NASATM-83281

NASA Technical Memorandum 83281

AN ACCURATE METHOD FOR EVALUATING THE KERNEL OF THE INTEGRAL EQUATION RELATING LIFT TO DOWNWASH IN UNSTEADY POTENTIAL FLOW

ROBERT N. DESMARAIS

APRIL 1982

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665

AN ACCURATE AND EFFICIENT METHOD FOR EVALUATING THE KERNEL OF THE INTEGRAL EQUATION RELATING PRESSURE TO NORMALWASH IN UNSTEADY POTENTIAL FLOW

Robert N. Desmarais
NASA Langley Research Center
Hampton, Virginia 23665

SUMMARY

This paper describes an accurate economical method for generating approximations to the kernel of the integral equation relating unsteady pressure to normalwash in nonplanar flow. The method is capable of generating approximations of arbitrary accuracy. It is based on approximating the algebraic part of the non elementary integrals in the kernel by exponential functions and then integrating termwise. The exponent spacing in the approximation is a geometric sequence. The coefficients and exponent multiplier of the exponential approximation are computed by least squares so the method is completely automated. Exponential approximates generated in this manner are two orders of magnitude more accurate than the exponential approximation that is currently most often used for this purpose. Coefficients for 8, 12, 24, and 72 term approximations are tabulated in the report. Also, since the method is automated, it can be used to generate approximations to attain any desired trade-off between accuracy and computing cost.

NOMENCLATURE

a_k	coefficient in exponential approximation
b	exponent multiplier in exponential approximation
b_k	exponent in exponential approximation
c_{jk}	element of matrix of least squares normal equations
d_j	elements of right-hand-side vector of normal equations
E(b)	least squares penalty function
E(t)	error in f(t), g(t) - f(t)
E(s,r)	error in F(s,r), $\int_s^\infty e^{-irt}[g(t) - f(t)]\,dt$
f(t)	function to be approximated, equation (12)
F(s,r)	Computational form of incomplete modified Struve function occuring in planar part of kernel
$\bar{F}(s,r)$	Incomplete modified Struve function occuring in planar part of kernel
g(t)	an approximation to f(t)
G(s,r)	Computational form of incomplete modified Struve function occuring in nonplanar part of kernel
$\bar{G}(s,r)$	Incomplete modified Struve function occuring in nonplanar part of kernel
p_k	exponent coefficient in equation (A2). In this paper either $p_k = k$ or $p_k = 2^{k/m}$.
r	Frequency term occuring in argument of exponential factor ($r = (\omega/U) \cdot \sqrt{(y_o^2 + z_o^2)}$ times the variable of integration of equations (6) and (7))
s	Lower limit of integration ($s = s_1/\sqrt{(y_o^2 + z_o^2)}$)
t	variable of integration in definition of F(s,r) and G(s,r)

The symbols occuring in equations (1) thru (7) not defined above are defined in the references. They are not used elsewhere in the paper. Equations (1) thru (7) merely illustrate a use for F(s,r) and G(s,r).

INTRODUCTION

This paper describes a method for computing two incomplete modified Struve functions that occur in unsteady aerodynamics. These two functions are the nonelementary part of the unsteady kernel of the integral equation relating lift to downwash in three dimensional potential flow.

The incomplete Struve functions approximated in this report occur whenever one attempts to compute the three dimensional velocity field induced by unsteady doublet distributions. The most common occurance is when evaluating the acceleration potential kernel of the downwash integral equation as in reference 1. However, the same Struve functions occur even if the velocity potential is used, although the kernel is very different. For example, a 24 term approximation developed herein is used to compute the wake contribution in a velocity potential program called SOUSSA (reference 2).

The method developed in this paper is the familiar technique of replacing the algebraic factor in the defining integral by an exponential approximation and then evaluating the integral in closed form. The new exponential approximations described in this paper have coefficients computed by an easily automated least squares process and hence can be used to approximate the incomplete Struve functions to any desired accuracy.

The paper explains the reason for using exponential approximations instead of series expansions or direct numerical integration. It also surveys several previously published approximations and assesses their accuracy.

Coefficients for four new approximations are tabulated in the paper. The least squares process is explained in sufficient detail to enable the reader to generate his own approximations to fit the accuracy or computing time constraints of his problem.

STATEMENT OF PROBLEM

Consider the integral equation relating unsteady pressure to normalwash in three dimensional potential flow

$$w(x,s) = 1/(8\pi) \left(\sum_{n=1}^{N} \iint_{S_n} K(x,\xi;\ s,\sigma;\ \omega/U,M)p(\xi,\sigma)d\xi d\sigma \right) \quad (1)$$

A convenient representation of the kernel, K, above is the one given by Harder and Rodden (reference 3)

$$K = \exp(-i\omega x_0/U)(K_1 T_1/r^2 + K_2 T_2/r^4) \quad (2)$$

where x_0 is the distance in the free stream direction between the sending and receiving points and

$$T_1 = \cos(\gamma_r - \gamma_s) \quad (3)$$

$$T_2 = (z_0 \cos\gamma_r - y_0 \sin\gamma_r)(z_0 \cos\gamma_s - y_0 \sin\gamma_s) \quad (4)$$

$$r = (y_0^2 + z_0^2)^{1/2} \quad (5)$$

where y_0 and z_0 are the distance from sending to receiving points measured normal to the freestream and γ_r, γ_s are dihedral angles.

$$K_1 = -u(x_0 - Br)\left\{\frac{Mr^2}{R}\left[\frac{\exp(-i\omega s_2/U)}{(r^2 + s_2^2)^{1/2}} + \frac{\exp(-i\omega s_1/U)}{(r^2 + s_1^2)^{1/2}}\right] + \int_{s_1}^{s_2} \frac{r^2 \exp(i\omega s/U)}{(r^2 + s^2)^{3/2}}\, ds\right\} \quad (6)$$

$$K_2 = u(x_0 - Br)\left\{\frac{Mr^4 \exp(-i\omega s_2/U)}{R(r^2 + s_2^2)^{3/2}} \cdot\left[2 - \frac{B^2(r^2 + s_2^2)}{R^2} - \frac{Ms_2}{R}\right] - i\,\frac{M^2 r^3(\omega r/U)\exp(i\omega s_2/U)}{R^2(r^2 + s_2^2)^{1/2}} + \frac{Mr^4 \exp(-i\omega s_1/U)}{R(r^2 + s_1^2)^{3/2}}\left[2 - \frac{B^2(r^2 + s_1^2)}{R^2} + \frac{Ms_1}{R}\right] + i\,\frac{M^2 r^3(\omega r/U)\exp(-i\omega s_1/U)}{R^2(r^2 + s_1^2)^{1/2}} + 3\int_{s_1}^{s_2} \frac{r^4 \exp(i\omega s/U)}{(r^2 + s^2)^{5/2}}\, ds\right\} \quad (7)$$

The unit function, u(x), in equation (6) is one if $x > 0$ and is zero is $x < 0$. The only non elementary terms in the above expressions for K_1 and K_2 are the integrals. They can be rescaled to

$$\bar{F}(s,r) = \int_s^\infty e^{-irt}(1 + t^2)^{-3/2} dt \quad (8)$$

and

$$\bar{G}(s,r) = \int_s^\infty e^{-irt}(1 + t^2)^{-5/2} dt \quad (9)$$

These integrals are called incomplete modified Struve functions. In subsonic flow the upper limit of integration is infinity as in equations (8) and (9) above. In supersonic flow the upper limit is variable so the integrals in equations (6) and (7) are expressed as differences of incomplete Struve functions. The algebraic part of equations (8) or (9) could be approximated by an exponential function. However, to facilitate comparison with other exponential approximations the exponential approximations were used in the integrals

$$F(s,r) = \int_s^\infty e^{-irt}\left(1 - \frac{t}{\sqrt{(1+t^2)}}\right) dt \quad (10)$$

and

$$G(s,r) = \int_s^\infty e^{-irt}\, t\left(1 - \frac{t}{\sqrt{(1+t^2)}}\right) dt \qquad (11)$$

that are related to $\overline{F}$ and $\overline{G}$ by integration by parts (see reference 4 for details of the integration by parts). The function to be approximated by exponential approximations is

$$f(t) = 1 - t/\sqrt{(1 + t^2)} \qquad (12)$$

The exponential approximations are

$$g(t) = \sum_{k=1}^{n} a_k \exp(-b_k t) \qquad (13)$$

This reduces the evaluation of F and G to the evaluation of elementary integrals. Equation (13) can only be used if t is non-negative. If t is negative then $g(t) = 2 - g(|t|)$.

Replacing f(t) by an exponential approximation is not the only way to evaluate F(s,r) and G(s,r). Before electing to use this method several other techniques were investigated in some detail. Direct numerical integration was found to be too expensive. A series expansion of one or the other integral factor followed by a closed form integration is economical; but every such expansion has a very limited area of application in the s-r plane. To evaluate F(s,r) by a combination of an asymptotic series for large s (obtained by iterated integration by parts) and closed form integration of various series expansions of exp (-irt) and $t/\sqrt{(1+t^2)}$ would take at least half a dozen different algorithms. This would result in a computer subprogram that, while inexpensive to use, would be very expensive to develop, certify, and maintain.

For the reasons stated above, exponential approximations having the form shown in equation (13) are the most commonly used method for evaluating the integrals F(s,r) and G(s,r). In the next section several currently used approximations are discussed.

A SURVEY OF PREVIOUS EXPONENTIAL APPROXIMATIONS

There are two reasons for surveying previous exponential approximations to f(t). One, of course, is to point out their deficiencies to justify developing a new family of approximations. A more important reason is to show how they contributed to the development of the least squares method described herein and to show which features of previous methods have been retained.

The first use of an exponential approximation to f(t) to evaluate the integral F(s,r) was by Watkins, Woolston, and Cunningham (reference 1). They used the approximation

$$g(t) = \sum_{k=1}^{4} a_k \exp(-b_k t) \qquad (14)$$

where the a_k, b_k are tabulated below:

$$\begin{aligned} a_1 &= .101 \\ a_2 &= .899 \\ a_3 &= .0047404665\ i \\ a_4 &= -.0047404665\ i \\ b_1 &= .329 \\ b_2 &= 1.4067 \\ b_3 &= 2.9 + 3.1415926\ i \\ b_4 &= 2.9 - 3.1415926\ i \end{aligned} \qquad (15)$$

This approximation will be referred to as W4 (Watkins, 4 term). The error in this approximation to f(t), that is e(t) = g(t) - f(t), is plotted as short dashes in Figure 1.

The approximation W4 was derived long before the ready availability of high speed computing machinery. The coefficients, a(k), and exponents, b(k), were generated as follows:

1. A large scale plot of the function $f(t) = 1-t/\sqrt{(1 + t^2)}$ was constructed on graph paper.
2. Several plots of a exp(-bt) for different values of a and b were drawn on the same sheet of paper. Eventually, by trial and and error, an a, b were found that appeared to give a best fit.
3. The difference between the function being fitted, f(t), and the approximation a exp (-bt) was drawn on a large scale plot.
4. Steps 2 and 3 above were repeated several times, using successive difference plots, to give an exponential series with several terms.

A more commonly used approximation to f(t) is the 11 term exponential polynomial

$$g(t) = \sum_{k=1}^{11} a_k e^{-kbt} \qquad (16)$$

due to Laschka (reference 5). The exponent, b, and the coefficients, a_k, appearing in this approximation are tabulated below:

$$\begin{aligned} b &= .372 \\ a_1 &= .24186198 \\ a_2 &= -2.7918027 \\ a_3 &= 24.991079 \\ a_4 &= -111.59196 \\ a_5 &= 271.43549 \\ a_6 &= -305.75288 \\ a_7 &= 41.1836 \\ a_8 &= 545.98537 \\ a_9 &= -644.78155 \\ a_{10} &= 328.72755 \\ a_{11} &= -64.279511 \end{aligned} \qquad (17)$$

This approximation will be referred to as L11. In L11, instead of 11 unrelated exponents b_k, there is a single exponent, b, and its multiples. This means that even though L11 has many more terms than W4 the integral F(s,r) can be evaluated almost as fast using L11 as using W4 because only a single exponential function has to be evaluated. Figure 1 (long dashes) also shows that L11 is more accurate than W4. The computer code to approximate the integral F(s,r) is somewhat easier to write using the approximation L11 than it is using the approximation W4. This is probably why the approximation L11 is so widely used.

Reference 5 does not state how the exponent, b, or the coefficients, a_k, in L11 were computed. Inspection of Figure 1 leads one to suspect that they were computed in essentially the same manner as the coefficients and exponents of W4. There are 12 free parameters in this approximation, 1 exponent and 11 coefficients. If they had been computed by some sort of systematic optimization procedures such as least squares or minimax then the number of nodes (points where e(t) = 0) on the error plot, Figure 1, would approximately equal the number of free parameters.

The approximation D11 has been in use for several years in a computer program based on reference 1 but the coefficients have not been published. It has the same form as L11, namely equation (16). However the coefficients and exponent were computed to minimize the integrated square error

$$E = \int_0^\infty w(t)\,(g(t) - f(t))^2 dt \qquad (18)$$

using the weight function $w(t) = t^{-1/2}$. The reason for this choice will be discussed later. The exponent and coefficients for D11 are tabulated below:

$$\begin{aligned} b &= .1539 \\ a_1 &= .0545 \\ a_2 &= -.8049 \\ a_3 &= 6.3712 \\ a_4 &= -10.8582 \\ a_5 &= -130.6866 \\ a_6 &= 915.2209 \\ a_7 &= -2793.8215 \\ a_8 &= 4800.8897 \\ a_9 &= -4792.6556 \\ a_{10} &= 2596.7163 \\ a_{11} &= -590.2260 \end{aligned}$$

The least squares procedure used to generate D11 was very easy to implement because there is only one non-linear parameter, b. The form of the approximation L11 was chosen to minimize the execution time of the computed approximation but it also makes the coefficients in the approximation easier to compute.

Figure 1 (solid curve) indicates that D11 is much more accurate than L11. Since it has the same form as L11 a program that uses L11 can be converted to one using D11 by only changing one or two data statements. This similiarity in form is the only reason for tabulating the D11 coefficients because it will be shown later that much more accurate approximations can be obtained by changing the expression for the exponents b_k to something other than k·b. Before considering this, however, it's instructive to see what happens to an approximation such as D11 when the number of terms, n is increased.

Consider the approximation

$$g(t) = \sum_{k=1}^{n} a_k\, e^{-kbt} \qquad (20)$$

where the exponent b and the coefficient a_k are chosen to minimize the integral E of equation (18). Figure 2 shows the e(t) vs. t plots for three of these approximations. For the three approximations shown the maximum error decreases with n as expected. The ragged behavior of the 24 term approximation for t near zero is due to cancellation errors. The coefficients a_k are very large while both the function being approximated f(t) and the approximation functions exp(-kbt) are near unity. For n slightly greater than 24 this cancellation due to large coefficients becomes the main source of error in these approximations and no further gain in accuracy is attainable using $b_k = k \cdot b$ unless the computer word length is increased.

The solution to this cancellation error problem is suggested by an exponential approxiation due to Jordan (ref. 6). The e(t) vs. t plot for this approxiation, called J10, is shown as the solid curve in Figure 3.

This approximation is of the form

$$g(t) = a_0 \exp(-b_0 t) + \sum_{k=1}^{9} a_k\, e^{-2^k bt} \qquad (21)$$

with exponents and coefficients tabulated below:

$$\begin{aligned} b_0 &= 3 \\ a_0 &= .7048426 \\ b &= .0625 \\ a_1 &= .002907843 \\ a_2 &= .002591528 \\ a_3 &= .02667074 \\ a_4 &= .070971 \\ a_5 &= .347837 \\ a_6 &= .5556069 \\ a_7 &= -.776979 \\ a_8 &= .07004561 \\ a_9 &= -.004557519 \end{aligned} \qquad (22)$$

Although the exponents b_k in this approximation are somewhat more complicated than in L11 or D11, the sum in equation (21), when substituted into the integral F(s,r), can still be evaluated without repeated calls to the exponentiation subprogram. This simplification occurs because $\exp(-2^k bt)$ is equal to $(\exp(-2^{(k-1)} \cdot b \cdot t))^2$ so that the exponential functions can be computed recursively as in L11.

The approximation J10 is much more accurate than L11 and is somewhat more accurate than

D11. It would take a 15 term approximation similar to D11 to attain the accuracy of J10. However it isn't the accuracy of J10 that is important. It is the fact that the coefficients, a_k, in J10 are all bounded by unity so cancellation errors are very small.

Reference 6, like reference 1 and 5, does not state how the exponents and coefficients were computed. The coefficients in J10 were probably computed in a manner similar to that used for the W4 coefficients. This is indicated by the fact that the number of nodes shown in figure 3 (eight) is much less than the number of free parameters (twelve). By contrast, the approximations D8.1 and D12.1, developed using the methods of the next section, and also plotted in figure 3, have the same number of nodes as free parameters.

EXPONENTIAL APPROXIMATIONS OF ARBITRARY ACCURACY

In this section a family of new exponential approximations of the form

$$g(t) = \sum_{k=1}^{n} a_k \exp\left(2^{k/m} bt\right) \qquad (23)$$

will be investigated. For each of these approximation the exponent b and the coefficients a_k were computed to minimize the integral E of equation (18). The coefficients a_k enter into the error minimization process linearly and the exponent b enters nonlinearly. As a consequence, for each value of n and m there are several values of b for which E is a relative minimum. Of these relative minima the value of b chosen was not the value which minimized E in the absolute sense but rather, the values that minimized the maximum values of $|e(t)| = |g(t) - f(t)|$.

Exponents and coefficients for four of these approximations, designated D8.1, D12.1, D24.2, and D72.3 (Dn.m) are tabulated in TABLE I. The error e(t) in D8.1 and D12.1 is also plotted in Figure 3. A summary of some of the properties of these approximations is listed in TABLE II. Properties of the approximations discussed previously are also included in TABLE II.

Approximations D8.1 and D12.1 can be used to rapidly evaluate the integral F(s,r). The execution times (TABLE II) are comparable to those of approximations, W4, L11/D11, and J10. However D8.1 and D12.1 are much more accurate. The author knows of no study to determine how accurately the integral F(s,r) has to be computed for various aeroelastic applications. Approximation D24.2 permits computing the integral F(s,r) to about 3.5 more significant figures than D12.1. This is about full register accuracy on a short word length computer. Approximation D72.2 is probably more accurate than is needed for any engineering purpose. These coefficients are included because they were used to assess the error of the other approximations.

The form of the expression chosen for the exponents, namely

$$b_k = 2^{k/m} b \quad . \qquad (24)$$

was dictated to a large extent by the approximations discussed in the previous section. In L11 the form k·b was introduced to reduce the number of exponential function evaluations that occured when evaluating F(s,r). However this choice also reduced the number of nonlinear parameters in the least squares error minimization process used to generate D11. This made it possible to automate the process. The choice, $b_k = 2^k \cdot b$, suggested by J10, is almost as good as that of D11 for convenience in computing the integral F(s,r) and is better for the least squares process. This is because the matrix of the normal equations has a much lower condition number when $2^k \cdot b$ is used than it has when k·b is used.

The choice, $b_k = 2^k \cdot b$, corresponding to m = 1 in equation (23), was used to generate the approximation D8.1 and D12.1. However when an attempt was made to generate D24.1 this way the result was only slightly more accurate than D12.1. The exponent, b enters into the least squares process in a nonlinear manner so there is more than one value of b for which E = E(b) is a relative minimum. When n is 12 there are eight of these relative minima and the lowest value of E is 1.56E-9. Increasing n to 24 increases the number of relative minima to twenty but only reduces the lowest value of E to 9.07E-10. In fact, when n is 24 there are ten relative minima with integrated square errors, E, between 9.07E-10 and 9.9E-10. As n is increases a value of n is reached for which increasing n increases the number of relative minima without decreasing the value of E at the lowest relative minimum. This happens because when n is increased from 12 to 24 the value of b was rescaled so that effectively the additional b_k were all inserted ahead of b_1, where their effect was to reduce the $|e(t)|$ for t near zero, and they were also inserted after b_{12}, where their effect was to reduce $|e(t)|$ for t very large. None of them were interpolated between the b_k to reduce $|e(t)|$ in the middle of its range. Changing b_k from $2^k \cdot b$ to $2^{k/2} \cdot b$ has the desired effect of reducing $|e(t)|$ in the middle of its range. When n is set to 24 and m is set to 2 only seven relative minima are obtained and the value of E at the lowest is 1.78E-12, a significant improvement over m = 1. One might think that since m = 2 is an improvement when n = 24, that m = 3 would be even better. When m was set to 3 the number of relative minima was reduced to four but the lowest value of E was increased to 1.9E-10. Also for n = 24 and m = 3 the coefficients a_k were very large so some cancelation errors can occur. For most values of n there is a single optimum value of m. If there are two optimum values of m for a particular value of n then the lower should be chosen because evaluating F(s,r) requires m exponential function evaluations.

Exponential approximations of the form used in equation (23) can be used to generate approximations to f(t) that have arbitrary

accuracy. All that has to be done is to keep increasing the number of terms n and whenever this fails to increase the accuracy, then increase the separation parameter m. There are no cancelation problems due to the a_k becoming extremely large as happens when $b_k = k \cdot b$. The normal equations of the least squares process have matrices whose condition number increases with n but the rate of increase is very slow.

DISCUSSION

The following recommendations regarding the use of exponential approxiamtions are based on the information in table II:

1. If execution time is very important use approximation D8.1.

2. Any programs for which execution time is not critically important that currently use approximations W4 or L11 should be changed to use approximation D12.1. This will reduce the error by two order of magnitude at the expense of a negligible increase in execution time. In fact, it was found when converting a particular program from L11 to D12.1 that the execution time was decreased. This was because during the recoding some compiler generated complex arithmetic was replaced by separate evaluation of real and imaginary parts.

3. New programs should use approximation D24.2. It gives full word length accuracy on short word length computers and takes less than twice as long to execute as D12.1.

4. Programs that currently use approximation J10 should only be converted to D12.1 if a considerable amount of future use of the program is anticipated. Approximation D12.1 is no faster than J10 and is only an order of magnitude more accurate.

In non-planar flow two incomplete modified Struve functions are required. They both use the same exponential approximation but the non-planar term has an extra variable of integration factor. This acts as a weight function on the error. As a consequence the exponential approximations described in this report introduce more error into the non-planar part of the kernel than into the planar part. This was not investigated in any detail. It is felt that it is not very important because all successful non-planar programs, such as the various doublet lattice programs, use very large panels so the kernel is either planar or is attenuated by distance.

The integrals F(s,r) and G(s,r) are integrated by merely substituting g(t), defined by equation (23), for f(t), in equations (10) or (11) and then integrating termwise. Thus, if $s \geq 0$

$$F(s,r) \approx e^{-irs} \sum_{k=1}^{n} a_k e_k / (2^{k/m} b + ir) \qquad (25)$$

$$G(s,r) \approx sF(s,r) - e^{-irs} \sum_{k=1}^{n} a_k e_k / (2^{k/m} b + ir)^2 \qquad (26)$$

where

$$e_k = \exp(-2^{k/m} bs) \qquad (27)$$

The exponential factors in equations (25) and (26) should be computed from equation (27) only if $k < m$. If k is greater than m then e_k should be computed from the recursion

$$e_k = (e_{k-m})^2 \qquad (28)$$

Occasionally it is necessary to compute F(s,r) and G(s,r) for complex r. This occurs when the frequency ω in equation (1) has an imaginary component. If $\mathrm{Im}\{r\} \leq 0$ then the integrals (10) and (11) exist and the sums (25) and (26) are valid approximations. A minor problem is that the error estimates given by the third column of table II may be too low because they are based on a search of the s-r plane for real s and r. If $\mathrm{Im}\{r\} > 0$ then the integrals (10) and (11) do not exist. This is because the oscillations are damped and the integrals (10) and (11) usually represent integration over the subsonic wake. Even if $\mathrm{Im}\{r\} > 0$ the sums (25) and (26) exist. They are approximation to Abel sums of the divergent integrals (10) and (11). The Abel sum of an exponentially diverging integral is obtained by inserting a factor $\exp(-\epsilon t^2)$ into the integrand and taking the limit as ϵ approaches zero. If it can be shown on physical grounds that the integrals should exist then they are defined by their Abel sums.

CONCLUDING REMARKS

Several different methods for evaluating the incomplete modified Struve functions that occur in unsteady aerodynamics were investigated. Of the methods investigated it was found that an exponential approximation to the algebraic part of the defining integral followed by a closed form integration furnished the best compromise among accuracy, execution speed, and code maintainability. If the exponent spacing of the exponential approximation is either an arithmetic sequence, as suggested by Laschka, or a geometric sequence, as suggested by Jordan, then the number of exponential function evaluations required to evaluate the incomplete Struve functions is reduced. This same sort of spacing also simplifies the evaluation of the coefficients of the approximation if least squares are used. Of the two spacing sequences the geometric is much better. For either spacing sequence least squares provides an easily automated way of computing the coefficients, a_k and the sequence multiplier b. Approximations with coefficients computed by least squares are more accurate than those of the same order with coefficients computed by manual curve fitting.

In the paper a very limited class of exponential approximations was investigated, namely those with arithmetic and geometric spacing exponent sequences and with coefficients computed by least squares. It is possible that other sequences and schemes for computing the coefficients would give more accuracy. In particular using a minimax algorithm to generate the coefficients and exponent multiplier would

certainly give a small increase in accuracy. A program to implement a minimax algorithm is much harder to write and is much more expensive to execute than a least squares program. Furthermore the increase in accuracy would only be very slight. However if a minimax program were available it could be used to generate exponential approximations over subintervals of the positive real axis with no greater effort than is required to fit the whole axis. This is not true for least squares because of the integrals that have to be computed (the ones that contained quarter order Bessel functions). Splitting the interval into subintervals can lead to a large increase in accuracy with no increase in computing time. The only penalty is in program maintenance and development cost.

REFERENCES

[1]Watkins, C. E.; Woolston, D. S.; and Cunningham, H. J.: A Systematic Kernel Function Procedure for Determining Aerodynamic Forces on Oscillating or Steady Finite Wings at Subsonic Speeds. NACA TR R-48, June 1959.

[2]Morino, L.: Steady, Oscillatory, and Unsteady Subsonic and Supersonic Aerodynamics -- Production Version (SOUSSA P1.1) Vol. 1 -- Theoretical Manual. NASA CR 159130, January 1980.

[3]Harder, R. L.; and Rodden, W. P.: Kernel Function for Nonplanar Oscillating Surfaces in Supersonic Flow. Journal of Aircraft, Vol. 8, Aug. 1971, pp. 677-679.

[4]Giesing, J. P.; Kalman, T. P.; and Rodden W. P.: Subsonic Unsteady Aerodynamics for General Configurations. AFFDL-TR-75-5, Part 1, Vol. 1, November 1971.

[5]Laschka, B.: Interfering Lifting Surfaces in Subsonic Flow. Presented at 29th AGARD Structures and Materials Panel Meeting, Istanbul, Turkey, 1969.

[6]Jordan, P. F.: Numerical Evaluation of the Three-Dimension Harmonic Kernel. 2.Flugwiss.24(1976), Heft4, pp. 205-209.

[7]Erdeyli, et. al.: Table of Integral Transforms, Vol. 1. McGraw-Hill, New York, 1964.

[8]Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables, NBS Applied Math. series No. 55, edited by M. Abramowitz and I. Stegun. U.S. Dept. of Commerce, June 1964.

TABLE I

Exponent multiplier, b and coefficients, a_k, for several approximations to $1-t/\sqrt{(1+t^2)}$ of the form indicated by equation (23).

Approximation D8.1

n = 8 m = 1 b = .035003907466

a_1 = .004329519485	a_2 = .001601370746		
a_3 = .033195062769	a_4 = .098682301170		
a_5 = .376739860841	a_6 = .822464185014		
a_7 = -.380262739620	a_8 = .043400039240		

Approximation D12.1

n = 12 m = 1 b = .009054814793

a_1 = .000319759140	a_2 = -.000055461471		
a_3 = .002726074362	a_4 = .005749551566		
a_5 = .031455895072	a_6 = .106031126212		
a_7 = .406838011567	a_8 = .798112357155		
a_9 = -.417749229098	a_{10} = .077480713894		
a_{11} = -.012677284771	a_{12} = .001787032960		

Approximation D24.2

n = 24 m = 2 b = .005209230865

a_1 = .000305311497	a_2 = -.001412280807		
a_3 = .003845227615	a_4 = -.007196572664		
a_5 = .011385147609	a_6 = -.014763498650		
a_7 = .018969114027	a_8 = -.019842326360		
a_9 = .025618710871	a_{10} = -.020313397232		
a_{11} = .036575115249	a_{12} = -.010202806435		
a_{13} = .069407344423	a_{14} = .037308217964		
a_{15} = .177803740980	a_{16} = .198282197469		
a_{17} = .433959048197	a_{18} = .354218469431		
a_{19} = .104676453558	a_{20} = -.715978991168		
a_{21} = .407542943867	a_{22} = -.104393578248		
a_{23} = .015398943987	a_{24} = -.001192670868		

Approximation D72.3

n = 72 m = 3 b = .000065986269

a_1 = .000000487572	a_2 = -.000003844799		
a_3 = .000015710073	a_4 = -.000044143564		
a_5 = .000096360019	a_6 = -.000174937155		
a_7 = .000276395746	a_8 = -.000392188471		
a_9 = .000511902333	a_{10} = -.000625480027		
a_{11} = .000725938341	a_{12} = -.000808476891		
a_{13} = .000872553959	a_{14} = -.000917456689		
a_{15} = .000947455433	a_{16} = -.000961741082		
a_{17} = .000968885391	a_{18} = -.000962841735		
a_{19} = .000960418999	a_{20} = -.000941494817		
a_{21} = .000943838490	a_{22} = -.000912640747		
a_{23} = .000939494732	a_{24} = -.000883200418		
a_{25} = .000971868480	a_{26} = -.000847331705		
a_{27} = .001082186979	a_{28} = -.000771912330		
a_{29} = .001357091273	a_{30} = -.000556797879		
a_{31} = .001997471929	a_{32} = .000067960260		
a_{33} = .003487618691	a_{34} = .001801105035		
a_{35} = .007010443761	a_{36} = .006406650025		
a_{37} = .015440810290	a_{38} = .018199043007		
a_{39} = .035536908427	a_{40} = .047032188464		
a_{41} = .081594062558	a_{42} = .111356164158		
a_{43} = .174138343702	a_{44} = .222492227365		
a_{45} = .288676153073	a_{46} = .263088797407		
a_{47} = .143795607125	a_{48} = -.194655408459		
a_{49} = -.414538285804	a_{50} = -.093514761246		
a_{51} = .562053915950	a_{52} = -.395454683729		

a_{53} =	.132558644137	a_{54} =	-.010481139795
a_{55} =	-.022964836837	a_{56} =	.027218864137
a_{57} =	-.024548039318	a_{58} =	.020804555835
a_{59} =	-.017250853224	a_{60} =	.014078375164
a_{61} =	-.011265419745	a_{62} =	.008772771052
a_{63} =	-.006583208288	a_{64} =	.004702560548
a_{65} =	-.003149336650	a_{66} =	.001939706215
a_{67} =	-.001071671331	a_{68} =	.000513691017
a_{69} =	-.000203826307	a_{70} =	.000062322287
a_{71} =	-.000012950446	a_{72} =	.000001360075

TABLE II
Properties of various approximations to $1 - t/\sqrt{(1 + t^2)}$.

	max $\lvert e(t) \rvert$	max $\lvert E(s,r) \rvert$	time
W4	1.6E-3	2.3E-2	355
L11	1.3E-3	1.8E-2	382
D11	3.5E-4	2.6E-3	382
J10	1.3E-4	1.7E-3	407
D8.1	1.6E-4	1.1E-3	293
D12.1	2.5E-5	1.9E-4	411
D24.2	3.5E-7	2.1E-6	768
D72.3	3.0E-10	------	2250

The first column of the table is the approximation designator.

The second column is the maximum error in the integrand approximation. It is very easy to compute but not too closely related to the error in the integral.

The third column is the maximum error in the integral for the planar component only. It is extremely expensive to compute.

The fourth column is the computer execution time for a single evaluation of the planar integral with positive lower limit and infinite upper limit. It is expressed as a multiple of the computer's add time.

APPENDIX

THE LEAST SQUARES PROCEDURE FOR COMPUTING COEFFICIENTS AND EXPONENTS

The function

$$f(t) = 1 - \frac{t}{\sqrt{(1 + t^2)}} \qquad \text{(A1)}$$

is to be approximated by an exponential sum

$$g(t) = \sum_{k=1}^{n} a_k \exp(-bp_k t) \qquad \text{(A2)}$$

The number of terms, n, is preassigned as are the exponent coefficients p_k. Although the only p_k arrays for which calculations have been performed are $p_k = k$ and $p_k = 2^{k/m}$ the least squares procedure is valid for any preassigned p_k array. The only restriction is that all the p_k be distinct.

In the least squares procedure the coefficients a_k and the exponent b in equation (A2) are chosen to minimize the integral

$$E = \int_0^\infty w(t)\,[g(t) - f(t)]^2\,dt \qquad \text{(A3)}$$

There is no reason to suppose that this is an optimum choice for the a_k and b. In fact, there is some evidence the minimizing the maximum error $\lvert g(t) - f(t) \rvert$ would be a better choice. However, minimizing E certainly gives a good set of coefficients and exponent and the process is very easy to automate. If the weight function w(t) in equation (A3) is set to 1 then the absolute value of the error function

$$\lvert e(t) \rvert = \lvert g(t) - f(t) \rvert \qquad \text{(A4)}$$

takes on its maximum values near the ends of the approximation interval. The weight function

$$w(t) = (t-a)^{-1/2}(b-t)^{-1/2} \qquad \text{(A5)}$$

is frequently used to make a least squares approximation over a finite interval (a,b) closely resemble a minimax approximation. In equation (A3) $b = \infty$ and $a = 0$ so the $(b-t)^{-1/2}$ factor is undefined and the $(t-a)^{-1/2}$ factor is

$$w(t) = t^{-1/2} \qquad \text{(A6)}$$

Use of this weight function also makes some required integrals easier to evaluate.

The coefficients a_k and the exponent multiplier b are computed by setting $\partial E/\partial a_k$ to zero for any value of b and by setting dE/db to zero where the a_k are functions of b defined by the condition $\partial E/\partial a_k = 0$.

Let $$\frac{1}{2}\frac{\partial E}{\partial a_\ell} = 0 \qquad \text{(A7)}$$

Then

$$\int_0^\infty t^{-1/2}\,[g(t) - f(t)]\exp(-bp_\ell t)\,dt = 0 \qquad \text{(A8)}$$

That is

$$\sum_{k=1}^{n} a_k \int_0^\infty t^{-1/2} \exp[-b(p_\ell + p_k)t]\,dt = \int_0^\infty t^{-1/2} \exp(-bp_\ell t)\, f(t)\,dt \qquad \text{(A9)}$$

The Laplace transform of $t^{-1/2}$ is

$$\int_0^\infty e^{-pt}\, t^{-1/2}\,dt = \sqrt{\pi}\, p^{-1/2} \qquad \text{(A10)}$$

so equation (A9) can be written

$$\sum_{k=0}^{n} c_{\ell k} a_k = d_\ell \qquad \text{(A11)}$$

where

$$c_{\ell k} = (p_\ell + p_k)^{-1/2} \qquad \text{(A12)}$$

$$d_\ell = \sqrt{(b/\pi)}\, H(bp_\ell/2) \qquad \text{(A13)}$$

and

$$H(y) = \sqrt{(\tfrac{\pi}{2y})} - \int_0^\infty e^{-2yt}\, t^{1/2}\, (1+t^2)^{-1/2}\, dt \qquad \text{(A14)}$$

A closed form expression for H(y) is obtained by differentiating the Laplace transform pair 4.3(32) of ref. 7 and then exploiting some relationships between parabolic cylinder functions and Bessel functions, equations 19.3.7 and 19.15.9 of ref. 8. The expression is

$$H(y) = \pi(\tfrac{\pi}{2}y)^{1/2} \left[J_{1/4}(y)\, J_{-3/4}(y) - J_{-1/4}(y)\, J_{3/4}(y) + \sqrt{2}\, J_{1/4}(y)\, J_{3/4}(y)\right] \qquad \text{(A15)}$$

After the a_k are compute then E can be evaluated as a function of b.

$$E = \int_0^\infty t^{-1/2}\, (g-f)\, g\, dt - \int_0^\infty t^{-1/2}\, (g-f)\, f\, dt \qquad \text{(A16)}$$

The first integral in (A16) is zero because of equation (A8) so

$$E = E_0 - \sum_{k=1}^{n} a_k\, H(bp_k/2) \qquad \text{(A17)}$$

where

$$E_0 = \int_0^\infty t^{-1/2}\, [f(t)]^2\, dt \qquad \text{(A18)}$$

This can also be integrated in closed form

$$E_0 = \frac{\pi}{\sqrt{2}} \left[\frac{8\sqrt{(2\pi)}}{(\Gamma(1/4))^2} - 1\right] \approx 1.16741087 \qquad \text{(A19)}$$

The exponent b is obtained by solving dE/db = 0

$$\frac{dE}{db} = -\sum_{k=1}^{n} \left[a'_k\, H(bp_k/2) + a_k\, (p_k/2)\, H'(bp_k/2)\right] \qquad \text{(A20)}$$

where $a'_k = da_k/db$ is obtained by differentiating equation (A11) and solving

$$\sum_{k=1}^{n} c_{\ell k} \frac{da_k}{db} = \frac{1}{2\sqrt{(\pi b)}} H(\tfrac{bp_\ell}{2}) + \frac{p_\ell}{2} \sqrt{(b/\pi)} H'(\tfrac{bp_\ell}{2}) \qquad \text{(A21)}$$

and where

$$H'(y) = -\frac{1}{2y} H(y) + \sqrt{(\pi y/2)} \left\{J^2_{-3/4}(y) - J^2_{-1/4}(y) + J^2_{3/4}(y) - J^2_{1/4}(y) + \sqrt{2}\left[J_{-3/4}(y)\, J_{3/4}(y) + J_{1/4}(y)\, J_{-1/4}(y)\right]\right\} \qquad \text{(A22)}$$

A detailed description of the algorithms used to implement the above equations follows:

I. Since E_0 is independent of n and b it only has to be computed once. The quantity $\Gamma(1/4)$ appearing in equation (A19) was computed by first using equations 6.1.48 of ref. 8 to compute $\Gamma(145+1/4)$ and then using the functional equation for $\Gamma(z)$, equation 6.1.15 of ref. 8, to compute $\Gamma(1/4)$.

II. The number of terms, n is read into the program. The elements of the matrix $C = [c_{\ell k}]$, which are independent of b, are computed from equation (A12). Since C is symmetric only the diagonal and upper triangle of C are stored. The C matrix, which is positive definite, is factored without pivoting

$$C = LDU \qquad \text{(A23)}$$

where L is a unit ($\ell_{kk} = 1$) lower triangular matrix, D is a diagonal matrix, and U is a unit upper triangular matrix. Since L is the transpose of U only D and U are stored. They are overwritten onto C in memory. The algorithm dependent spectral norm condition number of C, $\max |d_{kk}| / \min |d_{kk}|$, is also computed.

III. For each value of b both E(b) and $E'(b) = dE/db$ are computed from equations (A17) and (A20). The a_k and a'_k are obtained by solving equations (A11) and (A20) using the LDU factorization described in step II above. The quantities H(y) and H'(y) for $y = b \cdot p_k$, k=1 to n, are computed from the four Bessel functions $J_{-3/4}$, $J_{-1/4}$, $J_{1/4}$, and $J_{3/4}$. These Bessel functions are computed using the Miller-Abramowitz algorithm described in example 1 of section 9.12 of ref. 7. The example is for integer order and uses equation 9.1.46 for normalization. Since $J_{1/4}$, etc. are of non-integer order equation 9.1.87 of ref. 7 has to be used for normalization. The highest argument gamma function in eq. 9.1.87 is computed as in Step I and the rest of the gamma functions are computed recursively from eq. 6.1.15.

IV. Finding a value of b for which E(b) is a minimum is performed in two steps.

A. A plot of $\log_{10}E$ vs. $\log_{10}b$ is constructed in order to determine the number of minima and their approximate locations.

B. Each approximate minimum is identified by three consequtive b values, b_1, b_2, and b_3, for which the central value of E(b), E_2 is lower than either of the two adjacent values, E_1, and E_3. These three values of b and the associated values of E'(b), E'_1, E'_2, and E'_3 are used as starters for Muller iteration which is used to solve the equation E'(b) = 0.

V. Step IV above determines many values of b for which E(b) is a relative minimum. For each of these values of b the maximum value of $|e(t)|$ was obtained from a plot of e(t) vs t. The plot for which $\max|e(t)|$ was a minimum gave the optimum value of b.

The program to implement the least squares process was originally written for $p_k = k$. For this choice of p_k array the C-matrix ($c_{\ell k} = (\ell + k)^{-1/2}$) is extremely ill conditioned so all calculations were performed in double precison. Double precision is not needed for $p_k = 2^{k/m}$ but was used because the program had already been written.

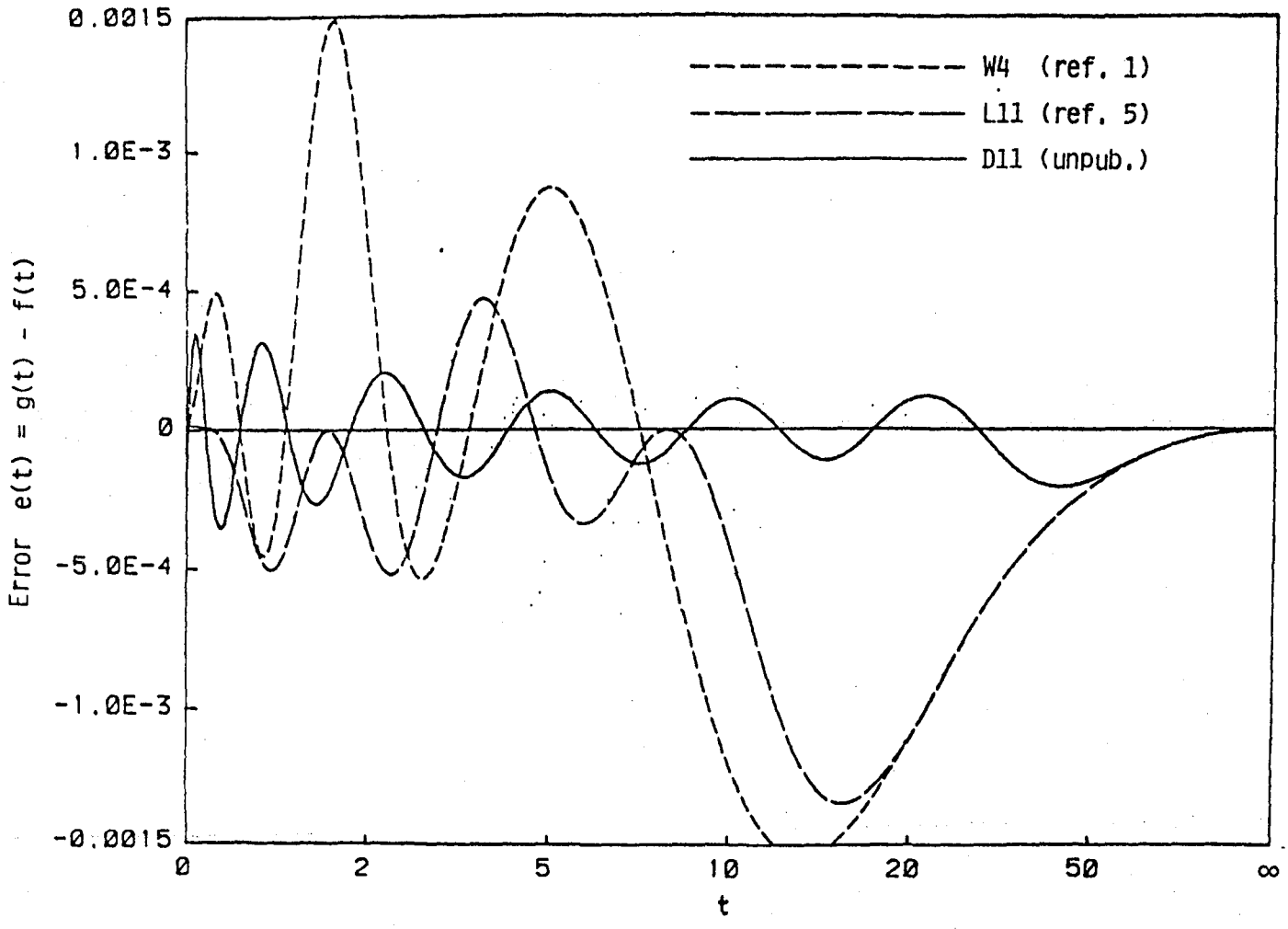

Figure 1. - Comparison of error in the exponential approximations W4, L11, and D11 to the function $1-t/\sqrt{(1+t^2)}$. The nonlinear t-scale on the plot results from mapping the $(0, \infty)$ t-interval into a $(-1,1)$ u-interval by the bilinear transformation $u = (t+10)/(t-10)$.

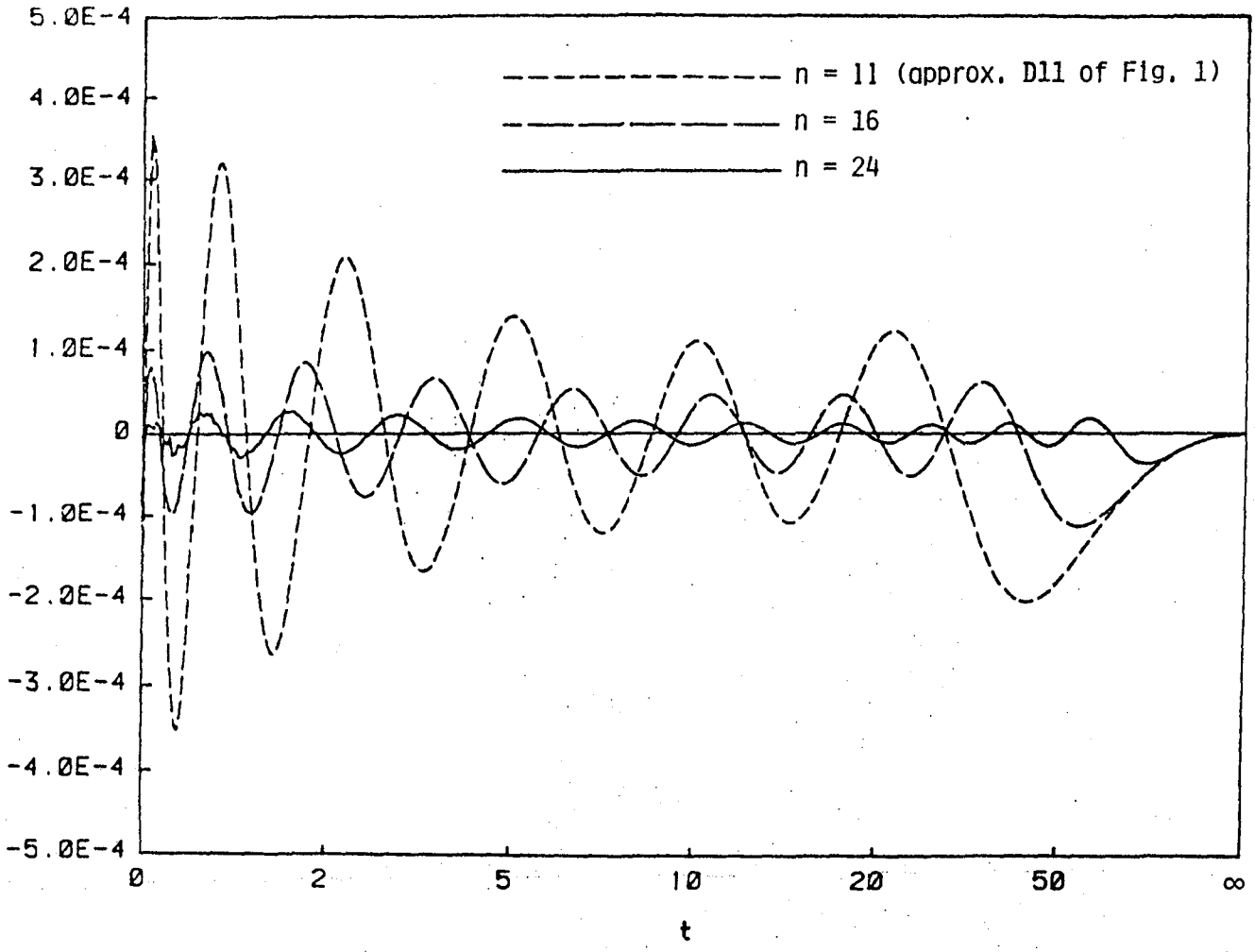

Figure 2. - Errors in least squares approximations for three values of n using Laschka's form for the exponents. Notice that the vertical scale is different from that of figure 1.

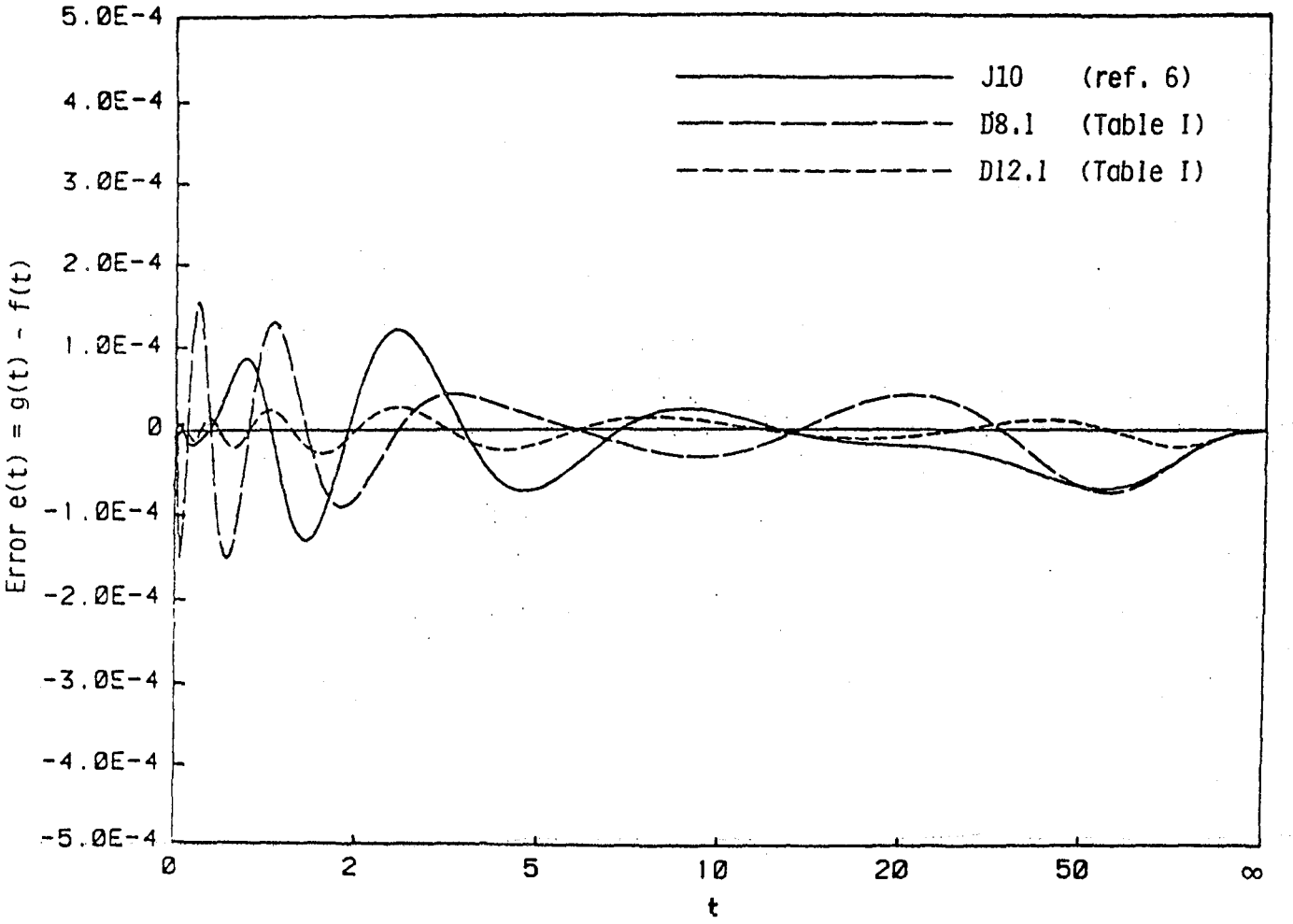

Figure 3. - Errors in least squares approximation using geometric exponent sequences. Even though max |e(t)| is less for J10 than for D8,1 the latter is a better approximation because it has more nodes. (See E(s,r) in table II).

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TM 83281		

4. Title and Subtitle	5. Report Date
AN ACCURATE METHOD FOR EVALUATING THE KERNEL OF THE INTEGRAL EQUATION RELATING LIFT TO DOWNWASH IN UNSTEADY POTENTIAL FLOW	April 1982
	6. Performing Organization Code 2250
7. Author(s) Robert N. Desmarais	8. Performing Organization Report No.
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665	10. Work Unit No. 505-33-53-07
	11. Contract or Grant No.
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546	13. Type of Report and Period Covered Technical Memorandum
	14. Sponsoring Agency Code

15. Supplementary Notes

This paper was presented at the AIAA/ASME/ASCE/AHS 23rd Structures, Structural Dynamics and Materials Conference, May 10-12, 1982, New Orleans, Louisiana, AIAA Paper No. 82-687.

16. Abstract

This paper describes an accurate economical method for generating approximations to the kernel of the integral equation relating unsteady pressure to normalwash in nonplanar flow. The method is capable of generating approximations of arbitrary accuracy. It is based on approximating the algebraic part of the non elementary integrals in the kernel by exponential approximations and then integrating termwise. The exponent spacing in the approximation is a geometric sequence. The coefficients and exponent multiplier of the exponential approximation are computed by least squares so the method is completely automated. Exponential approximates generated in this manner are two orders of magnitude more accurate than the exponential approximation that is currently most often used for this purpose. Coefficients for 8, 12, 24, and 72 term approximations are tabulated in the report. Also, since the method is automated, it can be used to generate approximations to attain any desired trade-off between accuracy and computing cost.

17. Key Words (Suggested by Author(s))	18. Distribution Statement
Flutter Unsteady Aerodynamics	Unclassified - Unlimited Subject Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	14	A01

For sale by the National Technical Information Service, Springfield, Virginia 22161