Rapid Design of Subcritical Airfoils

Prabir Daripa

Department of Mathematics

Texas A&M University

email: daripa@math.tamu.edu

In this paper, we present a fast, efficient and accurate algorithm for rapid design of subcritical inviscid airfoils. The compressible potential flow equations in the hodograph plane are cast into a nonlinear integral equation which is solved using an iterative method in this approach. Some numerical results are presented.

1. Introduction.

We consider the problem of inverse design of airfoils in subcritical compressible flows. This problem has a long history, especially, on the numerical approaches to solving this problem ([3]-[6]). With the advent of high speed computers, considerable effort has been directed towards efficient design of airfoils because it has the potential of significantly reducing the cost of aircraft design. Moreover, similar approaches may be the applicable to the efficient design of ship hulls, turbine and compressor blades, to name just a few.

In this paper, we describe a nonlinear integral equation approach to efficient inverse design of airfoils. Numerical results are presented.

2. Formulation.

Let us consider the subcritical compressible fluid flow past an airfoil. The design problem is to find the unknown profile L_z of the airfoil from the following data:

$$q = q(s), \quad s \in [0,1] \text{ and } \theta(|z| = \infty) = \theta_\infty. \tag{1}$$

Here arclength s, speed q, and the coordinate $z = x + iy$ are dimensionless variables. Here and below, dimensionless variables are normalized appropriately by the sonic velocity and the arclength of the airfoil.

We introduce the stream function ψ and potential function ϕ through

$$\rho \vec{q} = \nabla \times (c\psi \vec{k}), \quad \vec{q} = \nabla\phi, \tag{2}$$

where c is a suitably chosen constant (see [4]). Without any loss of generality, ϕ and ψ are taken to be zero at the stagnation point $x = y = 0$ on the airfoil and ψ is taken to be zero on the airfoil. Since the airfoil is a streamline, it follows from (2) that on the airfoil

$$\psi = 0, \ \ \phi = \int q(s)ds. \tag{3}$$

Equation (3), in essence, defines the image of the airfoil in the potential plane. The eqn. (3) maps the data $q(s)$ onto a slit, L_ω, and also provides the data $q(\phi)$ on the slit. The gap in the slit is the circulation Γ.

Since the slit (the image of the unknown airfoil) and the boundary data $q(\phi)$ are known in the potential plane, it is appropriate to consider the flow equations in this plane. If we introduce the new variables

$$\omega = \phi + i\psi, \quad \tau = -\nu + i\theta, \tag{4}$$

where θ is the flow direction and $\nu = \int_1^q \frac{\beta dq}{q}$ with $\beta = 1 - M^2$, then the governing equations in this plane can be written as

$$\tau_{\overline{\omega}} = \mu \tau_\omega, \tag{5}$$

where $\mu = \frac{1-K}{1+K}$, $K = K(\nu)$ and τ_ω and $\tau_{\overline{\omega}}$ are the generalized derivatives (see [4]).

In order to avoid dealing with the infinite domain in the ω-plane, we define a conformal mapping $\sigma = \sigma(\omega)$ so that ω-plane maps onto the interior of the unit circle $\Omega = \{\sigma : |\sigma| < 1\}$ with slit $L_\omega \to \partial\Omega$. The governing equation in the circle plane is then

$$\tau_{\overline{\sigma}} = \chi \tau_\sigma, \tag{6}$$

with $\chi = \frac{\mu(\overline{\omega_\sigma})}{\omega_\sigma}$. An overbar denotes complex conjugate. We note that χ is complex and its magnitude is bounded between 0 and 1. The boundary data (1) maps into

$$\text{Real } (\tau(\sigma = \mathrm{e}^{\mathrm{i}\alpha})) = -\nu(\alpha), \quad \alpha \in [0, 2\pi], \tag{7}$$

$$\text{Imag } (\tau(\sigma = 0)) = \theta_\infty. \tag{8}$$

The solution of (6) subject to the above boundary data provides $\tau(\sigma)$. The profile L_z of the airfoil, which is the image of the unit circle $\sigma = e^{i\alpha}$, is then obtained from

$$z(\alpha) = \int_0^\alpha \frac{e^{i\theta}}{q} \frac{d\phi}{d\alpha} d\alpha. \tag{9}$$

3. Representation of the Solution.

Using the Cauchy-Green formula [1]

$$\tau(\sigma) = \frac{1}{2\pi} \int\limits_{\partial\Omega} \frac{\tau(\zeta)}{\zeta - \sigma} d\zeta - \frac{1}{\pi} \iint\limits_{\Omega} \frac{\tau_{\bar{\xi}}}{\zeta - \sigma} d\xi d\eta, \tag{10}$$

the solution of equation (6) is written as

$$\tau(\sigma) = T_1 h(\sigma) + g(\sigma), \tag{11}$$

where

$$h(\sigma) = \chi(\sigma)\tau_\sigma, \tag{12}$$

and the operator $T_1 h(\sigma)$ in (11) is defined through

$$T_m h(\sigma) = -\frac{1}{\pi} \iint\limits_{\Omega} \frac{h(\zeta)}{(\zeta - \sigma)^m} d\xi d\eta, \quad \zeta = \xi + i\eta, \tag{13}$$

where m=1 or 2. The operator $T_2 h(\sigma)$ is defined in the unit disk as a Cauchy principal value and the function $h(\sigma)$ is assumed Hölder continuous which is sufficient for our purposes here.

It should be noted that $h(\sigma)$ in (1.2) depends on τ even though such dependency is explicitly suppressed from this notation. The function $h(\sigma)$ depends on the generalized derivative $\tau_\sigma(\sigma)$ given by

$$\tau_\sigma(\sigma) = T_2 h(\sigma) + g_\sigma(\sigma), \tag{14}$$

which follows from (11) (see [4]). Therefore, solving equation (6) using the representation (11) requires evaluation of both, $T_1 h(\sigma)$ and $T_2 h(\sigma)$. A very efficient technique for evaluation of these operators is discussed in the following section (see [5],[6]).

4. Efficient Evaluation of the Singular Integrals $T_m h(\sigma)$

The present algorithm takes into account the convolution nature of these integrals and some of the properties of such convolution integrals in the Fourier space (see [5],[6] for details). This process leads to a recursive algorithm in the Fourier space that divides the entire domain into a collection of annular regions and expands the integral in Fourier series with radius dependent Fourier coefficients. Some exact recursive relations are obtained which are then used to produce the Fourier coefficients of these integrals. These recursive relations involve appropriate scaling of one-dimensional integrals in annular regions, which significantly improves the computational complexity. The desired integrals at all grid points are then easily obtained from the Fourier coefficients by the FFT (fast Fourier transform). The process of evaluation of these integrals has thus been optimized in this paper giving a net operation count of the order $O(\ln N)$ per point. For $N = 128$, this means a reduction of over two thousand times. It is worth emphasizing that computing both integrals, $T_1 h(\sigma)$ and $T_2 h(\sigma)$, costs little more than computing just $T_1 h(\sigma)$. Moreover, this algorithm has the added advantage of working in place, meaning that no additional memory storage is required beyond that of the initial data.

This algorithm is based on the following theorem ([5],[6]).

4.1 Theorem. *If $h(\sigma)$ is Hölder continuous in the unit disk with exponent γ, $0 < \gamma < 1$, $h(re^{i\alpha}) = \sum\limits_{n=-\infty}^{\infty} h_n(r)e^{in\alpha}$ and $m = 1$ or 2, then the nth Fourier series coefficient $S_{n,m}(r)$ of $T_m h(re^{i\alpha})$ can be written as*

$$S_{n,m}(r) = \begin{cases} C_{n,m}(r) + B_{n,m}(r), & r \neq 0, \\ 0, & r = 0 \text{ and } n \neq 0, \\ S_{0,m}(0), & r = 0 \text{ and } n = 0, \end{cases} \tag{15}$$

where

$$C_{n,m}(r) = \begin{cases} \frac{2(-1)^{m+1}}{r^{m-1}} \binom{-n-1}{m-1} \int\limits_0^r \left(\frac{r}{\rho}\right)^{m+n-1} h_{m+n}(\rho)d\rho, & n \leq -m, \\ 0, & -m < n < 0, \\ -\frac{2}{r^{m-1}} \binom{m+n-1}{m-1} \int\limits_r^1 \left(\frac{r}{\rho}\right)^{m+n-1} h_{m+n}(\rho)d\rho, & n \geq 0, \end{cases} \tag{16}$$

and $B_{n,m}(r)$ and $S_{0,m}(0)$ are defined as follows.

$$S_{0,m}(0) = -2 \lim_{\varepsilon \to 0} \int\limits_{\varepsilon}^{1} \rho^{1-m} h_m(\rho)d\rho, \tag{17}$$

and

$$B_{n,m}(r) = \begin{cases} 0, & m = 1, \\ h_{n+2}(r), & m = 2. \end{cases} \tag{18}$$

Use of this theorem in constructing a fast algorithm to evaluate the integral $T_m h(\sigma)$ requires following recursive relations which follow from this theorem (see [5,6]).

4.2. Recursive Relations

Corollary 4.1. *It follows from (2.2) that* $C_{n,m}(1) = 0 \quad$ *for* $n \geq 0$, *and* $C_{n,m}(0) = 0 \quad$ *for* $n \leq -m$. *We repeat from (2.2) that* $C_{n,m}(r) = 0 \quad$ *for* $-m < n < 0$ *for all values of* r *in the domain.*

Corollary 4.2. *If* $r_j > r_i$ *and*

$$C^{i,j}_{n,m} = \begin{cases} \frac{2(-1)^{m+1}}{r_j^{m-1}} \binom{-n-1}{m-1} \int\limits_{r_i}^{r_j} \left(\frac{r_j}{\rho}\right)^{m+n-1} h_{m+n}(\rho) d\rho, & n \leq -m, \\ \frac{2}{r_i^{m-1}} \binom{m+n-1}{m-1} \int\limits_{r_i}^{r_j} \left(\frac{r_i}{\rho}\right)^{m+n-1} h_{m+n}(\rho) d\rho, & n \geq 0, \end{cases} \tag{19}$$

then

$$C_{n,m}(r_j) = \left(\frac{r_j}{r_i}\right)^n C_{n,m}(r_i) + C^{i,j}_{n,m}, \qquad n \leq -m, \tag{20}$$

$$C_{n,m}(r_i) = \left(\frac{r_i}{r_j}\right)^n C_{n,m}(r_j) - C^{i,j}_{n,m}, \qquad n \geq 0. \tag{21}$$

Corollary 4.3. *Let* $0 = r_1 < r_2 < \cdots < r_M = 1$, *then*

$$C_{n,m}(r_l) = \begin{cases} \sum\limits_{i=2}^{l} \left(\frac{r_l}{r_i}\right)^n C^{i-1,i}_{n,m} & \text{for } n \leq -m \text{ and } l = 2, \ldots, M, \\ -\sum\limits_{i=l}^{M-1} \left(\frac{r_l}{r_i}\right)^n C^{i,i+1}_{n,m} & \text{for } n \geq 0 \text{ and } l = 1, \ldots, M-1. \end{cases} \tag{22}$$

4.3. Formal Description of the Fast Algorithm

Recall that the unit disk $\overline{B}(0;1)$ is discretized using $N \times M$ lattice points with N equidistant points in the circular direction and M equidistant points in the radial direction. The following is a formal description of the fast algorithm useful for programming purposes.

The Algorithm I:

INPUT: $m \geq 1$, M, N and $h(r_l e^{2\pi i k/N})$, $l \in [1, M]$, $k \in [1, N]$.

OUTPUT: $T_m h(r_l e^{2\pi i k/N})$, $l \in [1, M]$, $k \in [1, N]$.

Step 1

Set $K = N/8$, $r_1 = 0$, and $r_M = 1$.

Step 2

Compute the Fourier coefficients $h_n(r_l), \forall\, l \in [1, M]$ & $n \in [-K+m, K]$ from known values of $h(r_l e^{2\pi ik/N})$, $k = 1, 2, \ldots, N$ using the FFT.

Step 3

Compute $C_{n,m}^{i,i+1}$, $\forall\, i \in [1, M-1]$ & $n \in \{[-K, -m] \cup [0, K-m]\}$ using equation (19).

Step 4

Note: [Compute $C_{n,m}(r_l)$, $\forall\, l \in [1, M]$ & $n \in [-K, K-m]$ using corollaries 4.1 through 4.3]

set $C_{n,m}(r_1) = 0\ \forall\, n \in [-K, -m]$

do $n = -K, \ldots, -m$

do $l = 2, \ldots, M$

$$C_{n,m}(r_l) = \left(\frac{r_l}{r_{l-1}}\right)^n C_{n,m}(r_{l-1}) + C_{n,m}^{l-1,l}$$

enddo

enddo

set $C_{n,m}(r_M) = 0\ \forall\, n \in [0, K-m]$

do $n = 0, 1, \ldots, K-m$

do $l = M-1, \ldots, 1$

$$C_{n,m}(r_l) = \left(\frac{r_l}{r_{l+1}}\right)^n C_{n,m}(r_{l+1}) - C_{n,m}^{l,l+1}$$

enddo

enddo

If $m > 1$, **then**

do $n = -m+1, \ldots, -1$

do $l = 1, \ldots, M$

$$C_{n,m}(r_l) = 0.$$

enddo

enddo

end if

Step 5

Note: [Compute $B_{n,m}(r_l),\ \forall\ l \in [2, M]$ & $n \in [-K, K-m]$ using equations (18).]

If $m = 1$, **then**

set $B_{n,m}(r_l) = 0\ \forall\ l \in [2, M]$ & $n \in [-K, K-m]$.

else

do $n = -K, \ldots, K-m$

do $l = 2, \ldots, M$

$$B_{n,m}(r_l) = h_{n+2}(r_l).$$

enddo

enddo

end if

Step 6

NOTE: [Compute the Fourier coefficients $S_{n,m}(r_l), \forall\ l \in [2, M]$ & $n \in [-K, K-m]$ using equation (15).]

do $n = -K, \ldots, K-m$

do $l = 2, \ldots, M$

$$S_{n,m}(r_l) = B_{n,m}(r_l) + C_{n,m}(r_l).$$

enddo

enddo

Compute $S_{n,m}(0),\ n \in [-K, K-m]$ from (15) and (17). (2.5) for the Case 2.

Step 7

Compute $T_m h(r_l e^{2\pi ik/N}) = \sum_{n=-K}^{K-m} S_{n,m}(r_l) e^{2\pi ikn/N},\ \forall\ l \in [1, M]$ & $k \in [1, N]$.

4.4. The Algorithmic Complexity

The asymptotic time complexity is $O(MN \ln N)$ and the asymptotic storage requirement is $O(MN)$ (see [5],[6]). The algorithm is also inherently parallelizable and thus these estimates can be improved upon if the algorithm is implemented on a parallel machine which is a topic of further research.

5. Numerical Method

An iterative method for solving (6)-(8) using representation (11) and the fast algorithm for evaluation of the singular integrals has been implemented for the design of airfoil. An initial guess of the flow field starts the iteration procedure. This is easily obtained numerically using FFT by ignoring the initial contribution due to $T_1 h(\sigma)$ to the solution of (6). The solution is usually considered to have converged if $L\infty$ norm of the difference in values of a computed flow variable (such as speed) between two successive iterations were less than 10^{-5}. The number of iterations required in most cases were less than six. The converged computed flow variables were the used to compute the airfoil.

Our fast algorithm is inherently parallelizable. In order to assess the efficiency of our parallel algorithm, we also implemented the algorithm on a Ncube parallel machine with 4 and 8 nodes. some results are discussed below.

6. Results

In this section we present some results using this fast algorithm. The computations were done on serial as well as parallel (Ncube) machine using eight-digit arithmetics (single precision). The Euler pressure distribution is used to generate the airfoil and compute the free stream flow variables. The closure and overposed issues are handled very effectively.

The figure 1 shows the Euler pressure distribution on a 12Kutta-Joukowski airfoil at $M = 0.6$ and zero angle of attack. The 12are shown in figures 2 and 3 respectively. The agreement is remarkable and the computational time is of the order of a second in most cases. We have found that the numerical method is remarkably fast and the use of the fast algorithm mentioned here reduces the complexity by almost two thousand times on a single processor computer.

There is further computational savings on the multiprocessor parallel machines. Further speed up is possible on a multiprocessor parallel machine. Figure 4 shows the further speedup that we obtain on a parallel machine with 32 grid. In practice, we need less number of grid points (32×32) for the design of airfoil. We see in this figure that we obtain a speed of approximately 2.65 on a 4 node Ncube machine (dimension 2) and approximately 3.14 on a 8 node Ncube machine (dimension 4). Further speed up is difficult die to significant communication time that is necessary between different processors.

Further work is in progress and more results will be presented at the conference.

References

1. Steven R. Bell, "The Cauchy Transform, Potential Theory and Conformal Mapping," CRC Press, New York, 1992.
2. L. Bers, " Mathematical Aspects of Subsonic and Transonic Gas Dynamics," 1958.
3. P. Daripa, " An Exact Inverse Method for Subcritical Flows," Quart. Appl. Math., Vol. **XLVI**, No. 3, (1988), 505-526.
4. P. Daripa, " On Applications of a Complex Variable Method in Compressible Flows," J. Comput. Phys., **88**, No. 2, (1990), 337-361.
5. P. Daripa, " A fast algorithm to solve nonhomogeneous Cauchy-Riemann equations in the complex plane," SIAM J. Sci. Stat. Comput., **13**, No. 6, (1992), 1418-1432.
6. P. Daripa and D. Mashat, " Singular Integral Transforms and Fast Numerical Algorithms," Numerical Algorithms, To appear.