared by powers and polynomials, since a given single power or polynomial of degree m becomes, upon translation, a different polynomial of degree m, which can always be expressed as a linear combination of powers or polynomials of degree less than or equal to m.

In conclusion, powers, (orthogonal) polynomials, real exponentials, and complex exponentials are all good candidates for the functions φ_{mk}. In practice, all these possibilities were tested by us and the algorithm worked equally well using any one of them. None turned out to have a definite speed advantage over the others, but the choice of simple powers led to a significantly shorter code, so that we settled on this choice for further work.

3. THE CONVOLUTION-INTEGRAL ALGORITHM

In the following we shall illustrate the algorithm resulting from the particular choice of powers for φ_{mk}, i.e.,

$$\varphi_{mk} = (\vartheta - \bar{\vartheta}_k)^m,$$

where $\bar{\vartheta}_k$ is the mean point of each interval, given by $\bar{\vartheta}_k = (\vartheta_{k-1} + \vartheta_k)/2$. Correspondingly, the integrals of Eq. (4) will become

$$M_{mk} = \int_{\vartheta_{k-1}}^{\vartheta_k} f(\vartheta)(\vartheta - \bar{\vartheta}_k)^m \, d\vartheta \tag{5}$$

which are usually called "moments" of the function f.

In order to attain a fast computation of all the moments required for Eq. (3), our algorithm first creates a database of moments relating to a properly chosen set of intervals of varying sizes and then calculates the actual convolution integrals according to Eq. (3), in sequence for all target points, extracting the moments it needs from the database. For speed of access the database is given a tree structure, with every record that represents a larger interval containing links to the records that represent the smaller parts into which it is subdivided. A convenient choice turns out to be a binary-tree structure, with a first interval which coincides with the whole circumference divided in two parts, each of which is divided in two, and so on down to intervals small enough that they contain about M points each, which are not subdivided further. Of course, with a non-uniform point set initially assigned, the terminal intervals will turn out to be of different sizes in different parts of the circumference, in a way that follows the local density of points.

An overall flowchart of the convolution algorithm is shown in Fig. 1. The source function is first interpolated by a periodic cubic spline and the exact moments of the spline are calculated on the smallest intervals. The moments are then translated to different centers $\bar{\vartheta}_k$, by a recursive

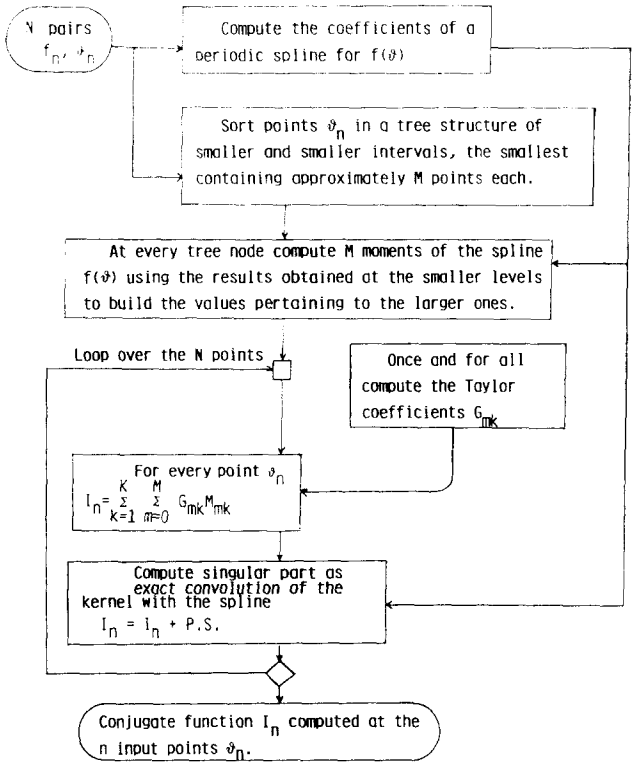

FIG. 1. Flowchart of the convolution algorithm for numerical function conjugation on a non-uniformly spaced discretization set.

formula derived from Horner's rule (see Appendix A), and combined two at a time over larger and larger intervals. Finally, Eq. (3) is computed for all target points by traversing the tree structure and selecting the largest intervals such that their size is less than a given fraction of their distance from the target point; the coefficients G_{mk} may be calculated either initially or on demand by a fast iterative rule derived from one of the differential equations of which the cotangent is a solution (described in Appendix B). The "singular" interval left over about the target point, where the expansion (2) does not apply, is dealt with by a Laurent expansion of the kernel, calculated once and for all by an iterative rule (also given in Appendix B), of which the regular terms are related to moments of the source function just as everywhere else, whilst the singular term proportional to $1/(\vartheta - \vartheta')$ is exactly convoluted with the spline by using the spline coefficients of the (roughly M) points falling in this interval.

In a typical run, computation time turns out to be subdivided as follows: about 60% of the time is spent calculating Eq. (3), 30% for the singular intervals, and 10% calculating the moments. Minor fractions of the order of 1% are taken by the initial setup of the tree structure and by the calculation of the coefficients of the periodic spline from the initially given values of the source function f.

4. CONFORMAL MAPPING AND TEST RESULTS

The just described new algorithm for convolution integrals has been combined with the quadratic-convergence Newton-style iterative technique of Ref. [11] into a general conformal-mapping algorithm with non-uniform discretization. Since it is equally easy with this algorithm to let the discrete points which are put into correspondence with each other move either on the original profile or on the circumference at each iteration, we have the possibility of fixing the discretization points on the original profile, in the way that best fits its shape, and then transforming these points, first, with the analytical pretransformations and, then, with the iterative algorithm and obtaining their correspondents on the circumference. This procedure automatically takes care of the phenomenon of crowding, because the points on the original profile are initially chosen, and the algorithm calculates their correspondents on the circumference, however close together they may turn out to be.

A flowchart of the resulting algorithm is given in Fig. 2, where each hexagonal box corresponds to an application of the FFT-less convolution algorithm.

To test the algorithm we have calculated the conformal mapping of a cascade of airfoils for several values of blade spacing. Figure 3a illustrates a cascade of NACA 4418 airfoils with a spacing equal to the chord. The airfoil profile has

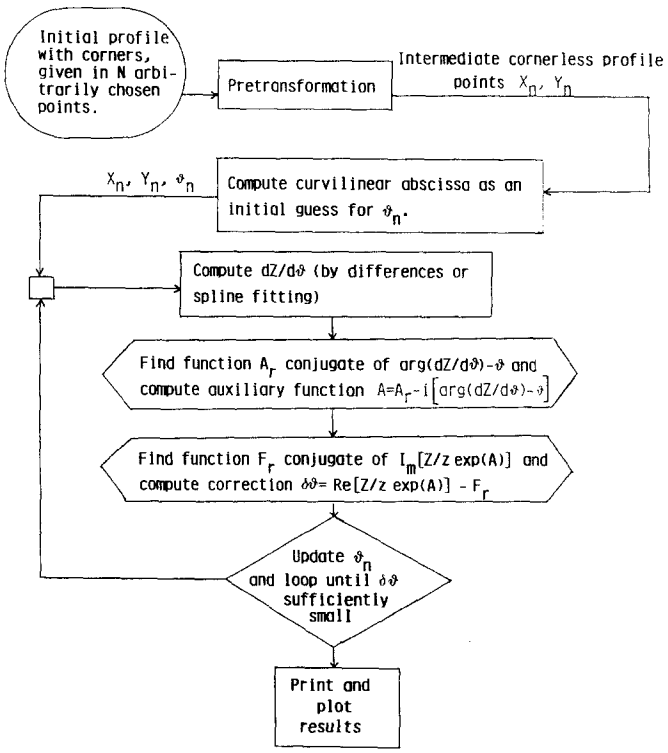

FIG. 2. Flowchart of the iterative conformal mapping algorithm of Ref. [11] modified to include the new function-conjugation procedure.

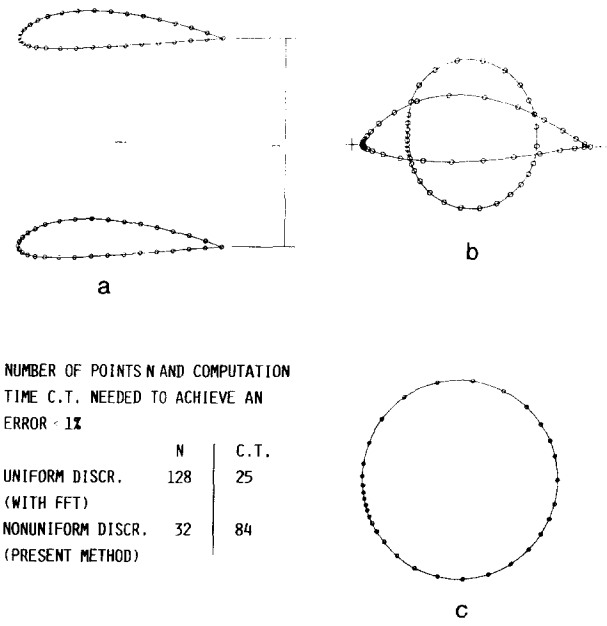

FIG. 3. Conformal mapping of an infinite cascade of NACA 4418 airfoils with a spacing equal to the chord: (a) initial profile and discretization points; (b) shapes obtained after the first and second pretransformation; (c) final position on the circumference of the discretization points.

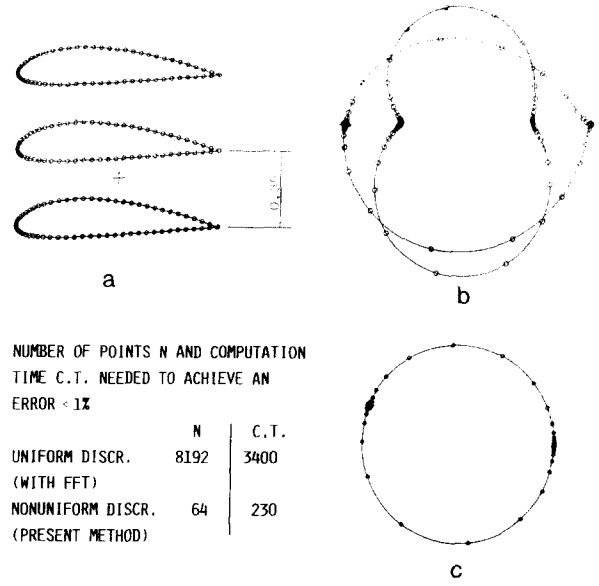

FIG. 5. Conformal mapping of an infinite cascade of NACA 4418 airfoils with a spacing equal to 0.36 times the chord: (a) initial profile and discretization points; (b) shapes obtained after the first and second pretransformation; (c) final position on the circumference of the discretization points.

been discretized in the 32 points shown in the figure and subjected to the necessary pretransformations, following the procedure outlined in Ref. [11], becoming first the intermediate single blade and then the cornerless "blob" shown together in Fig. 3b. The final location of the discrete points on the circumference calculated by the iterative algorithm appears in Fig. 3c. Similar data are given for cascades with spacing of 0.5 and 0.36 times the chord in Figs. 4–5. It may be seen that the discretization points tend to accumulate near the leading and trailing edges of the intermediate blade profile with decreasing spacing. In order to give an error comparable to that of the variable-spacing algorithm (a few parts per thousand), the standard algorithm using uniform spacing and the FFT required 128 points and 25 s in the first case, 1024 points and 270 s in the second, and 8192 points and 3400 s in the third, against 32 points and 84 s in the first two cases and 64 points and 230 s in the third for the variable-spacing method.

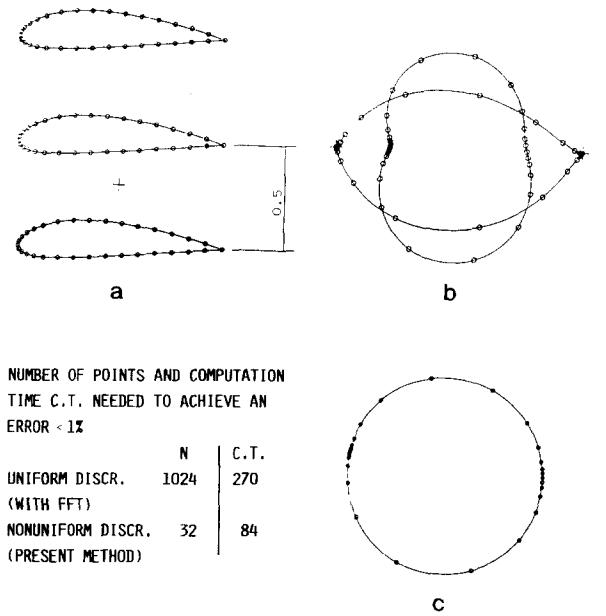

FIG. 4. Conformal mapping of an infinite cascade of NACA 4418 airfoils with a spacing equal to 0.5 times the chord: (a) initial profile and discretization points; (b) shapes obtained after the first and second pretransformation; (c) final position on the circumference of the discretization points.

5. CONCLUSIONS

A numerical conformal mapping technique has been presented which is capable of operating on a non-uniformly spaced point set thanks to a new technique for the calculation of the convolution integral that constitutes its most time-consuming step. The new convolution-integral algorithm achieves almost $O(N \log N)$ computation time by exploiting the existence of several different space scales in the problem.

To this end the calculation domain (unit circumference) is partitioned into a binary tree of intervals, that is in two parts, each of which is further subdivided in two, and so on, stopping when the smallest intervals contain approximately M points each. M moments of the source function are then

calculated on each interval by exploiting the results obtained for the smaller intervals to construct the values pertaining to the larger ones. Finally, the convolution integral is calculated for each target point from the moment information, relative to a set of intervals whose widths are roughly proportional to their distances from the target point, a set which contains $O(\log N)$ intervals. Computation time is thus $O(MN \log N)$.

Application of the new technique to conformal mapping of a cascade of blades shows that, for a blade spacing of 0.5 times the chord or less, a standard constant-spacing FFT-based conformal mapping algorithm requires an unreasonably large number of points and computation time, whereas the new variable-spacing algorithm retains a computation time comparable to the one required for larger spacing.

In addition, it is perhaps worth mentioning that the convolution integral algorithm developed in this work is quite general and lends itself to applications other then conformal mapping in several domains of computational physics.

APPENDIX A: CALCULATION OF MOMENTS OVER A TREE-LIKE SET OF INTERVALS

The moments of a function, defined as

$$M_m = \int_{\vartheta_1}^{\vartheta_2} f(\vartheta)(\vartheta - \bar{\vartheta})^m \, d\vartheta \tag{A.1}$$

are obviously additive with respect to the interval of integration, and therefore the moment over a "father" interval which is, in the tree structure, the union of two "son" intervals is simply given by the sum of the moments relating to the sons. The only problem is that the center $\bar{\vartheta}$ of the father interval is different from the center of each son, so that the moments need to be translated to a different center before adding. If we take the origin as the old center and $\bar{\vartheta}$ as the new center, the problem is that of expressing Eq. (A.1) in terms of moments calculated with respect to the origin. This is certainly possible, since Eq. (A.1) represents a polynomial in $\bar{\vartheta}$, and the result can be quickly evaluated by a formula analogous to Horner's rule which we shall now illustrate. Let us define

$$M_{mn} = \int_{\vartheta_1}^{\vartheta_2} f(\vartheta)\, \vartheta^n (\vartheta - \bar{\vartheta})^m \, d\vartheta. \tag{A.2}$$

We have

$$\vartheta^n(\vartheta - \bar{\vartheta})^m = \vartheta^{n+1}(\vartheta - \bar{\vartheta})^{m-1} - \bar{\vartheta}\vartheta^n(\vartheta - \bar{\vartheta})^{m-1} \tag{A.3}$$

and, therefore,

$$M_{mn} = M_{m-1,n+1} - \bar{\vartheta} M_{m-1,n}. \tag{A.4}$$

Using Eq. (A.4) and starting with M_{0n} given for $0 \leqslant n \leqslant M$, we can successively calculate M_{1n} for $0 \leqslant n \leqslant M-1$, M_{2n} for $0 \leqslant n \leqslant M-2$, and, finally, M_{m0} for $0 \leqslant m \leqslant M$, which is the required result.

APPENDIX B: TAYLOR AND LAURENT EXPANSION OF COTG(ϑ/2)

In order to apply Eq. (2) with $\varphi_{mk} = (\vartheta - \vartheta_k)^m$ we need a quick way to calculate the Taylor coefficients G_{mk} of $g(\vartheta) = \operatorname{cotg}(\vartheta/2)$ about an arbitrary point ϑ_k. Noting that $g(\vartheta)$ obeys the differential equation

$$g' = -(1 + g^2)/2, \tag{B.1}$$

introducing formal Taylor series into each side and equating the coefficients of equal powers of $(\vartheta - \vartheta_k)$, we obtain

$$G_{1k} = -(1 + G_{0k}^2)/2 \tag{B.2}$$

$$G_{mk} = -\left(\sum_{i=0}^{m-1} G_{ik} G_{m-1-i,k}\right) \Big/ 2m \qquad (m > 1) \tag{B.3}$$

which solve the problem posed.

In addition, in order to deal with the interval that encloses the origin, we also need the Laurent expansion of g about $\vartheta = 0$. If we write $g = 2/\vartheta + g_1$ and insert this position into Eq. (B.1), we obtain the new equation

$$2g_1' = \left(1 + \frac{4}{\vartheta} g_1 + g_1^2\right), \tag{B.4}$$

whence, if $g_1(\vartheta) = \sum_{m=0}^{\infty} G_m^{(1)} \vartheta^m$,

$$G_1^{(1)} = -\tfrac{1}{6} \tag{B.5}$$

$$G_m^{(1)} = -\left(\sum_{i=0}^{m-1} G_i^{(1)} G_{m-1-i}^{(1)}\right) \Big/ (2m+4) \quad (m > 1). \tag{B.6}$$

(Note that g_1, just as g, is an odd function of ϑ and therefore all the even coefficients are zero.)

ACKNOWLEDGEMENT

Work supported by the Italian Ministry of Public Education.

REFERENCES

1. T. Theodorsen and I. E. Garrick, NACA Report No. 452, 1933 (unpublished).
2. I. E. Garrick, NACA Report No. 542, 1936 (unpublished).
3. I. E. Garrick, NACA Report No. 788, 1944 (unpublished).
4. D. C. Ives, *AIAA J.* **14**, 1006 (1976).
5. M. H. Gutknecht, *J. Comput. Appl. Math.* **14**, 31 (1986).
6. L. N. Trefethen (Ed.), *Numerical Conformal Mapping* (North-Holland, Amsterdam, 1986) (reprint of *J. Comput. Appl. Math.* **14**).
7. P. Henrici, *Applied and Computational Complex Analysis*, Vol. 3 (Wiley, New York, 1986).
8. B. A. Vertgeim, *Dokl. Akad. Nauk SSSR* **119**, (1958). [Russian]
9. O. Huebner, *Mitt. Math. Seminar Giessen* **140**, 1–32 (1979); *J. Comput. Appl. Math.* **14**, 19 (1986).
10. R. Wegmann, *J. Comput. Appl. Math.* **14**, 7 (1986); transl. of *Numer. Math.* **30**, 453 (1978). [German]
11. P. Luchini and F. Manzo, *AIAA J.* **27**, 345 (1989).
12. R. Wegmann, *J. Comput. Appl. Math.* **14**, 79 (1986).