NASA TECHNICAL NOTE

NASA TN D-5438

C. 1

NASA TN D-5438

# BODIES OF REVOLUTION HAVING MINIMUM TOTAL DRAG IN HYPERSONIC FLOW

*by Waldo I. Oehman and Sylvia A. Wallace*

*Langley Research Center*
*Langley Station, Hampton, Va.*

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • SEPTEMBER 1969

| | | |
|---|---|---|
| 1. Report No.<br>NASA TN D-5438 | 2. Government Accession No. | 3. Recipient's Catalog No. |
| 4. Title and Subtitle<br>BODIES OF REVOLUTION HAVING MINIMUM TOTAL DRAG IN HYPERSONIC FLOW | | 5. Report Date<br>September 1969 |
| | | 6. Performing Organization Code |
| 7. Author(s)<br>By Waldo I. Oehman and Sylvia A. Wallace | | 8. Performing Organization Report No.<br>L-6451 |
| | | 10. Work Unit No.<br>126-13-02-27-23 |
| 9. Performing Organization Name and Address<br>NASA Langley Research Center<br>Hampton, Va. 23365 | | 11. Contract or Grant No. |
| 12. Sponsoring Agency Name and Address<br>National Aeronautics and Space Administration<br>Washington, D.C. 20546 | | 13. Type of Report and Period Covered<br>Technical Note |
| | | 14. Sponsoring Agency Code |
| 15. Supplementary Notes | | |

16. Abstract

The calculus of variations was used to study minimum-drag bodies in hypersonic flow. Newton's formula for pressure drag was assumed, and viscous drag was formulated by use of a variable local skin-friction coefficient. Numerical results are included to assess the effect of the viscous drag for bodies having either given length and base height or given length and volume. The computer program is included for convenience.

| | | | |
|---|---|---|---|
| 17. Key Words Suggested by Author(s)<br>Minimum-drag bodies<br>Hypersonic flow<br>Calculus of variations | | 18. Distribution Statement<br>Unclassified – Unlimited | |
| 19. Security Classif. (of this report)<br>Unclassified | 20. Security Classif. (of this page)<br>Unclassified | 21. No. of Pages<br>51 | 22. Price*<br>$3.00 |

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

# BODIES OF REVOLUTION HAVING MINIMUM TOTAL DRAG IN HYPERSONIC FLOW

By Waldo I. Oehman and Sylvia A. Wallace
Langley Research Center

## SUMMARY

The calculus of variations was used to study minimum drag bodies of revolution in hypersonic flow. Newton's formula for pressure drag was assumed, and viscous drag was formulated by use of a variable skin-friction coefficient. Constraints of either given length and base height or given length and volume were imposed on the bodies.

Fineness ratios to about 9 were investigated. The minimum drag bodies obtained were characterized by flat noses having, at most, a diameter of 3.5 percent of the maximum body diameter. The effect of viscous drag was a reduction of the volume of the bodies for the larger fineness ratios. Practical body shapes, for which the body slope was required to be greater than zero, have an upper limit on the value of fineness ratio. By use of a "slender body" approximation, other investigators have obtained a similar limit on the fineness ratio. For a few cases investigated for which the ratio of volume to body length was held constant rather than the fineness ratio, it was found that viscous drag had almost a negligible effect on the minimum drag body shapes.

## INTRODUCTION

The meridian shapes of bodies of revolution that produce minimum Newtonian pressure drag have been calculated by Eggers, and others. (See ref. 1.) Suddath and Oehman (refs. 2 and 3) calculated the minimum pressure drag shapes for bodies with elliptical cross sections. Several authors have recently included the effect of viscous forces in the problem of determining the shapes of minimum drag bodies in hypersonic flow. In reference 4, Kennet considered slender bodies (that is, bodies for which the local slope is much smaller than unity) having an assumed constant skin-friction drag coefficient. Bryson, in chapter 18 of reference 6, dropped the slender-body approximation, but retained the assumption of a constant skin-friction coefficient. A study by Miele and Cole (refs. 5 and 6) of two-dimensional shapes and slender pointed bodies of revolution included a variable skin-friction coefficient.

The present investigation extends the previous work by considering nonslender bodies, both pointed and flat-nosed, and a variable skin-friction coefficient. In this extension a calculus of variations solution of the problem has been obtained, numerically, on a digital computer. Computed minimum-total-drag body shapes are presented to assess the effect of friction drag and to illustrate the use of the computer program. In addition to the constraint of given length and base height, body shapes were computed for the constraint of given length and volume. Skin-friction coefficients for either laminar or turbulent boundary layers have been used.

## SYMBOLS

$A$ constant in skin-friction-coefficient formula

$C_D$ drag coefficient, $\frac{\text{Drag}}{q_\infty \pi {y_1}^2 L^2}$

$C_{DF}$ friction drag coefficient

$c_f$ local skin-friction coefficient

$D_b$ base drag of body

$D_f$ friction drag

$D_p$ pressure drag

$D_{total}$ sum of $D_b$, $D_f$, and $D_p$

$f,F,g$ integrand functions

$f(x,y,y')$ integrand function in equation (6)

$F(x,y,y',u,\lambda,\beta)$ integrand function in equation (10)

$$F_{uu} = \frac{\partial^2 F}{\partial u^2}$$

$$F_{y'y'} = \frac{\partial^2 F(x,y,y',u,\lambda,\beta)}{\partial y'^2}$$

$$F_{y'} = \frac{\partial F(x,y,y',u,\lambda,\beta)}{\partial y'}$$

$g(y)$ integrand function (see eq. (7))

| | |
|---|---|
| $G$ | value of integral in equation (7) |
| $I$ | functional defined by equation (6) |
| $J$ | functional defined by equation (9) |
| $L$ | length of body |
| $n$ | fineness ratio defined as ratio of length to base diameter of body of revolution |
| $q_\infty$ | free-stream dynamic pressure |
| $u,y'$ | body slope, $\frac{dy}{dx}$ |
| $V$ | volume of body of revolution |
| $x,y$ | body coordinates, nondimensional with respect to $L$, along and perpendicular, respectively, to axis of rotation |
| $\alpha$ | exponent in skin-friction-coefficient formula |
| $\beta$ | Lagrange multiplier |
| $\delta$ | variation consistent with prescribed boundary conditions |
| $\tilde{\delta}$ | variation taken at constant station $x$ |
| $\sigma$ | constant in skin-friction-coefficient formula |
| $\lambda$ | multiplier function |
| $\Phi(y_o)$ | function of body radius at $x = 0$ (see eq. (6)) |

Subscripts:

| | |
|---|---|
| o | value at body nose, $x = 0$ |
| 1 | value at body base, $x = 1$ |
| i | ith value |

Dot over a symbol denotes derivative with respect to $x$. A prime also denotes a derivative with respect to $x$.

## PROBLEM FORMULATION AND SOLUTION

### Statement of the Problem

A body of revolution, at zero angle of attack, is considered to be in a hypersonic flow of air. The total drag of the body may be expressed as the sum of pressure drag, skin-friction drag, and base drag, or

$$D_{total} = D_p + D_f + D_b \tag{1}$$

The pressure drag is assumed to satisfy Newton's law of resistance and is formulated as follows:

$$D_p = 2\pi q_\infty L^2\left(y_o{}^2 + \int_0^1 \frac{2yy'^3\ dx}{1 + y'^2}\right) \tag{2}$$

Formulation of the skin-friction drag is the following integral:

$$D_f = 2\pi q_\infty L^2 \int_0^x yc_f\ dx \tag{3}$$

The local skin-friction coefficient is assumed to satisfy

$$c_f = \frac{A}{\left(\sigma + x\sqrt{1 + y'^2}\right)^\alpha} \tag{4}$$

where the value of the parameter $A$ depends on the physics of the flow conditions, and the value of the exponent $\alpha$ depends on the character of the boundary layer. Usually, $\alpha = 0.2$ is associated with a turbulent boundary layer and $\alpha = 0.5$ is associated with a laminar boundary layer. The quantity in the parentheses represents, approximately, the distance along the body meridian. The constant $\sigma$ is included to avoid the problem associated with an infinite skin-friction coefficient when $x = 0$.

Base drag is assumed to be negligibly small and is set equal to zero. Also, the effect of boundary-layer thickness on the pressure distributions has not been included in equation (2). Furthermore, the boundary layer does not have a transition region.

Combining equations (1), (2), and (3), the total drag is

$$D_{total} = 2\pi q_\infty L^2\left\{y_o{}^2 + \int_0^1 y\left[\frac{2y'^3}{1 + y'^2} + \frac{A}{\left(\sigma + x\sqrt{1 + y'}\right)^\alpha}\right] dx\right\}$$

or, in nondimensional form,

$$\frac{D_{total}}{2\pi q_\infty L^2} = y_o{}^2 + \int_0^1 y\left[\frac{2y'^3}{1 + y'^2} + \frac{A}{\left(\sigma + x\sqrt{1 + y'^2}\right)^\alpha}\right] dx \tag{5}$$

If the function $y(x)$ is known, the drag factor $D_{total}/2\pi q_\infty L^2$ may be evaluated by performing the integration indicated by equation (5). However, the problem of interest is to determine the function $y(x)$ that makes the drag factor a minimum. In the development of equation (5), it has been assumed that the length of the body is known. Therefore, the minimization of the drag factor is to be accomplished for a given length and either a given base height or a given volume.

### Method of Solution

The solution of the problem is obtained by applying the calculus of variations and by using a digital computer to solve the resulting two-point boundary-value problem. A brief development of conditions that the solution must satisfy is presented in functional form.

For convenience, a quantity I is defined by

$$I \triangleq \frac{D_{total}}{2\pi q_\infty L^2} = \Phi\left(y_o\right) + \int_0^1 f(x,y,y')\, dx \tag{6}$$

where

$$\Phi\left(y_o\right) \triangleq y_o{}^2$$

and

$$f(x,y,y') = y\left[\frac{2y'^3}{1 + y'^2} + \frac{A}{\left(\sigma + x\sqrt{1 + y'^2}\right)^\alpha}\right]$$

If the volume of the body is specified, the functional I is to be minimized so that an integral of the form

$$G = \int_0^1 g(y)\, dx \tag{7}$$

has the prescribed value. Furthermore, it will be convenient to substitute u for y' in f(x,y,y') and introduce the differential equation

$$u - y' = 0 \tag{8}$$

as a subsidiary condition. A new functional J which is to be minimized is given by

$$J = \Phi\left(y_0\right) + \int_0^1 \left[f(x,y,u) + \lambda(x)\ (u - y') + \beta g(y)\right] dx \tag{9}$$

or

$$J = \Phi\left(y_0\right) + \int_0^1 F(x,y,y',u,\lambda,\beta)\ dx \tag{10}$$

where $\lambda(x)$ is a variable multiplier and $\beta$ is a constant multiplier. Minimization of the functional J is equivalent to minimization of I subject to the integral constraint (eq. (7)) and the subsidiary condition (eq. (8)).

Calculating the first variation of J and setting it equal to zero leads to

$$\delta J = \left(\frac{\partial\Phi}{\partial y} + \frac{\partial F}{\partial y'}\right)\delta y\Bigg]_{x=0}^{x=1} + \int_0^1 \left[\left(\frac{\partial F}{\partial y} - \frac{d}{dx}\frac{\partial F}{\partial y'}\right)\tilde{\delta} y + \frac{\partial F}{\partial u}\tilde{\delta} u + \frac{\partial F}{\partial \lambda}\tilde{\delta}\lambda\right] dx = 0 \tag{11}$$

where the symbol $\delta$ denotes a variation consistent with prescribed boundary conditions and $\tilde{\delta}$ denotes a variation taken at a constant station x. From familiar arguments concerning the arbitrary variations $\delta y$, $\tilde{\delta} y$, $\tilde{\delta} u$, and $\tilde{\delta}\lambda$, equation (11) leads to the following three necessary conditions:

(1) Satisfaction of Euler equations

(2) Satisfaction of end conditions

(3) Satisfaction of Legendre condition.

Euler equations.- The Euler equations which must be satisfied over the interval $0 \leqq x \leqq 1$ are when the length and base height are given:

$$\left.\begin{aligned} \frac{d\lambda}{dx} &= -\frac{\partial F}{\partial y} = -\frac{\partial f}{\partial y} \\ \frac{dy}{dx} &= u \\ \frac{\partial F}{\partial u} &= \frac{\partial f}{\partial u} + \lambda = 0 \end{aligned}\right\} \tag{12a}$$

and when the length and volume are given:

$$\left.\begin{aligned} &\frac{d\lambda}{dx} = - \frac{\partial F}{\partial y} = - \frac{\partial f}{\partial y} - \beta \frac{\partial g}{\partial y} \\ &\frac{dy}{dx} = u \\ &\frac{\partial F}{\partial u} = \frac{\partial f}{\partial u} + \lambda = 0 \end{aligned}\right\} \tag{12b}$$

It is possible to obtain a first integral of the Euler equations that contains an additional variable multiplier. However, since the integration provides no additional information for these problems, the integral has been omitted.

End conditions.- The initial condition that must be satisfied is

$$\left.\frac{\partial \Phi}{\partial y_o} - \frac{\partial F}{\partial y'}\right]_{x=0} = 0 \tag{13}$$

which is applicable for both problems, and the final condition is

$$\left.F_{y'}\right]_{x=1} = 0 \tag{14}$$

which is applicable only for the problem with given length and volume.

Legendre condition.- The solution $y(x)$ must be obtained so that the Legendre condition is satisfied in the interval $0 \leqq x \leqq 1$; that is,

$$F_{uu} \geqq 0 \tag{15}$$

This set of necessary conditions must be satisfied by the functions $y(x)$, $u(x)$, and $\lambda(x)$ in order to extremize the functional of equation (9). Furthermore, the resulting extremal is called a weak extremal. It has been implied, in this development, that the body slope is greater than zero for all values of $x$.

## Specific Solutions

Given length and base height.- The integrand function for the problem in which the length and base height are given (see eqs. (9) and (10)) is written explicitly as

$$F = y\left[\frac{2u^3}{1 + u^2} + \frac{A}{\left(\sigma + x\sqrt{1 + u^2}\right)^\alpha}\right] + \lambda\left(u - \frac{dy}{dx}\right) \tag{16}$$

The Euler equations are

$$\left.\begin{aligned} \frac{d\lambda}{dx} &= - \frac{2u^3}{1 + u^2} - \frac{A}{\left(\sigma + x\sqrt{1 + u^2}\right)^\alpha} \\ \frac{dy}{dx} &= u \\ y\Bigg[\frac{2u^2\left(3 + u^2\right)}{\left(1 + u^2\right)^2} &- \frac{\alpha Aux}{\sqrt{1 + u^2}\left(\sigma + x\sqrt{1 + u^2}\right)^{\alpha+1}}\Bigg] + \lambda = 0 \end{aligned}\right\} \tag{17}$$

Applying the end condition at $x = 0$ (eq. (13)) gives

$$\lambda_o = -2y_o \tag{18}$$

which is the initial condition for the differential equation for $\lambda$. The value of $y_o$ is not specified and must be chosen so that the integration of equation (18) from $x = 0$ to $x = 1$ yields the given value of $y_1$. Choices of $y_o = 0$ or $y_o > 0$ lead to separate consequences. If $y_o > 0$, then, by using equations (17) and (18), the value of $u_o$ is unity. If $y_o = 0$, the last equation of the system (eqs. (17)) is an identity and does not provide any information for evaluation of $u_o$. However, if the Euler equation of the drag integral (eq. (6)) is evaluated at $x = 0$, an equation in $y_o'$ is obtained. The Euler equation is

$$\frac{d}{dx}\frac{\partial f}{\partial y'} - \frac{\partial f}{\partial y} = 0$$

where the function $f$ is defined for equation (6). Since $f$ is of the form

$$f(x,y,y') = yg(x,y')$$

the Euler equation becomes

$$\frac{d}{dx}\left(y\frac{\partial g}{\partial y'}\right) - g = 0$$

or

$$y\frac{d}{dx}\left(\frac{\partial g}{\partial y'}\right) + y'\frac{\partial g}{\partial y'} - g = 0$$

When $x = 0$, $y_o = 0$ and the first term of the Euler equation is zero. Thus, at $x = 0$,

$$\left(y' \frac{\partial g}{\partial y'} - g\right)\Bigg]_{x=0} = 0$$

Explicitly, the last form of the Euler equation becomes

$$y_o' \frac{2y_o'^2\left(3 + y_o'^2\right)}{\left(1 + y_o'^2\right)^2} - \frac{2y_o'^3\left(1 + y_o'^2\right)}{\left(1 + y_o'^2\right)^2} - \frac{A}{\sigma^\alpha} = 0$$

or more simply,

$$\frac{4y_o'^3}{\left(1 + y_o'^2\right)^2} - \frac{A}{\sigma^\alpha} = 0$$

which, on expansion, becomes a fourth-degree equation in $y_o'$ as follows:

$$y_o'^4 - \frac{4\sigma^\alpha}{A} y_o'^3 + 2y_o'^2 + 1 = 0 \tag{19}$$

Thus, for $y_o = 0$, this form of the Euler equation evaluated at $x = 0$ (eq. (19)) provides four values for the initial body slope $y_o' = u_o$. Application of simple algebraic theorems shows that two of the roots of equation (19) are real and positive and that the other roots are a conjugate complex pair. Only the real roots are significant for the present investigation.

The Legendre condition (inequality (15)) which must hold for all $x$ is

$$F_{uu} = \frac{y}{1 + u^2}\left(\frac{4u\left(3 - u^2\right)}{\left(1 + u^2\right)^2} + \frac{\alpha A}{\left(\sigma + x\sqrt{1 + u^2}\right)^{2+\alpha}}\left\{x^2\left[\left(1 + \alpha\right)u^2 - 1\right] - \frac{\sigma x}{\sqrt{1 + u^2}}\right\}\right) \geqq 0 \tag{20}$$

The conditions of the problem require that $y(x) > 0$ and that $u(x) > 0$ for $0 < x \leqq 1$. Consequently, inequality (20) may be solved numerically to obtain $u(x)$ for $0 < x < 1$. When $x = 0$, $y_o$ must be greater than or equal to zero $\left(y_o \geqq 0\right)$ and $u_o$ must be greater than zero $\left(u_o > 0\right)$. Obviously, when $y_o = 0$, $F_{uu} = 0$ and any positive value of $u_o$ satisfies the Legendre condition; and, when $y_o > 0$ any $0 < u_o \leqq \sqrt{3}$ satisfies the Legendre condition. A plot of $u$ as a function of $x$ is presented in figure 1 to illustrate the permissible values of $u$ that satisfy the Legendre condition. Thus, any body shape, with $y(x) > 0$, generated from the other necessary conditions will minimize the drag integral only if the body slope $u$ is on or between the boundary curves $\left(F_{uu} = 0\right)$

of figure 1. If $y_0 = 0$, the slope $u_0$ may take on any positive value, but for $0 < x \leqq 1$, the slope $u$ must be on or between the boundary curves.

Given length and volume.- For the problem in which the length and volume are given, the volume is the integral of equation (7) with $g(y) = y^2$ or

$$G = \frac{V}{\pi L^3} = \int_0^1 y^2 \, dx \tag{21}$$

The explicit integrand function of equation (10) is

$$F = y\left[\frac{2u^3}{1 + u^2} + \frac{A}{\left(\sigma + x\sqrt{1 + u^2}\right)^{\alpha}}\right] + \lambda\left(u - \frac{dy}{dx}\right) + \beta y^2 \tag{22}$$

The Euler equations are:

$$\left.\begin{aligned} &\frac{d\lambda}{dx} = \frac{-2u^3}{1 + u^2} - \frac{A}{\left(\sigma + x\sqrt{1 + u^2}\right)^{\alpha}} - 2\beta y \\ &\frac{dy}{dx} = u \\ &y\left[\frac{2u^2(3 + u^2)}{\left(1 + u^2\right)^2} - \frac{\alpha A u x}{\sqrt{1 + u^2}\left(\sigma + x\sqrt{1 + u^2}\right)^{\alpha+1}}\right] + \lambda = 0 \end{aligned}\right\} \tag{23}$$

The end condition at $x = 0$ is the same as equation (18), that is, $\lambda_0 = -2y_0$; and the end condition at $x = 1$ (see eq. (14)) requires that

$$\lambda_1 = 0 \tag{24}$$

Since $y_0$ and $\beta$ are not specified, they must be chosen so that $G$ has the specified value and $\lambda_1 = 0$ when equations (23) are integrated from $x = 0$ to $x = 1$. In order for $\lambda$ to go to zero at $x = 1$, $\beta$ must be a negative number. The discussion of the Legendre condition in the preceding section also applies to this problem. As before, $u_0 = 1$ for $y_0 > 0$ and $u_0 > 0$ for $y_0 = 0$. At $x = 1$, however, the last equation of the system (eqs. (23)) may be solved for $u_1$. That is, $u_1$ must be a solution of

$$y_1 u_1\left(\frac{2u_1\left(3 + u_1^2\right)}{\left(1 + u_1^2\right)^2} - \frac{\alpha A}{\sqrt{1 + u_1^2}\left(\sigma + \sqrt{1 + u_1^2}\right)^{\alpha+1}}\right) = 0 \tag{25}$$

The solution $u_1 = 0$ is not admissible since the Legendre condition requires $u_1 > 0$. Consequently, the quantity in parentheses in equation (25) must be zero; and further analysis shows that $u_1$ must be near zero. Furthermore, the computed value of $u_1$ must be greater than the value on the lower Legendre boundary at $x = 1$ in figure 1 for an acceptable solution of the problem.

## RESULTS AND COMPUTATIONS

The preceding development of the necessary conditions from the first variation of the drag integral has led to the problem of solving sets of ordinary, first-order differential equations (Euler eqs. (17) and (23)). Some, but not all, of the initial and final conditions are known. Therefore, a two-point boundary-value problem must be solved. Usually, an iterative procedure with some sort of convergence is applied to obtain specific solutions. The approach taken in the present investigation has been to consider the unknown initial conditions $y_o$ or $u_o$ and $\beta$ as parameters and to obtain solutions of the Euler equations for particular values of these parameters. In addition to initial conditions, the constants A, $\alpha$, and $\sigma$ are parameters that reflect the flow conditions and a solution (that is, a body shape) may be obtained for each set of parameters. Because the numerical results presented in the following sections were obtained for a few selected sets of these parameters, the digital computer program is presented, for convenience, in the appendix. The computer time is about 3 minutes for each set of initial conditions.

### Given Length and Base Height

A family of flat-nosed minimum-drag bodies has been computed for the parameter sets (A, $\alpha$, $\sigma$) of (0, -, -), (0.0002, 0.5, 0.01), (0.0005, 0.2, 0.01), and (0.0009, 0.2, 0.01), and for initial body ordinates $y_o$ ranging from $10^{-2}$ to $10^{-30}$. The initial body slope was, of course, $u_o = 1$ for each body of the family. A plot of $y_o$ as a function of fineness ratio (that is, the ratio of length to base diameter) is presented in figure 2. The minimum pressure drag bodies (A = 0) are characterized by the continued increase of fineness ratio as $y_o$ tends to zero. However, the fineness ratio of minimum drag bodies (A > 0) approaches an upper limit as $y_o$ tends to zero. The upper limit of the fineness ratio for each set (A, $\alpha$, $\sigma$) is the value of n indicated for $y_o = 10^{-7}$ in figure 2. These values are also the same as were obtained for $y_o = 10^{-30}$. The main result for flat-nose bodies that produce minimum drag is that the choice of fineness ratio for given flow conditions is limited.

For pointed bodies $(y_o = 0)$, solution of equation (19) yields initial values of the body slope $y_o'$. The real values, computed for the present investigation, are

| A, $\alpha$, $\sigma$ | $y_o' = u_o$ |
|---|---|
| 0.0002, 0.5, 0.01 | 0.0797058575; 1999.99900 |
| .0005, 0.2, .01 | .0681782789; 3184.85674 |
| .0009, 0.2, .01 | .0830586184; 1769.36407 |

The values of $y_o'$ depend on the set (A, $\alpha$, $\sigma$) which embodies the flight conditions. Furthermore, since $y_o(y_o = 0)$, $y_o'(y_o' = u_o)$, and $\lambda_o(\lambda_o = 0)$ are known, the minimum-drag body shape is completely determined by integrating equations (17) from $x = 0$ to $x = 1$. Consequently, for a given length, the base height cannot be specified arbitrarily but is dependent on the flight conditions. For each set (A, $\alpha$, $\sigma$), there are two values of $y_o'$ each of which leads to a solution (body shape) of the problem. Both body shapes are almost identical for $x > 10^{-3}$. However, the total drag of the body having the smaller initial slope is slightly less than the total drag of the body having the larger initial slope. Therefore, when solving equation (19) for $y_o'$, only the smaller value yields the correct minimum-drag body slope. It should be emphasized that the restriction on the choice of fineness ratio noted here and in the preceding paragraph is a consequence of requiring that $u(x) > 0$. However, the body shapes for larger fineness ratios obtainable when $u(x) = 0$ for some $0 < x < 1$ are not compatible with the mathematical model of the drag given herein. The restriction on the choice of fineness ratio is the same as may be derived from the results given in reference 5 for "slender" bodies with a subcritical value of the friction parameters. However, the minimum-drag bodies of reference 5 are blunt bodies that have an infinite slope at the nose.

The body shapes presented in figure 3 for fineness ratios of 2 and 5 illustrate the effect of the friction drag. In figure 3(a), the body shapes ($n = 2$) are almost identical for the three sets of (A, $\alpha$, $\sigma$) that is, (0, -, -), (0.0005, 0.2, 0.01), and (0.0009, 0.2, 0.01). Thus, the body shaping is dominated by the pressure drag. However, for $n = 5$, increasing the local skin-friction coefficient (that is, increasing A) causes a decrease of the local body radius. (See fig. 3(b).) Thus, the viscous term in the drag integral leads to a decrease of the volume of a minimum drag body with a given fineness ratio. The curves in figure 4 show that the friction drag coefficient becomes larger than the pressure drag coefficient as the fineness ratio increases. Thus, the importance of including viscous drag in the problem formulation is emphasized.

## Given Length and Volume

Sets (A, $\alpha$, $\sigma$) of (0.0005, 0.2, 0.01) and (0.0009, 0.2, 0.01) were combined with values of $y_o$ ranging from $10^{-2}$ to $10^{-7}$ (flat-nosed bodies) to compute minimum-drag bodies having given length and volume. The calculations were performed by iterating to obtain a value of $\beta$ for each set of (A, $\alpha$, $\sigma$) and $y_o$ that would force $\lambda$ to go to zero when

$x = 1$. Figure 5 presents $y_o$ and the corresponding multiplier $\beta$ for the volume ratio $V/L^3$. Two points for $A = 0$ are also presented in figure 5(a). (These points for flat-nose bodies are taken from ref. 1.) This figure shows that, for a specific volume ratio, a slightly different value of $y_o$ is required for each set $(A, \alpha, \sigma)$. Figure 6 shows that the resulting fineness ratio is not appreciably affected by changes of A. Consequently, the effect of viscous drag on the body shape is almost negligible although the viscous drag coefficient is as much as 80 percent of the total drag coefficient. (See fig. 7.)

One effect of viscous drag was the requirement imposed by the Legendre condition that the body slope at the base must be greater than zero. (Minimum pressure drag bodies of refs. 2 and 3 with given length and volume have a zero slope at the base.) However, the required slope is very small compared with unity. Figure 8 presents some minimum-drag body shapes for given length and volume for several values of the volume ratio.

## CONCLUDING REMARKS

An analytical investigation was made to determine the meridian shapes of minimum-drag bodies having either given length and base height or given length and volume. The flow was assumed to be hypersonic and Newton's formula for pressure drag to be applicable. Viscous drag was formulated by using a skin-friction coefficient that varied with distance along the body meridian and the boundary-layer flow was considered to be either laminar or turbulent. A solution was obtained by application of the calculus of variations, and equations obtained from the necessary conditions were programed for a digital computer.

A limited parametric analysis has been made to assess the effect of viscous drag on minimum-drag body shapes and to illustrate the use of the computer program. Minimum-drag body shapes, with fineness ratios to about 9, were characterized by flat noses having, at most, a diameter of 3.5 percent of the maximum body diameter. The effect of viscous drag was a reduction of the volume of these bodies for the larger fineness ratios. Practical body shapes, for which the body slope was required to be greater than zero, have an upper limit on the value of fineness ratio. A similar limit on the fineness ratio has been obtained by other investigators who have used a "slender body" approximation. The main conclusion for flat-nose minimum-drag bodies having given length and volume is that the viscous drag has almost a negligible effect on the body shapes.

Langley Research Center,
National Aeronautics and Space Administration,
Langley Station, Hampton, Va., June 17, 1969,
126-13-02-27-23.

# APPENDIX

## DESCRIPTION OF PROGRAM

The computer program was written in the FORTRAN IV language under SCOPE Version 3.0 for Control Data Corporation's Series 6000 computers at the Langley Research Center. A brief description of the program, as well as a flow diagram and actual listing of the program is included in this appendix. The output from an example problem is also given.

### Contents of the Program

The program numerically integrates the Euler equations (eqs. (23)) which must be satisfied over the interval $0 \leqq x \leqq 1$. Subroutines FALG, INT1A, and ITR2 are used for the solution of the quartic (eq. (19)) for the two real values of $u_o$, for the integration of the differential equations, and for the implicit solution of the $u$ (eqs. (17) or (23)), respectively. The program determines the drag given in equation (5) using the trapezoidal rule to compute the separate integrals for the pressure drag and friction drag coefficients.

### Subprograms

The program makes use of three library subroutines, FALG, INT1A, and ITR2, from the Langley Research Center. A complete listing of these subroutines is included in this paper.

FALG is a subroutine which calculates the $n$ roots of a polynomial of degree $n$, where the coefficients may be either real or complex. In this program it is used to obtain the $u_o$ only if $y_o = 0$. Error returns are:

IERRF = 0; normal return

IERRF = 1; the leading coefficient is zero

IERRF = 2; one of the roots failed to converge in the initial iteration cycle

IERRF = 3; one of the roots failed to converge in the improvement iteration cycle.

INT1A is a closed subroutine for the solution of a set of simultaneous differential equations. It is a variable-interval-size routine and uses the fourth-order Runge-Kutta formula in conjunction with Richardson's extrapolation to the limit theory. Error returns are:

IERR = 1; normal return

IERR = 2; ELT block is not monotonic in the direction of integration

IERR = 3; variables have failed to meet the local truncation error requirements nine consecutive times

IERR = 4; variables have failed to meet the local truncation error requirements at least nine times over the last three intervals. An acceptable answer has been reached, however, and is in the VAR array.

ITR2 finds a value for $x$ within a given epsilon of relative error in a given interval for a given $F(x) = 0$. Error returns are:

ICODE = 0; normal return

ICODE = 1; maximum iterations (= 150) are exceeded

ICODE = 2; DELTX (the scanning interval) = 0, or negative

ICODE = 3; a root cannot be found within the given bounds, ALLU and BULU

ICODE = 4; ALLU > BULU

CHSUB and DERSUB are subroutines required by INT1A

FOFX is a function called by ITR2.

## Options

An option is available for including or omitting $\beta$ from the $\dot{\lambda}$ equation. A value for $\beta$ can be read in and will be included in equations (23), or it need not be read if it is not to be included in the $\dot{\lambda}$ equation. The variable controlling this option is listed under "Input."

## Input

The NAMELIST statement is used to put data in. NAMELIST names and the variable names contained in each with their explanations are listed in the following table:

| NAMELIST name | Variable names | Explanations of variables |
|---|---|---|
| NAM1 | A | A |
| | SIGMA | $\sigma$ |
| | ALPHA | $\alpha$ |
| | PRMIN | The absolute value of an increment of the independent variable which is the frequency of printing results. (See "Output.") |

# APPENDIX

| NAMELIST name | Variable names | Explanations of variables |
|---|---|---|
|  | CII, CIMAX | The initial computing interval and maximum desired computing interval, respectively, as required by INT1A. |
|  | ALLUI, BULUI, DELTX | The lower bound or initial guess for $u$, the upper bound or final guess for $u$, and the size of the scanning interval, respectively, as required by ITR2 in solving for $u$. |
|  | EP1, EP2 | Relative and absolute error criterion, respectively, used in ITR2 to stop the iteration when either of these convergence criteria are satisfied:<br>1. If $\lvert u_i \rvert > \mathrm{EP1}$,<br>$\left\lvert \frac{u_i - u_{i-1}}{u_i} \right\rvert \leqq \mathrm{EP1}$<br>2. If $u_i \leqq \mathrm{EP1}$,<br>$\lvert u_i - u_{i-1} \rvert \leqq \mathrm{EP2}$<br>Values of $\mathrm{EP1} = \mathrm{EP2} = 0.1 \times 10^{-6}$ have given satisfactory results. |
|  | NT | The number of values in the ELT block described below. |
|  | IOPBETA | Equals 0, $\beta$ is not read and is not included in the $\dot{\lambda}$ equation.<br>Equals 2, $\beta$ is read and is included in the $\dot{\lambda}$ equation. |
| NAM2 | VAR0 | A one-dimensional array containing the initial values of the independent variable followed by the three dependent variables, $x$, $y$, and $\lambda$, in that order. |

| NAMELIST name | Variable names | Explanations of variables |
|---|---|---|
| | ELE1, ELE2 | One-dimensional arrays containing the upper bounds of relative truncation error and "relative zeros," respectively, for the dependent variables as required by INT1A. If the error for any variable exceeds its respective ELE1 value, the computing interval is halved and the integration restarted at the beginning of the interval. Under certain criteria the interval is doubled. If the absolute value of any of the variables is less than its respective ELE2 value, the relative error criteria for that variable will not be applied. Satisfactory results have been obtained using ELE1 values = $10^{-7}$ and ELE2 values = $10^{-8}$. |
| | ELT | One-dimensional array of NT values, monotonic in the direction of integration, at which the user specifically desires control returned to his program from INT1A. (See "Output.") |
| NAM3 | BETA | $\beta$, read only if IOPBETA = 2. |

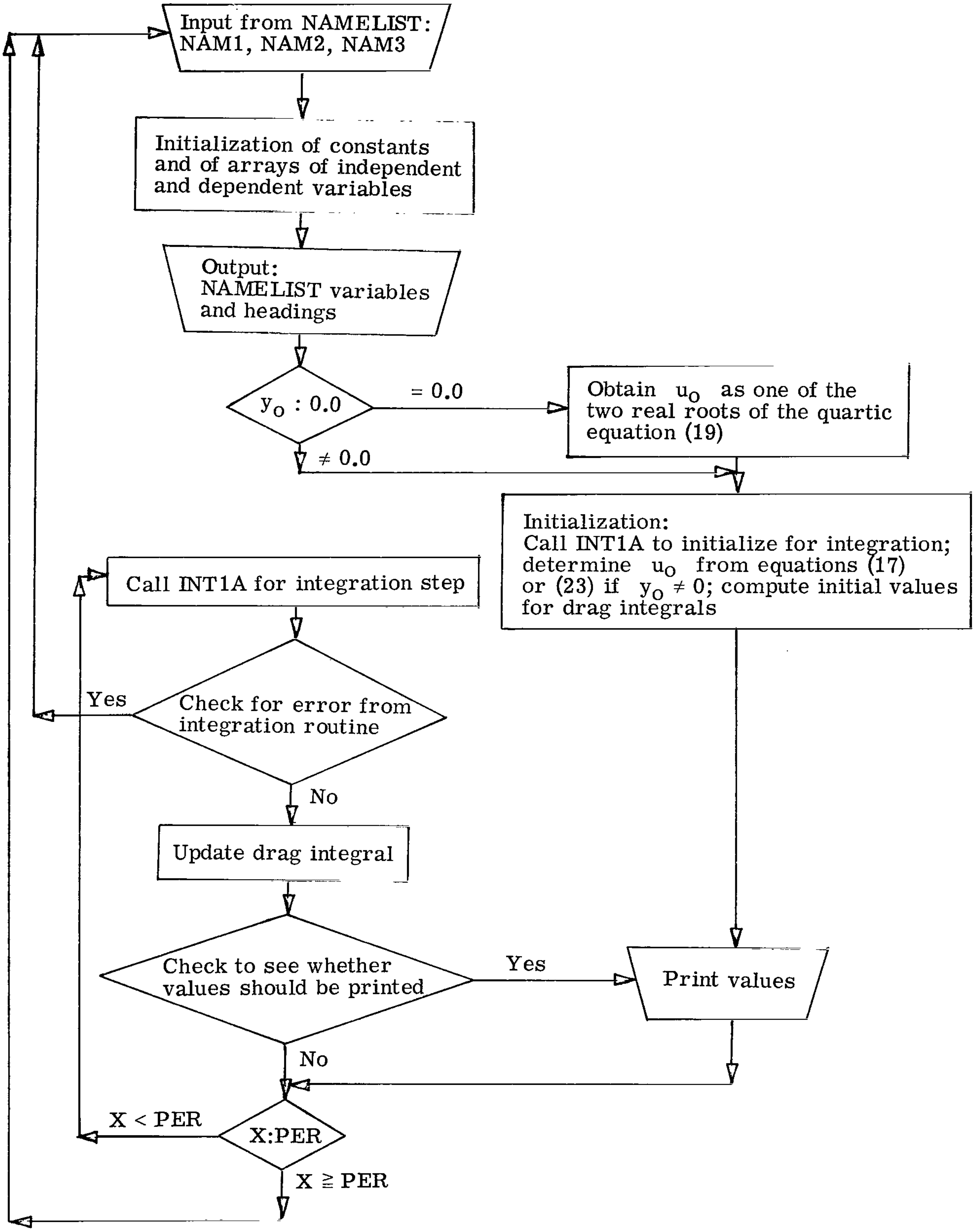
## Output

The frequency of printing results is determined by the input quantity PRMIN. The initial values are printed and the results will be printed again when the independent value is first greater than PRMIN and thereafter when the independent value has been updated by at least as much as PRMIN. The final value in the ELT array should be the final value of the independent variable (equals 1.0). INT1A iterates to obtain results at the specific values listed in ELT. If results at values of x other than 1.0 are desired, the program can be easily modified to have them printed. Values of the drag coefficients are printed at the end of each case. The total drag coefficient is given, as well as the pressure and friction drag coefficients.

## Computational Flow Diagram

A concise computational flow diagram is included here to show the steps in computing. Details can be readily obtained from the listing.

# APPENDIX


Input from NAMELIST:
NAM1, NAM2, NAM3
Initialization of constants and of arrays of independent and dependent variables
Output:
NAMELIST variables and headings
$y_o$ : 0.0
= 0.0
Obtain $u_o$ as one of the two real roots of the quartic equation (19)
≠ 0.0
Initialization:
Call INT1A to initialize for integration; determine $u_o$ from equations (17) or (23) if $y_o \neq 0$; compute initial values for drag integrals
Call INT1A for integration step
Yes
Check for error from integration routine
No
Update drag integral
Check to see whether values should be printed
Yes
Print values
No
X < PER
X:PER
X ≧ PER

# APPENDIX

## Listing of FORTRAN Program

The FORTRAN program, including FALG, INT1A, and ITR2 (Langley Research Center library subroutines) is as follows:

```
      PROGRAM BODREV(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C     PROGRAM TO COMPUTE BODIES OF REVOLUTION HAVING MINIMUM VISCOUS
C          PLUS (NEWTONIAN) PRESSURE DRAG IN HYPERSONIC FLOW.
C     VARIABLES NEEDED IN INTEGRATION ARE STORED IN THE VAR ARRAY
C          (TENTATIVE ANSWERS IN CORRESPONDING POSITIONS IN THE CUVAR
C          ARRAY)
C          VAR(1)     INDEPENDENT VARIABLE
C          VAR(2)     X
C          VAR(3)     Y
C          VAR(4)     LAMBDA
C
C     Y PRIME IS REFFERRED TO AS U IN THIS PROGRAM
C
C     OPTIONS
C     IF IOPBETA = 0,  BETA IS NOT READ AND IS NOT INCLUDED IN THE
C          LAMBDA DOT EQUATION.
C     IF IOPBETA = 2, BETA IS READ AND IS INCLUDED IN THE LAMBDA DOT
C          EQUATION.
C
      COMPLEX ROOTS,TEMP
      DIMENSION ELE1(3),ELE2(3),ELT(10),ERRVAL(3),COEFFS(10),ROOTS(4),
     1   TEMP(10)
      EXTERNAL DERSUB,CHSUB
      COMMON /BLK1/CUVAR(4),VAR(4),DER(4),VARO(4),A,SIGMA,ALPHA,U,ALLU,
     1 BULU,DELTX,FP1,FP2,ICODE,II,CI,BETA,IOPBETA,IOPUO,UO
      NAMELIST /NAM1/A,SIGMA,ALPHA,PRMIN,CII,CIMAX,ALLUI,BULUI,DELTX,
     1                  FP1,FP2,NT,IOPBETA
     2            /NAM2/VARO,ELE1,ELE2,ELT
     3            /NAM3/BETA
10    READ (5,NAM1)
      READ (5,NAM2)
C     READ QUANTITIES IN NAM3 ONLY IF THEY ARE REQUIRED FOR THE RUN.
C          SEE IOPUO AND IOPBETA ABOVE.
      READ (5,NAM3)
      NCASE=0
      IF(VARO(3).EQ.0.0)NCASE=1
11    CI=CII
      PER=0.99999999
      N=3
      ITEXT=0
      SPEC=0.0
      II=0
      NPLUS1=N+1
      DO 20 I=1,NPLUS1
20    VAR(I)=VARO(I)
C
      ALLU=ALLUI
      BULU=BULUI
      WRITE (6,NAM1)
      WRITE (6,NAM2)
      WRITE (6,NAM3)
      WRITE (6,70)
      WRITE (6,80) VAR(3),A,SIGMA,ALPHA
      IF(VARO(3).NE.0.0)WRITE(6,90)
      IF(VARO(3).EQ.0.0)WRITE(6,100)
      IF (IOPBETA.EQ.0) WRITE (6,120)
      IF (IOPBETA.EQ.2) WRITE (6,130) BETA
C
C     IF Y IS INITIALLY = 0, THE INITIAL U WILL BE OBTAINED FROM THE
C          QUARTIC IN U (IOPUO=2).  THERE WILL BE TWO REAL ROOTS, U01
C          AND U02.  COMPUTATIONS WILL BE DONE FOR BOTH CASES.  IF Y IS
C          NOT = 0 INITIALLY, THE INITIAL U IS OBTAINED FROM THE
C          IMPLICIT EULER EQUATION (IOPUO = 0).
      IF(VARO(3).NE.0.0)IOPUO=0
      IF(VARO(3).NE.0.0)GO TO 22
      IF(NCASE.NE.2)GO TO 26
      U0=U02
      GO TO 22
26    CONTINUE
      IOPUO=2
      NDEG=4
      ICOEFF=0
      COEFFS(1)=1.0
      COEFFS(2)=-4.0*(SIGMA**ALPHA)/A
      COEFFS(3)=2.0
      COEFFS(4)=0.0
      COEFFS(5)=1.0
      CALL FALG(COEFFS,NDEG,ICOEFF,ROOTS,TEMP,IERRF)
      IF(IERRF.EQ.0)GO TO 23
      WRITE(6,160)IERRF
      GO TO 10
23    CONTINUE
      IRR=1
      DO 24 I=1,4
      RI=AIMAG(ROOTS(I))
      IF(RI.NE.0.0)GO TO 24
      IF(IRR.EQ.2)GO TO 25
      U01=REAL(ROOTS(I))
      IRR=2
      GO TO 24
25    U02=REAL(ROOTS(I))
24    CONTINUE
      U0=U01
22    CONTINUE
C     INITIALIZE IN SUBROUTINE INT1A
      CALL INT1A (II,N,NT,CI,SPEC,CIMAX,IERR,VAR,CUVAR,DER,ELE1,ELE2,ELT
     1,ERRVAL,DERSUB,CHSUB,ITEXT)
C     COMPUTATION FOR DRAG INTEGRAL  (INITIALIZATION)
C     DRAG1 CONTAINS THE YO**2 TERM PLUS THE FIRST PART OF THE INTEGRAL
C     DRAG2 CONTAINS ONLY THE A TERM OF THE INTEGRAL
      FD1M1=VAR(3)*(2.0*U**3/(1.0+U**2))
      FD2M1=VAR(3)*(A/((SIGMA+VAR(2)*SQRT(1.0+U**2))**ALPHA))
      DRAG1=VAR(3)**2
      DRAG2=0.0
      DRAG=DRAG1+DRAG2
      STORET=VAR(1)
      WRITE (6,140)
      GO TO 50
C     START INTEGRATION
30    CALL INT1A (II,N,NT,CI,SPEC,CIMAX,IERR,VAR,CUVAR,DER,ELE1,ELE2,ELT
     1,ERRVAL,DERSUB,CHSUB,ITEXT)
      IF ((IERR.EQ.1).OR.(IERR.EQ.4)) GO TO 40
```

# APPENDIX

```
      WRITE (6,110) IERR
      GO TO 10
40    CONTINUE
      IF (II.EQ.2) GO TO 30
C     COMPUTATION FOR DRAG INTEGRAL
      FD1=VAR(3)*(2.0*U**3/(1.0+U**2))
      FD2=VAR(3)*(A/((SIGMA+VAR(2)*SQRT(1.0+U**2))**ALPHA))
      FDA1=(FD1M1+FD1)/2.0
      FDA2=(FD2M1+FD2)/2.0
      FDINT1=FDA1*(VAR(1)-STORET)
      FDINT2=FDA2*(VAR(1)-STORET)
      DRAG1=DRAG1+FDINT1
      DRAG2=DRAG2+FDINT2
      DRAG=DRAG1+DRAG2
      FD1M1=FD1
      FD2M1=FD2
      STORET=VAR(1)
C     TEST TO SEE IF VALUES SHOULD BE PRINTED
      IF (VAR(1).GE.PER) GO TO 50
      PREREQ=VAR(1)-PRVAL
      IF (PREREQ.GT.PRMIN) GO TO 50
      IF (VAR(1).LT.PER) GO TO 30
50    CONTINUE
      PRVAL=VAR(1)
      WRITE (6,60) VAR(2),VAR(3),U
      IF (VAR(1).LT.PER) GO TO 30
      CONVD=2.0/(VAR(3)**2)
      CDT=DRAG*CONVD
      CDP=DRAG1*CONVD
      CDF=DRAG2*CONVD
      WRITE (6,150) CDT,CDP,CDF
      IF(IOPU0.EQ.0)GO TO 10
      IF(NCASE.EQ.2)GO TO 10
      NCASE=2
      GO TO 11
C
60    FORMAT (4E18.8)
70    FORMAT (1H11X43HBODIES OF REVOLUTION HAVING MINIMUM VISCOUS1X44HPL
     1US (NEWTONIAN) PRESSURE DRAG IN HYPERSONIC1X4HFLOW)
80    FORMAT (///10X5HINPUT16X2HY017X1HA13X5HSIGMA13X5HALPHA/E36.8,3E18.
     18//)
90    FORMAT (18X*INITIAL U OBTAINED FROM EULER EQUATION*)
100   FORMAT (18X*INITIAL U OBTAINED FROM QUARTIC EQUATION*)
110   FORMAT (1X32HERROR RETURN FROM INT1.   IERR = I4//)
120   FORMAT (18X40HBETA NOT INCLUDED IN LAMBDA DOT EQUATION)
130   FORMAT (18X46HBETA INCLUDED IN LAMBDA DOT EQUATION.    BETA =E16.8)
140   FORMAT (//14X1HX17X1HY11X7HY PRIME/)
150   FORMAT (//6X16HTOTAL DRAG COEFF7X15HPRES DRAG COEFF3X19HFRICTION D
     1RAG COEFF/3E22.8)
160   FORMAT (1X*ERROR RETURN FROM FALG.   IERRF =*,I4)
      END
```

# APPENDIX

```
      SUBROUTINE CHSUB
      RETURN
      END
```

```
      SUBROUTINE DERSUB
      EXTERNAL FOFX
      COMMON /BLK1/CUVAR(4),VAR(4),DER(4),VARO(4),A,SIGMA,ALPHA,U,ALLU,
     1 BULU,DELTX,EP1,EP2,ICODE,II,CI,BETA,IOPBETA,IOPUO,UO
      IF ((IOPUO.NE.2).OR.(II.NE.0)) GO TO 10
      U=UO
      GO TO 30
10    CALL ITR2 (U,ALLU,BULU,DELTX,FOFX,EP1,EP2,150,ICODE)
      IF (ICODE.EQ.0) GO TO 20
      FUNU=CUVAR(3)*(2.0*U**2*(3.0+U**2)/((1.0+U**2)**2)-ALPHA*A*CUVAR(2
     1)*U/((SQRT(1.0+U**2))*((SIGMA+CUVAR(2)*SQRT(1.0+U**2))**(1.0+ALPHA
     2))))+CUVAR(4)
      IF (FUNU.LT.0.1E-06) GO TO 20
      WRITE (6,40) ICODE,U,FUNU
      STOP
20    CONTINUE
30    CONTINUE
      DER(1)=0.0
      DER(2)=1.0
      DER(3)=U
      DER(4)=-2.0*U**3/(1.0+U**2)-A/((SIGMA+CUVAR(2)*SQRT(1.0+U**2))**AL
     1PHA)
      IF (IOPBETA.EQ.2) DER(4)=DER(4)-2.0*BETA*CUVAR(3)
      RETURN
C
40    FORMAT (1X33HERROR RETURN FROM ITR2.   ICODE = I4,3X4HU = F16.8,3X7
     1HF(U) = F16.8//)
      END
```

```
      FUNCTION FOFX(X)
      COMMON /BLK1/CUVAR(4),VAR(4),DER(4),VARO(4),A,SIGMA,ALPHA,U,ALLU,
     1 BULU,DELTX,EP1,EP2,ICODE,II,CI,BETA,IOPBETA,IOPUO,UO
      FOFX=2.0*X**2*(3.0+X**2)/((1.0+X**2)**2)-(ALPHA*A*CUVAR(2)*X)/((SQ
     1RT(1.0+X**2))*((SIGMA+CUVAR(2)*SQRT(1.0+X**2))**(1.0+ALPHA)))+CUVA
     2R(4)/CUVAR(3)
      RETURN
      END
```

# APPENDIX

```
      SUBROUTINE  FALG(COEFFS,N,I,ROOT  ,TEMP,IERR)
********* DOCUMENT DATE 08-01-68   SUBROUTINE REVISED 08-01-68 *********
      DIMENSION COEFFS(1),TEMP(1),ROOT(1)
      DIMENSION XIDIFF(2),RDIFF(2),APPROX(3)
      COMPLEX F,FPR,APPROX,TEMP  ,ROOT
      COMPLEX TEMPM
      COMPLEX RELTST
      NSAVE=N
      IERR=0
      IB=0
      ICLEAN=2*N+2
C           CLEAR WORKING AREA
      DO 7771 LLL=1,ICLEAN
 7771 TEMP(LLL)=0.0
C           CLEAR ROOT STORAGE
      DO 7772 LLL=1,N
 7772 ROOT(LLL)=0.0
C           CONSTANTS TO TEST
C           CONVERGENCE
      CONST=.1E-6
C           OVERFLOW
      OVCON=1.E150
C           MAGNITUDE OF ROOTS
      RCONST=1.E-21
C           JONJON=0,FIRST ITERATION
      JONJON=0
C           CHECK CONSTANT TERM FOR ZERO
      JJJ=1
      NCO=N+1
  802 IF(I.NE.1)GO TO 800
C           COMPLEX COEFFICIENTS
      NCO=2*NCO
      IF(COEFFS(NCO-1).NE.0.)GO TO 101
C           HERE IF REAL COEFFICIENTS
  800 IF(COEFFS(NCO).NE.0.)GO TO 101
C           ROOT=ZERO
  801 ROOT(JJJ)=0.
      NCO=NCO-1
      JJJ=JJJ+1
C           REDUCE DEGREE AND IF 1,STORE ROOT AND EXIT
      N=N-1
      IF(N.NE.1)GO TO 802
      ROOT(JJJ)=0.
      GO TO 1006
C
C           ENTRY  FIRST AND SECOND ITERATIONS
  101 J=JJJ
      NTERMS=N+1
      KCONJ=0
C           CLEAR APPROX
      APPROX(1)=0.0
      APPROX(2)=0.0
      APPROX(3)=0.0
      IF(I.EQ.1) GO TO 43
C           REAL COEFFICIENTS
      DO 78 IJFF=1,NTERMS
   78 TEMP(IJFF)=CMPLX(COEFFS(IJFF),0.0)
      GO TO 700
C           COMPLEX COEFFICIENTS
   43 DO 79 IIIX=1,NTERMS
   79 TEMP(IIIX)=CMPLX(COEFFS(2*IIIX-1),COEFFS(2*IIIX))
```

# APPENDIX

```
C
C           CHECK LEADING COEFFICIENT FOR 0 OR 1
  700 TEMPL=REAL(TEMP(1))
      IF(TEMPL.NE.0.)GO TO 701
C           IF REAL IS ZERO,CHECK IMAGINARY
      TEMPL=AIMAG(TEMP(1))
      IF(TEMPL.NE.0.)GO TO 701
C           LEADING COEFFICIENT ZERO                        IERR=1
      IERR=1
      GO TO 1006
C           DIVIDE BY LEADING COEFFICIENT
  701 TEMPM=TEMP(1)
      DO 702 LLA=1,NTERMS
  702 TEMP(LLA)=TEMP(LLA)/TEMPM
C           KCONJ=1, TRIAL VALUE=CONJUGATE
C
  746 IF(KCONJ.NE.0)GO TO 47
C           FIRST TRIAL VALUE
      APPROX(1)=(.01,.01)
C     DIFFERENTIATE
   47 DO 8 II=1,NTERMS
      XPON=NTERMS-II
      NNOW=II+NTERMS
    8 TEMP(NNOW)=XPON*TEMP(II)
      NPON=NTERMS-1
C           KA=0 FOR FIRST TRIAL VALUE
      KA=0
C
C           JONJON=1     SECOND ITERATION
  749 IF(JONJON.EQ.1)APPROX(1)=ROOT(J)
C           CLEAR RDIFF,XIDIFF
      DO 7773 LLL=1,2
      RDIFF(LLL)=0.0
 7773 XIDIFF(LLL)=0.0
C
C           ROOT EVALUATION
C           MAXIMUM ITERATIONS =120
   13 L=2
      PARTR1= REAL(APPROX(1))
      PARTM1=AIMAG(APPROX(1))
      DO 12 K=2,121
C           EVALUATE F(X)
   10 F=(0.0,0.0)
      DO 9 II=1,NTERMS
      F=APPROX(L-1)*F+TEMP(II)
      XF=ABS(REAL(F))
      YF=ABS(AIMAG(F))
C           CHECK FOR OVERFLOW
      IF(XF .GT.OVCON.OR.YF .GT.OVCON)GO TO 14
    9 CONTINUE
C           EVALUATE FPRIME(X)
      FPR=(0.0,0.0)
      DO 11 JJ=1,NPON
      NNOW=JJ+NTERMS
      FPR=APPROX(L-1)*FPR+TEMP(NNOW)
      YFP=ABS(AIMAG(FPR))
      XFP=ABS(REAL(FPR))
C           CHECK FOR OVERFLOW
```

```
      IF(XFP.GT.OVCON.OR.YFP.GT.OVCON)GO TO 14
   11 CONTINUE
C            SEE IF FPRIME=0
      IF(XFP.EQ.0.0.AND.YFP.EQ.0.0)GO TO 14
C            IF NOT ZERO,NEW APPROXIMATION
      APPROX(L)=APPROX(L-1)-F/FPR
      PARTR2= REAL(APPROX(L))
      PARTM2=AIMAG(APPROX(L))
C            SET EITHER PART TO ZERO IF LESS THAN 1.E-21
      IF(ABS(PARTR2).LE.RCONST)PARTR2=0.
      IF(ABS(PARTM2).LE.RCONST)PARTM2=0.
      IF(PARTR2.EQ.0..AND.PARTM2.EQ.0.)GO TO 6732
      GO TO 732
C            ZERO ROOT
 6732 IF(L.EQ.3)APPROX(2)=APPROX(3)
      GO TO 81
C
  732 RDIFF(L-1)=ABS(PARTR2-PARTR1)
      XIDIFF(L-1)=ABS(PARTM2-PARTM1)
      IF(L.EQ.3) GO TO 18
      L=3
      PARTR1=PARTR2
      PARTM1=PARTM2
      GO TO 10
C
C            TEST 1
   18 IF((RDIFF(2)+XIDIFF(2)).LT.(RDIFF(1)+XIDIFF(1)))GO TO  8700
C            TEST 2
      RELTST=(APPROX(3)-APPROX(2))/APPROX(3)
      DIFFR=ABS(REAL(RELTST))
      DIFFXI=ABS(AIMAG(RELTST))
      IF(DIFFR.LT.CONST.AND.DIFFXI.LT.CONST)GO TO 81
 8700 APPROX(2)=CMPLX(PARTR2,PARTM2)
      PARTR1=PARTR2
      PARTM1=PARTM2
      RDIFF(1)=RDIFF(2)
   12 XIDIFF(1)=XIDIFF(2)
C
C            MAXIMUM ITERATIONS EXCEEDED OR
C            OVERFLOW OR
C            FPRIME=0
C            TRY AGAIN WITH SECOND TRIAL VALUE
   14 IF(JONJON.EQ.1)GO TO 136
      IF(KA.EQ.105)GO TO 71
      APPROX(1)=(1.,1.)
      KA=105
      GO TO 13
C            SECOND ITERATION  NONCONVERGENT ROOT  IERR=3
  136 IERR=3
C            STORE RESULT AND IMPROVE NEXT ROOT
      GO TO 82
```

# APPENDIX

```
C
C          FIRST ITERATION  ROOT R NONCONVERGENT   IFERR=2
C          IMPROVE (R-1) ROOTS
   71 IFERR=2
C          IB=LAST CONVERGENT ROOT
      IB=J
      ROOT(J)=APPROX(2)
      IF(IB.NE.1)GO TO 9971
C          FIRST ROOT FAILED    RETURN
      GO TO 1006
 9971 JONJON=1
      GO TO 101
C
C          STORE ROOTS
   81 IF(JONJON.EQ.1) GO TO 82
C          HERE IF FIRST ITERATION
    5 ROOT(J)=APPROX(2)
C          REDUCE POLYNOMIAL BY SYNTHETIC DIVISION
    1 NTERMS=NTERMS-1
      DO 7 IK=2,NTERMS
      TEMP(IK)=ROOT(J)*TEMP(IK-1)+TEMP(IK)
    7 CONTINUE
C          NEXT ROOT IF COMPLEX COEFFICIENTS
      IF(I.EQ.1)GO TO 745
C          HERE IF REAL COEFFICIENTS
      IF(KCONJ.EQ.0)GO TO 744
C          RESET KCONJ IF ROOT IS CONJUGATE OF PREVIOUS ROOT
      KCONJ=0
      GO TO 745
  744 X=REAL(ROOT(J))
      Y=AIMAG(ROOT(J))
      IF(X.EQ.0.)GO TO 745
C          SEE IF REAL OR COMPLEX
      IF(ABS(Y/X).LE.1.E-10)GO TO 745
C          COMPLEX ROOT      TRIAL VALUE=CONJUGATE
      APPROX(1)=CONJG(ROOT(J))
      KCONJ=1
C          NEXT ROOT
  745 J=J+1
      IF(J.NE.NSAVE)GO TO 746
C          LAST ROOT
      ROOT(J)=-TEMP(2)/TEMP(1)
      JONJON=1
      GO TO 101
C
C          IMPROVED ROOT FROM SECOND ITERATION
   82 ROOT(J)=APPROX(2)
      X=ABS(REAL(ROOT(J)))
      Y=ABS(AIMAG(ROOT(J)))
C          SET REAL OR IMAG. TO ZERO IF LESS THAN RCONST
      IF(X.LT.RCONST.AND.Y.LT.RCONST)GO TO 33
      IF(X.LT.RCONST)ROOT(J)=CMPLX(0.0,AIMAG(ROOT(J)))
      IF(Y.LT.RCONST)ROOT(J)=CMPLX(REAL(ROOT(J)),0.0)
      GO TO 108
   33 IF(X.GE.Y) GO TO 34
                         ROOT(J)=CMPLX(0.0,AIMAG(ROOT(J)))
      GO TO 108
   34                       ROOT(J)=CMPLX(REAL(ROOT(J)),0.0)
C
```

# APPENDIX

```
C              IF(R-1) ROOTS IMPROVED,RETURN
  108 IF(J.EQ.IB)GO TO 1005
      J=J+1
C              IF N ROOTS IMPROVED,RETURN
      IF(J.LE.NSAVE)GO TO 749
 1006 N=NSAVE
      RETURN
      END




      SUBROUTINE ITR2 (X,A,B,DELTX,FOFX,E1,F2,MAXI,ICODE)
      X=A
      KX=0
      LX=0
      IF (DELTX)111,111,112
  112 IF (B- A ) 113,113,114
  114 I =0
      IF (FOFX(A))1,2,3
    1 XB1=X
      IF(LX.NE.0)GO TO 1001
      X =X+DELTX
      IF(X-B)1000,1000,1004
 1004 X=B
      LX=1
 1000 IF (FOFX(X))1,2,4
    4 XB=X
      X=X-DELTX/(2.**(I+1))
  999 I=I+1
      IF(MAXI.LT.I)GO TO 444
      IF (FOFX(X))6,2,7
    6 L=1
      XX=XB
      GO TO 18
    7 L=2
      XX=XB1
      GO TO 18
    3 XB1=X
      IF(KX.NE.0)GO TO 1001
      X= X+DELTX
      IF(X-B)1002,1002,1003
 1003 X=B
      KX=1
 1002 IF(FOFX(X))5,2,3
    5 XB=X
      X=X-DELTX/(2.**(I+1))
 998  I=I+1
      IF(MAXI.LT.I)GO TO 444
      IF(FOFX(X))8,2,9
    9 L=3
      XX=XB
      GO TO 18
    8 L =4
      XX=XB1
```

```
   18 IF (ABS(X)-E1)36,36,37
   37 IF (ABS((XX-X)/X)-E1)2,2,17
   36 IF (ABS(XX-X)-E2)2,2,17
   17 GO TO (81,4,81,5),L
   81 XB1 =X
      X=X+DELTX/(2.**(I+1))
      GO TO (999,4,998,5),L
  111 ICODE =2
      GO TO 79
  113 ICODE =4
      GO TO 79
 1001 ICODE =3
      GO TO 79
  444 ICODE= 1
      GO TO 79
    2 ICODE =0
   79 CONTINUE
      RETURN
      END
```

```
      SUBROUTINEINT1A(II,N,NT,CI,SPEC,CIMAX,IERR ,VAR,CUVAR,DER,ELE1,
     1ELE2,ELT,ERRVAL,DERSUB,CHSUB,ITEXT)
      DIMENSION S1VAR(20),SELE1(20),ELE1(20),ELE2(20),DER(21),
     1     EDERV(21),SDY(20),SDY1(20),YINCR(20),ERRVAL(20),ERVOVH(20),
     2ELT(10),SELT(13),RELMIN(20),STEP(3)
      DIMENSION VAR(21),CUVAR(21)
      INTEGER TEX(15)
      INTEGER CODE,TPSH,SUMHAF,STEP,TEST,DCODE
      REAL K1
C     BEGIN INITIALIZATION
      IF (II .GT. 0) GO TO 520
    1 TP=0
      SSPEC=SIGN(SPEC,CI)
      SCIMAX=SIGN(CIMAX,CI)
      VAR1=      VAR(1)
      IF (CI .EQ. 0.0) GO TO 151
      IF (SSPEC .EQ. 0.0) GO TO 7
      IF (ABS(SCIMAX) .GT. ABS(SSPEC) .OR. SCIMAX .EQ..0.0)
     1SCIMAX=SSPEC
C     TEST TO SEE IF VAR IS ZERO
      IF(ABS(VAR1).GT.1.0E-11)GO TO 2
      TP=SSPEC
      GO TO 7
    2 IF ((VAR1/SSPEC) .GT. 1.E-13) GO TO 4
    3 K1=0.0
      GOTO6
    4 K1=1.0
    6 TP=VAR1-AMOD(VAR1,SSPEC)
      IF(ABS(TP-VAR1).LT.1.E-12)K1=1.0
      TP=TP+K1*SSPEC
      IF (ABS((TP-VAR1)/VAR1) .LT. 1.E-11) TP = TP + SPEC
C     TEST FOR DIRECTION OF INTEGRATION
    7 K1=1.0
      IF (CI .LT. 0.0) K1=-1.0
      CIK=CI*K1
```

# APPENDIX

```
      CIMAXK=SCIMAX*K1
      TPK=TP*K1
      VARK=VAR1*K1
C     SET UP STORAGE FOR INTERNAL USE
      NP1=N+1
      NELT=1
      REMAIN=0.0
      NHAF=0
      NTS=NT
      SUMHAF=0
      LOOP=0
      DO 91 I=1,3
   91 STEP(I)=0
      IERR=1
      DO 8 I=1,NP1
    8 CUVAR(I) =VAR(I)
      DO 101 I=1,N
  101 SELE1(I)=ELE1(I)
      IF (NT .EQ. 0) GO TO 13
  100 IF (NT .EQ. 1) GO TO 10
      NTM1=NT-1
      ELTK=K1*ELT(1)
      DO 9 I=1,NTM1
      ELTK2=K1*ELT(I+1)
      IF (ELTK .LT. ELTK2) GO TO9
      GO TO 500
    9 ELTK=ELTK2
   10 CONTINUE
      ELTK=K1*ELT(NELT)
      IF (VARK .LT. ELTK) GO TO11
      IF (NELT .EQ. NT) GO TO 13
      NELT=NELT+1
      GO TO 10
   11 NELTL=NT-NELT+1
      GO TO12
   13 NELTL=0
   12 DO 14 I=1,N
   14 RELMIN(I)=SELE1(I)/128.0
      IF (NT .EQ. 0) GO TO 996
      DO 995 I=1,NT
  995 SELT(I)=ELT(I)
  996 CALL DERSUB
      IF (II .EQ. 4) GO TO 120
      DO 15 I=1,N
   15 FDERV(I)=DER(I+1)
      II=1
      TEST=0
      DO 300 I=1,15
  300 TEX(I)=0
      TEX(1)=1
      TEX(2)=1
      KK3=1
      IF (ITEXT) 635,63,635
  151 PRINT 1000
 1000 FORMAT (//11H CI IS ZERO)
      STOP
C     END OF INITIALIZATION
```

```
 520  I1=1
      TPSH=0
      LTSH=0
      VARK=VAR(1)*K1
      CIK=CI*K1
      S1=VARK+CIK
      IF (SSPEC .EQ. 0.0) GO TO 525
      KK=1
      IF (NELTL.EQ. 0) GO TO 17
      IF (FLTK-TPK) 16,16,17
   16 CV=FLTK
      CODE=1
      GO TO 18
   17 CV=TPK
      CODE=2
   18 IF(ABS(CV).LT.1.E-12)GO TO 530
      IF (CV-S1)20,20,19
   19 IF(ABS((CV-S1)/CV).GE..1E-11)GO TO 535
   20 IF (NELTL .EQ. 0) GO TO 540
      IF(ABS((FLTK-TPK)/CV).LT..1E-11)GO TO 550
      IF (CODE .EQ. 1) GO TO 545
  540 DX=TP-VAR(1)
      TEX(5)=1
      TP=TP+SSPEC
      TPK=TP*K1
      TPSH=1
      GO TO 560
C     SHORT INTERVAL DUE TO BOTH
  550 TP=TP+SSPEC
      TEX(6)=1
      TPK=TP*K1
      TPSH=1
      GO TO 545
C     IF HERE CV IS LIKELY ZERO
  530 IF(S1.LT.-1.0E-12)GO TO 535
      IF (CODE .EQ. 1) GO TO 550
      IF (NELTL .EQ. 0) GO TO 540
      IF(ABS(FLTK).LT.1.0E-12)GO TO 550
      GO TO 540
C     SPEC IS ZERO
  525 IF(ABS(REMAIN).GT..1E-11)GO TO 96
      IF (NELTL .EQ. 0) GO TO 565
   94 IF(ABS(ELTK).GE.1.E-12)GO TO 21
      IF(S1.LT.-1.0E-12)GO TO 565
      GO TO 545
   21 S2=FLTK-S1
      IF(S2) 545,545,22
   22 IF(ABS(S2/ELTK).LT.1.0E-12)GO TO 545
      GO TO 565
C     SHORTINTERVAL IS DUE TO ELT BLOCK
  545 DFLT=      SFLT(NFLT)
      TEX(4)=1
      DX=DFLT- VAR(1)
      REMAIN=CI-DX
      REMAIK=REMAIN*K1
      LTSH=1
      NFLT=NFLT+1
      NFLTL=NELTL-1
      IF(NELTL.EQ.0)GO TO 560
      FLTK=K1*SFLT(NFLT)
      GO TO 560
  565 DX=CI
      TEX(3)=1
      GO TO 560
   96 IF (NELTL .EQ. 0) GO TO 98
      IF (ELTK .LT. (VARK+REMAIK)) GO TO 94
   98 DX=REMAIN
      TEX(7)=1
      REMAIN=0.0
      GO TO 560
  535 DX=CI
      TEX(3)=1
      TEST=1
      GO TO 555
C
C     BEGIN RUNGE-KUTTA
C
  560 TEST=0
  555 DO 24 I=1,N
   24 S1VAR(I)= VAR(I+1)
  575 CUVAR(1)=VAR(1)
  576 DO 25 I=1,N
      SDY(I)=DER(I+1)
   25 CUVAR(I+1)=S1VAR(I)+(DX*DER(I+1))/2.0
      CUVAR(1)=CUVAR(1)+DX/2.0
      CALL DERSUB
      IF (II .EQ. 4) GO TO 120
  580 DO 26 I=1,N
      SDY(I)=SDY(I)+2.0*DER(I+1)
   26 CUVAR(I+1)=S1VAR(I)+(DX*DER(I+1))/2.0
      CALL DERSUB
      IF (II .EQ. 4) GO TO 120
  585 DO 27 I=1,N
      SDY(I)=SDY(I)+2.0*DER(I+1)
   27 CUVAR(I+1)=S1VAR(I)+DX*DER(I+1)
      CUVAR(1)=CUVAR(1)+DX/2.0
      CALL DERSUB
      IF (II .EQ. 4) GO TO 120
  590 DO 90 I=1,N
      SDY(I)=(SDY(I)+DER(I+1))/6.0
   90 CONTINUE
      IF (LOOP) 28,28,29
   28 DO30 I=1,N
      SDY1(I)=SDY(I)
      YINCR(I)=0.0
   30 DER(I+1)=FDERV(I)
      DX=DX/2.0
      LOOP=1
      GO TO 575
C
C     LOOP WAS NOT ZERO
C
   29 DO 31 I=1,N
   31 YINCR(I)=YINCR(I)+SDY(I)
      IF (LOOP .EQ. 2) GO TO 33
      DO 32 I=1,N
      S1VAR(I)=VAR(I+1)+DX*YINCR(I)
   32 CUVAR(I+1)=S1VAR(I)
      CUVAR(1)=VAR(1)+DX
      LOOP=2
      CALL DERSUB
      IF (II .EQ. 4) GO TO 120
      GO TO 576
   33 LOOP=0
      H=2.0*DX
      DO 34 I=1,N
      ERVOVH(I)=(YINCR(I)/2.0-SDY1(I))/15.0
      ERRVAL(I)=H*ERVOVH(I)
   34 S1VAR(I)=S1VAR(I)+DX*SDY(I)+ERRVAL(I)
C
C     S1VAR HOLD THE APPROXIMATE ANSWERS
C
      IF (SCIMAX) 36,35,36
   36 IF(ABS(SCIMAX-CI).LT.1.0E-12)GO TO 38
   35 IF(ABS(H-CI).GT.1.0E-12)GO TO 38
      DCODE=0
      GO TO 605
   38 DCODE=1
  605 CONTINUE
      I=0
   40 I=I+1
      IF (I .GT. N) GO TO 45
      IF (ABS(S1VAR(I)) .LT. ELE2(I)) GO TO 40
      RELER=ABS(ERRVAL(I)/S1VAR(I))
      IF (RELER .GT. SELE1(I)) GO TO 615
      IF (RELER .GT. RELMIN(I)) DCODE=1
      GO TO 40
   45 CONTINUE
      IF (DCODE-1) 610,620,610
  610 CONTINUE
      IF (SSPEC)41,42,41
   42 IF (SCIMAX)41,43,41
   43 CI=2.0*CI
      TEX(8)=1
      NHAF=NHAF-1
      GO TO 620
   41 IF (2.0*ABS(CI) .LE. ABS(SCIMAX)) GO TO 43
   44 CI=SCIMAX
      TEX(8)=1
      GO TO 620
C
C     HALF INTERVAL
  615 NHAF=NHAF+1
      TEX(9)=1
      NVAR=I+1
      IF (NHAF-8)47,47,505
   47 IF (LTSH .EQ. 0) GO TO 48
      TEST=1
      LTSH=0
      NELT=NELT-1
      NELTL=NELTL+1
      ELTK=K1*SELT(NELT)
      REMAIN=0.0
   48 IF (TPSH .EQ. 0) GO TO 49
      TEST=1
      TP=TP-SSPEC
      TPK=K1*TP
      TPSH=0
   49 IF (SSPEC .NE. 0.0) GO TO 998
      TEST=0
      IF (ABS(CI-2.0*DX) .GT. 1.E-12)GO TO 1100
  998 CI=DX
  999 DX=DX/2.0
      CIK=K1*CI
      DO 50 I=1,N
      S1VAR(I)=VAR(I+1)
      DER(I+1)=FDERV(I)
      SDY1(I)=YINCR(I)-SDY(I)
   50 YINCR(I)=0.0
      KK3=2
      IF (ITEXT .EQ. 1) GO TO 637
   99 LOOP=1
      GO TO 575
 1100 CONTINUE
      IF (NHAF .GT. 1) GO TO 999
      NTS=NTS+1
      IF (NTS .GT. 13) GO TO 998
      ACV=VAR(1)+CI
      ACVK=ACV*K1
      IF (NELTL .EQ. 0) GO TO 1102
      NLT=NELT
 1101 FLTK1=SELT(NLT)*K1
      IF (ACVK .LT. FLTK1) GO TO 1103
      NLT=NLT+1
      IF (NLT .EQ. NTS) GO TO 1106
      GO TO 1101
 1102 SELT(NELT)=ACV
      GO TO 1105
 1103 NLTP1=NLT+1
      I=NTS
 1108 SELT(I)=SELT(I-1)
      IF( I   .EQ. NLTP1) GO TO 1106
      I=I-1
      GO TO 1108
 1106 SELT(NLT)=ACV
 1105 NELTL=NELTL+1
      TEX(9)=0
      TEX(10)=1
      FLTK=K1*SELT(NELT)
      GO TO 999
C
C     DOUBLE PRECISIION UPDATING
C
  620 LOOP=0
      DH=H
      DO 51 I=1,N
      PHI=FRVOVH(I)+YINCR(I)/2.0
      DPHI=PHI
   51 CUVAR(I+1)=VAR(I+1)+DH*DPHI
      CUVAR(1)=VAR(1)+DH
      CALL DERSUB
      IF (II .EQ. 4) GO TO 120
      CALL CHSUB
      IF (II-2) 54,600,121
  121 TEST=0
   54 DO 57 I=1,N
   57 FDERV(I)=DER(I+1)
      SUMHAF=SUMHAF+NHAF-STEP(1)
      STEP(1)=STEP(2)
      STEP(2)=STEP(3)
      STEP(3)=NHAF
      NHAF=0
      IERR=1
      IF (SUMHAF-8) 63,63,510
   63 DO 59 I=1,NP1
   59 VAR(I)=CUVAR(I)
      TEX(12)=1
  501 KK3=4
      IF (ITEXT .EQ. 1) GO TO 637
   58 IF (TEST .EQ. 1) GO TO 520
  120 RETURN
C
C     RECOMPUTE INTERVAL
C
  600 TEST=0
      NHAF=0
      II=1
      DX=CI
      TEX(11)=1
      KK3=3
      IF (ITEXT .EQ. 1) GO TO 636
   70 CIK=CI*K1
      DO 60 I=1,N
      DER(I+1)=FDERV(I)
   60  CUVAR(I)= VAR(I)
       CUVAR(N+1)= VAR(N+1)
      IF (TPSH .EQ. 0) GO TO 61
      TP=TP-SPEC
      TPK=TP*K1
      TPSH=0
   61 IF (LTSH .EQ. 0) GO TO 555
      NELT=NELT-1
      REMAIN=0.0
      NELTL=NELTL+1
      FLTK=SFLT(NELT)*K1
      GO TO 555
  636 PRINT 183,VAR(1),DX
      GO TO 102
  635 IF(TEX(1).EQ.1)PRINT 171,VAR(1)
      IF(TEX(2).EQ.1)PRINT 172,CI,CIMAX,SPEC
  637 IF(TEX(3).EQ.1)PRINT 173
      IF(TEX(4).EQ.1)PRINT 174,H
      IF(TEX(5).EQ.1)PRINT 175,H
      IF(TEX(6).EQ.1)PRINT 176,H
      IF(TEX(7).EQ.1)PRINT 184,H
      IF(TEX(8).EQ.1)PRINT 177,CI
      IF(TEX(9).EQ.1)PRINT 178,NVAR,CI
      IF(TEX(10).EQ.1)PRINT 185,NVAR,DX
      IF(TEX(11).EQ.1)PRINT 183,VAR(1),DX
      IF(TEX(12).EQ.1)PRINT 179,VAR(1)
      IF(TEX(13).EQ.1)PRINT 180
      IF(TEX(14).EQ.1)PRINT 181
      IF(TEX(15).EQ.1)PRINT 182
  102 DO 320 I=3,13
  320 TEX(I)=0
```

# APPENDIX

```
      GO TO (120,99,70,58),KK3
  171 FORMAT (33H INITIALIZATION STARTS AT VAR(1)=,F16.8/)
  172 FORMAT (4H CI=,F15.8,9H    CIMAX=,F15.8,8H    SPEC=,F15.8/)
  173 FORMAT (37H DX IS THE FULL COMPUTING INTERVAL CI/)
  174 FORMAT (28H DX IS A SHORTENED INTERVAL ,E15.8,25H  DUE TO A CRITIC
     1AL VALUE/)
  175 FORMAT (28H DX IS A SHORTENED INTERVAL ,E15.8,21H  DUE TO A SPEC V
     1ALUE/)
  176 FORMAT (28H DX IS A SHORTENED INTERVAL ,E15.8,39H  DUE TO BOTH A S
     1PEC AND CRITICAL VALUE/)
  177 FORMAT (27H CI HAS BEEN LENGTHENED TO ,F16.8/)
  178 FORMAT (5H VAR(,I2,32H) HAS CAUSED CI TO BE HALVED TO ,E16.8/)
  179 FORMAT (27H VAR(1) HAS BEEN UPDATED TO,F16.8,/)
  180 FORMAT (31H ERROR RETURN-ELT NOT MONOTONIC/)
  181 FORMAT (55H ERROR RETURN-HAVE HALVED 9 TIMES OVER LAST 3 INTERVALS
     1/)
  182 FORMAT (45H ERROR RETURN-HAVE HALVED 9 CONSECUTIVE TIMES/)
  183 FORMAT (31H INTERVAL RECOMPUTED AT VAR(1)=,E16.8,9H WITH DX=,F16.8
     1/)
  184 FORMAT(25H DX IS SHORTENED INTERVAL,E16.8,28H DUE TO A PREVIOUS FL
     1T VALUE/)
  185 FORMAT(5H VAR(,I2,32H) HAS CAUSED DX TO BE HALVED TO ,E16.8,38H BU
     1T NOT CI SINCE CI ALREADY SHORTENED/)
  500 IERR=2
      TEX(13)=1
      TEST=0
      GO TO 63
  505 IERR=3
      TEX(15)=1
      TEST=0
      GO TO 501
  510 IERR=4
      TEST=0
      TEX(14)=1
      GO TO 63
      END
```

# APPENDIX

## Sample Output

The following is an example case with the input quantities and results printed:

```
$NAM1

A        =  0.0,

SIGMA    =  0.1F-01,

ALPHA    =  C.2F+CO,

PRMIN    =  0.5E-01,

CII      =  0.2E-12,

CIMAX    =  C.4E-C3,

ALLUI    =  0.1E-04,

BULUI    =  C.12E+C1,

DELTX    =  0.1E+00,

EP1      =  0.1E-06,

EP2      =  0.1F-C6,

NT       =  1,

ICPBETA  =  0,

$END



$NAM2

VARO     =  C.0,  C.C,  0.995E-05, -0.199E-C4,

ELE1     =  C.1E-06,  0.1E-06,  0.1E-06,

ELE2     =  0.1E-07,  0.1E-C7,  0.2E-C7,

ELT      =  C.1E+01, I, I, I, I, I, I, I, I, I,

$END



$NAM3

BETA     = I,

$ENC
```

```
BODIES OF REVOLUTION HAVING MINIMUM VISCOUS PLUS (NEWTONIAN) PRESSURE DRAG IN HYPERSONIC FLOW


        INPUT              YO                      A              SIGMA             ALPHA
                    9.95000000E-06    0.                   1.00000000E-02    2.00000000E-01


                     INITIAL U OBTAINED FROM EULER EQUATION
                     BETA NOT INCLUDED IN LAMBDA DOT EQUATION


            X                 Y             Y PRIME

  0.                 9.95000000E-06    9.99999986E-01
  5.00465375E-02     5.24688594E-03    7.82930417E-02
  1.00446538E-01     8.83092859E-03    6.57406731E-02
  1.50846538E-01     1.19701598E-02    5.93708350E-02
  2.01246538E-01     1.48519491E-02    5.52342547E-02
  2.51646538E-01     1.75564510E-02    5.22272838E-02
  3.02046538E-01     2.01276547E-02    4.98934044E-02
  3.52446538E-01     2.25930939E-02    4.80023159E-02
  4.02846538E-01     2.49714927E-02    4.64226021E-02
  4.53246538E-01     2.72763376E-02    4.50726701E-02
  5.03646538E-01     2.95177632E-02    4.38985957E-02
  5.54046538E-01     3.17036372E-02    4.28630186E-02
  6.04446538E-01     3.38402272E-02    4.19390632E-02
  6.54846538E-01     3.59326325E-02    4.11067857E-02
  7.05246538E-01     3.79850745E-02    4.03510107E-02
  7.55646538E-01     4.00011000E-02    3.96599425E-02
  8.06046538E-01     4.19837262E-02    3.90242530E-02
  8.56446538E-01     4.39355483E-02    3.84364380E-02
  9.06846538E-01     4.58588197E-02    3.78903641E-02
  9.57246538E-01     4.77555131E-02    3.73809887E-02
  1.00000000E+00     4.93449116E-02    3.69745149E-02


   TOTAL DRAG COEFF        PRES DRAG COEFF    FRICTION DRAG COEFF
     4.09765800E-03         4.09765800E-03          0.
```

# REFERENCES

1. Eggers, A. J., Jr.; Resnikoff, Meyer M.; and Dennis, David H.: Bodies of Revolution Having Minimum Drag at High Supersonic Airspeeds. NACA Rep. 1306, 1957. (Supersedes NACA TN 3666.)

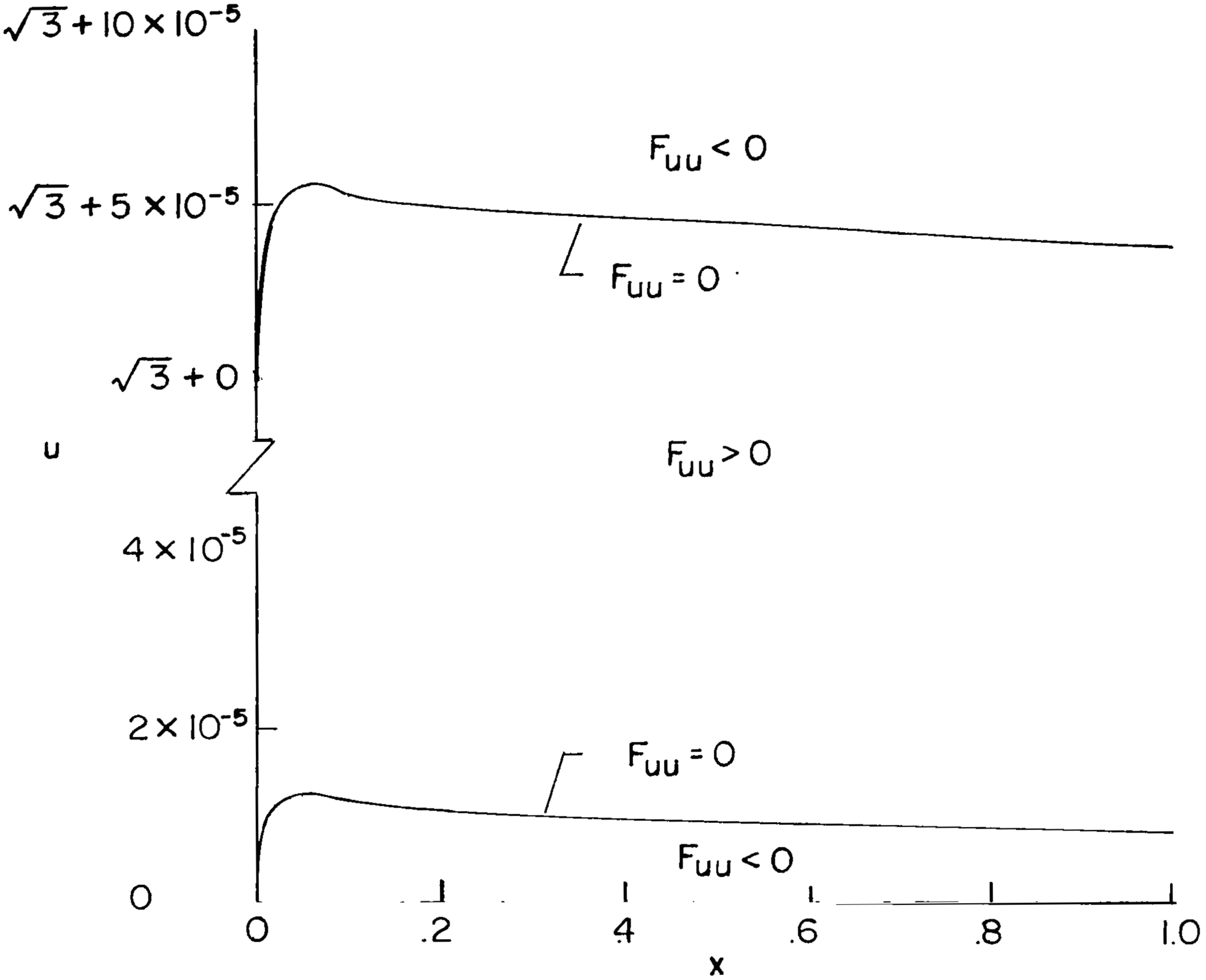
2. Suddath, Jerrold H.; and Oehman, Waldo I.: Minimum Drag Bodies With Cross-Sectional Ellipticity. NASA TN D-2432, 1964.

3. Suddath, Jerrold H.; and Oehman, Waldo I.: Minimum Drag Bodies With Elliptical Cross Sections. J. Astronaut. Sci., vol. XII, no. 4, 1965, pp. 135-141.

4. Kennet, H.: The Effect of Skin Friction on Optimum Minimum-Drag Shapes in Hypersonic Flow. J. Aerospace Sci., vol. 29, no. 12, Dec. 1962, pp. 1486-1487.

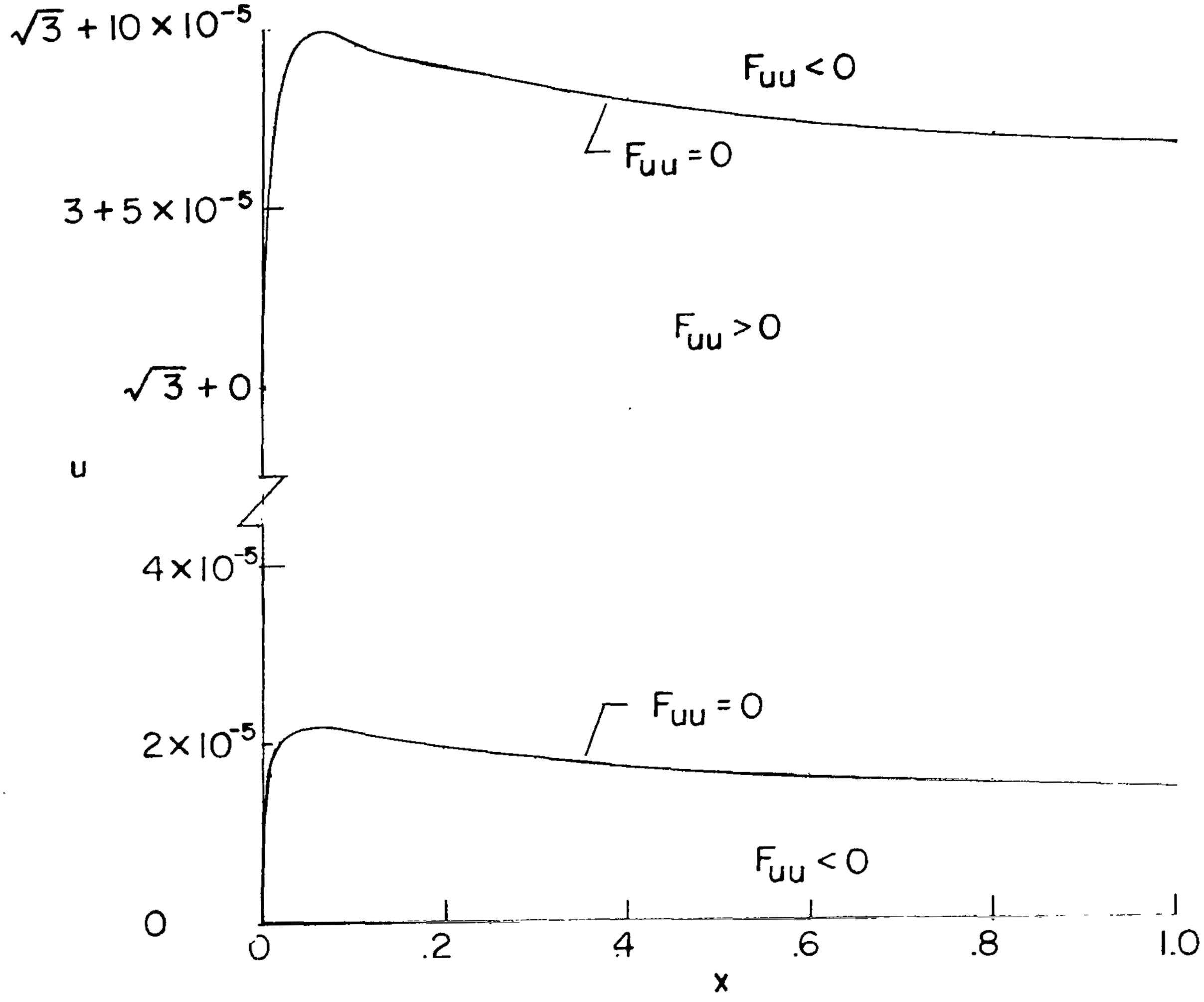
5. Miele, Angelo; and Cole, Julian: Optimum Slender Bodies in Hypersonic Flow With a Variable Friction Coefficient. AIAA J., vol. 1, no. 10, Oct. 1963, pp. 2289-2293.

6. Miele, Angelo (Editor): Theory of Optimum Aerodynamic Shapes. Academic Press, Inc., c.1965.



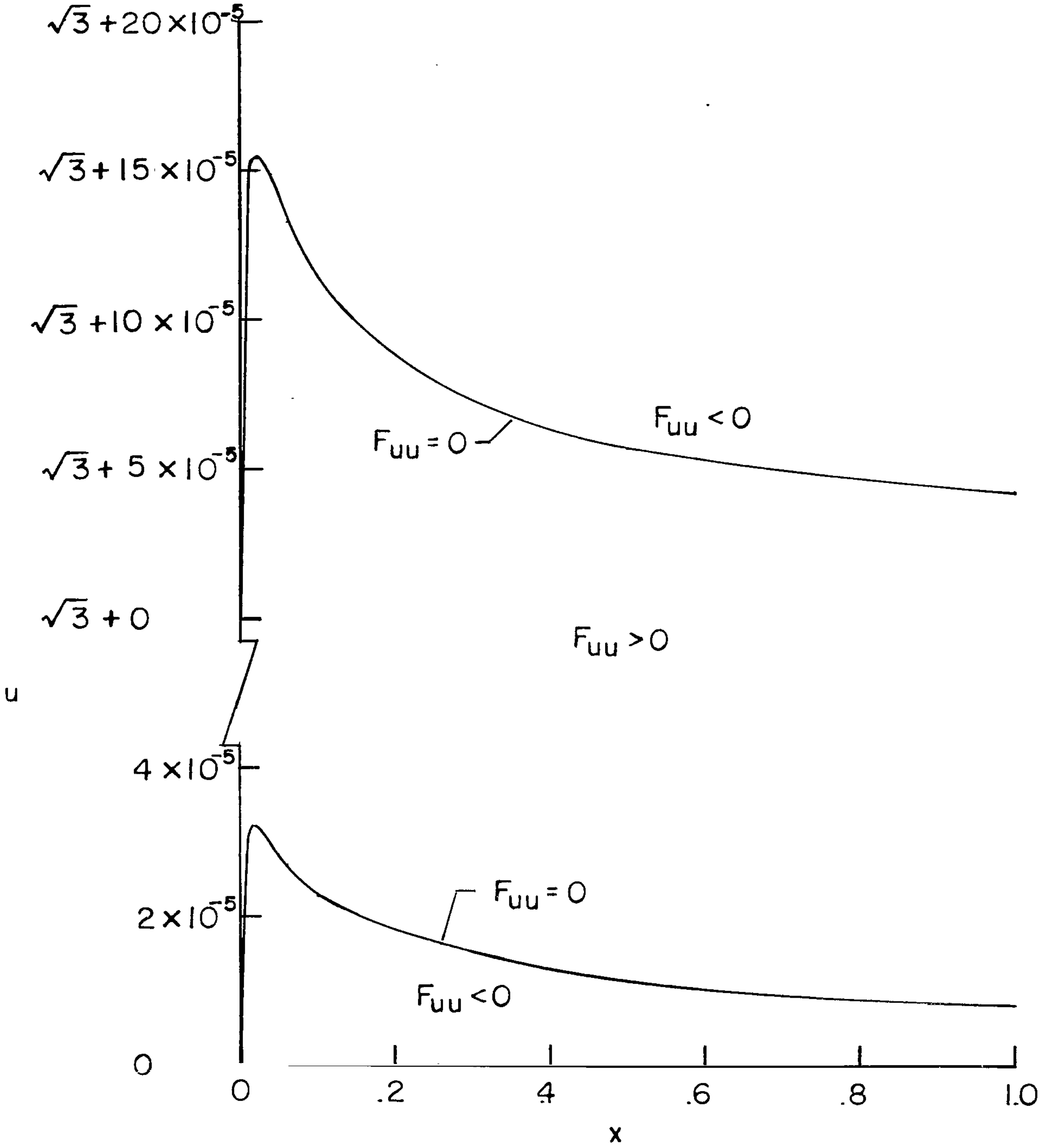
(a) A = 0.0005; α = 0.2.

Figure 1.- Legendre boundaries. σ = 0.01.



(b) A = 0.0009; α = 0.2.

Figure 1.- Continued.



(c) A = 0.0002; α = 0.5.

Figure 1.- Concluded.

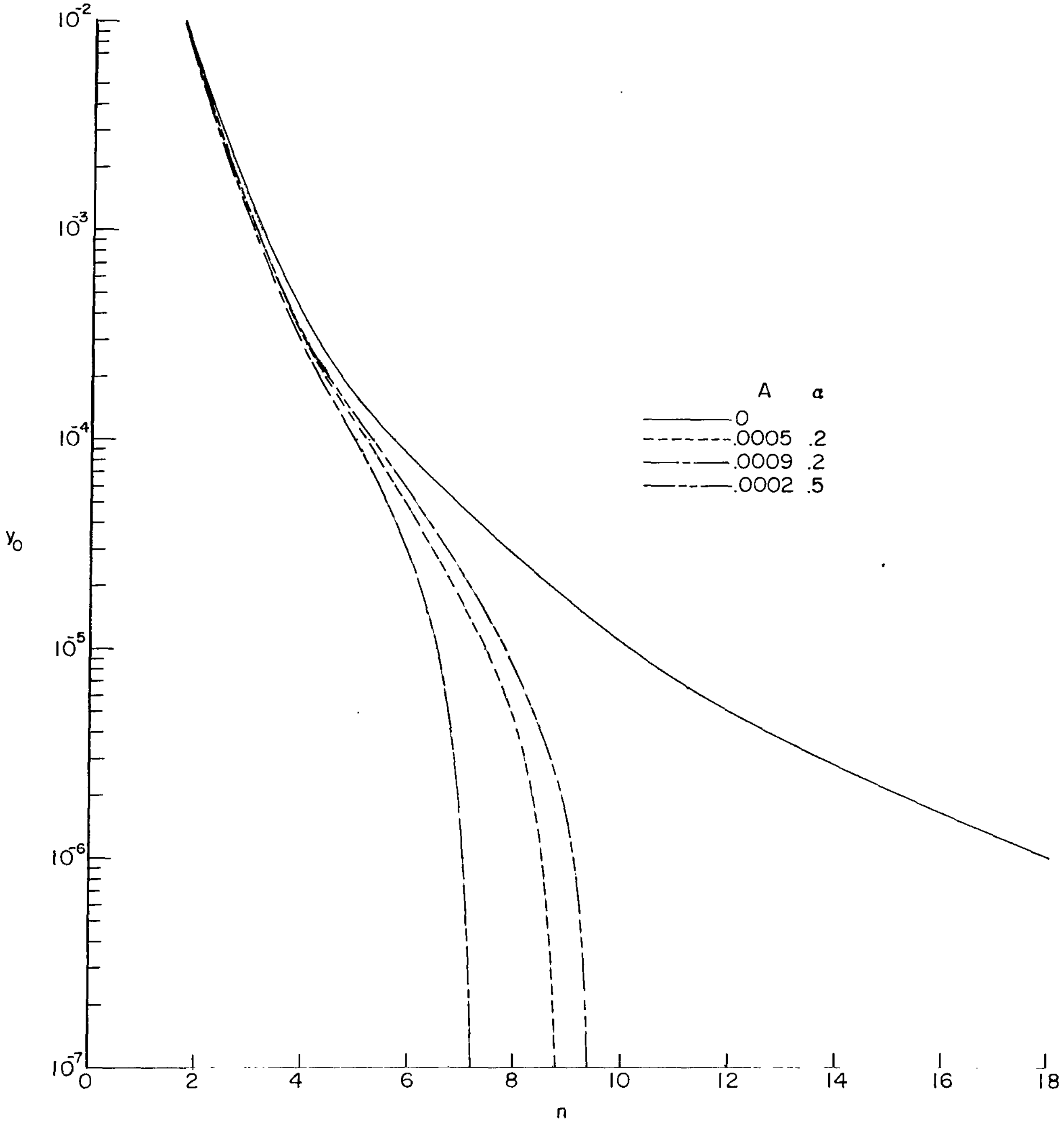
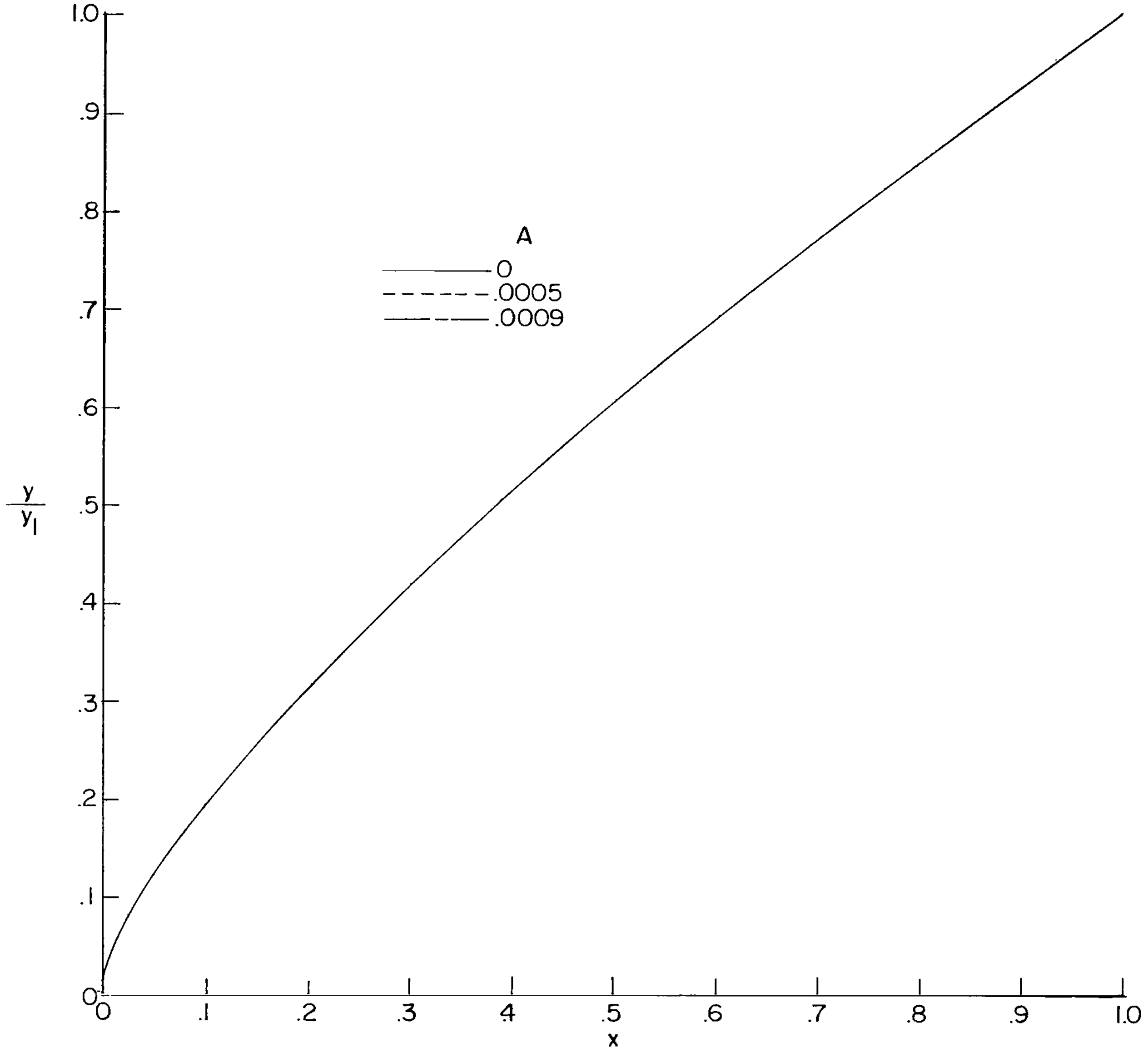
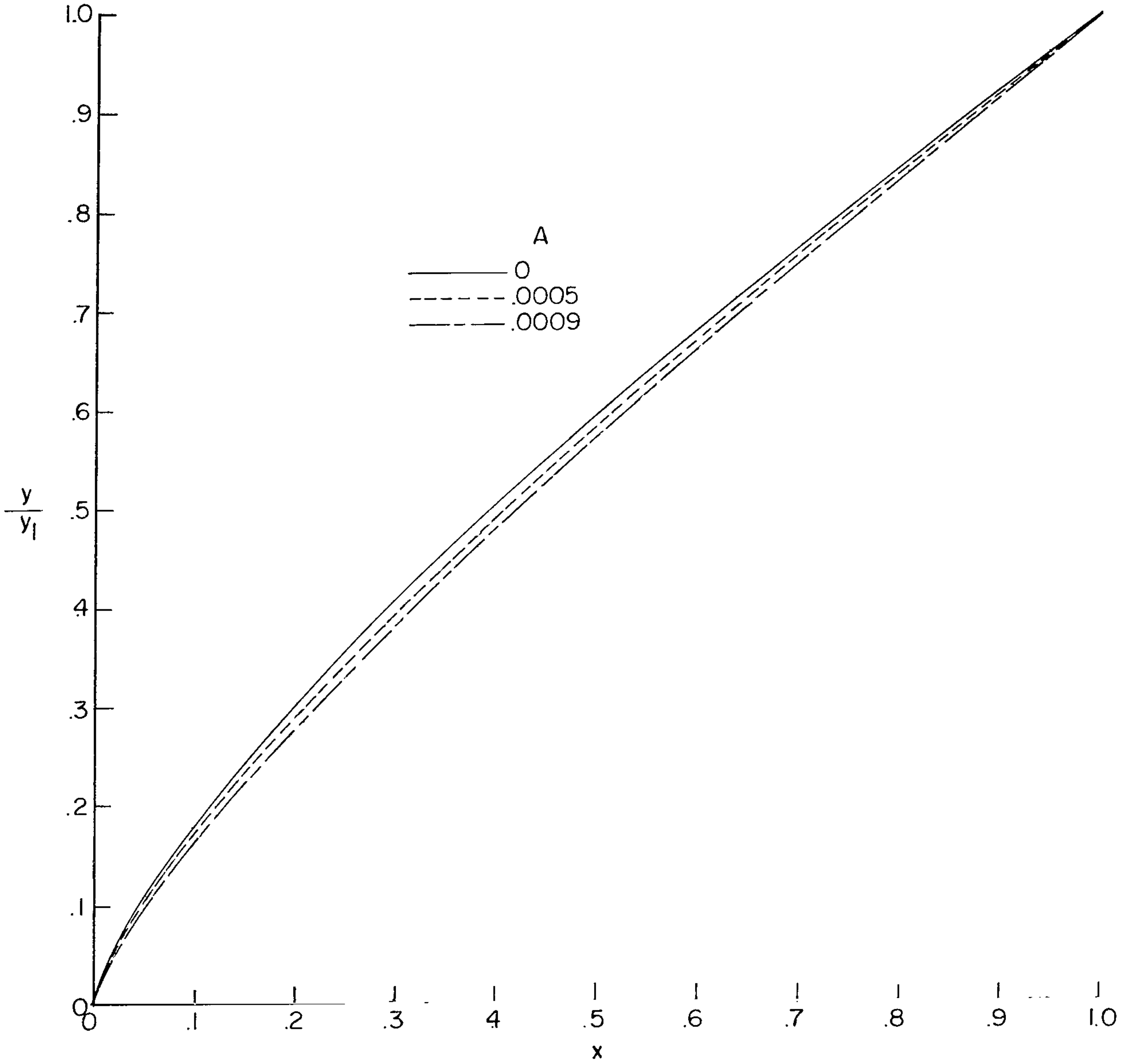


Figure 2.- Initial condition $y_0$ required to obtain a given fineness ratio n. $\sigma$ = 0.01.



(a) Fineness ratio, 2.

Figure 3.- Minimum-drag body profiles. Length and base height given; σ = 0.01.



(b) Fineness ratio, 5.

Figure 3.- Concluded.

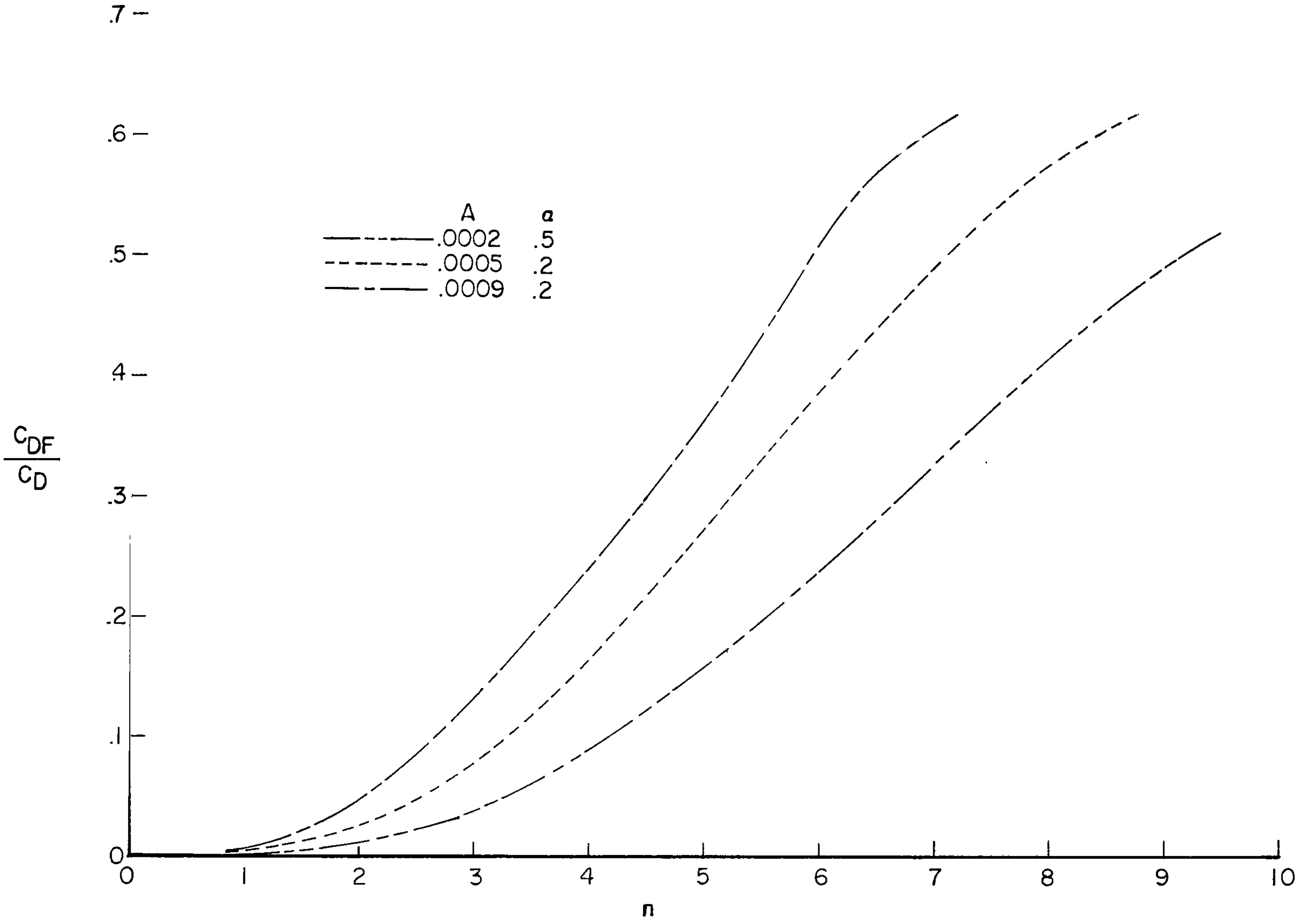
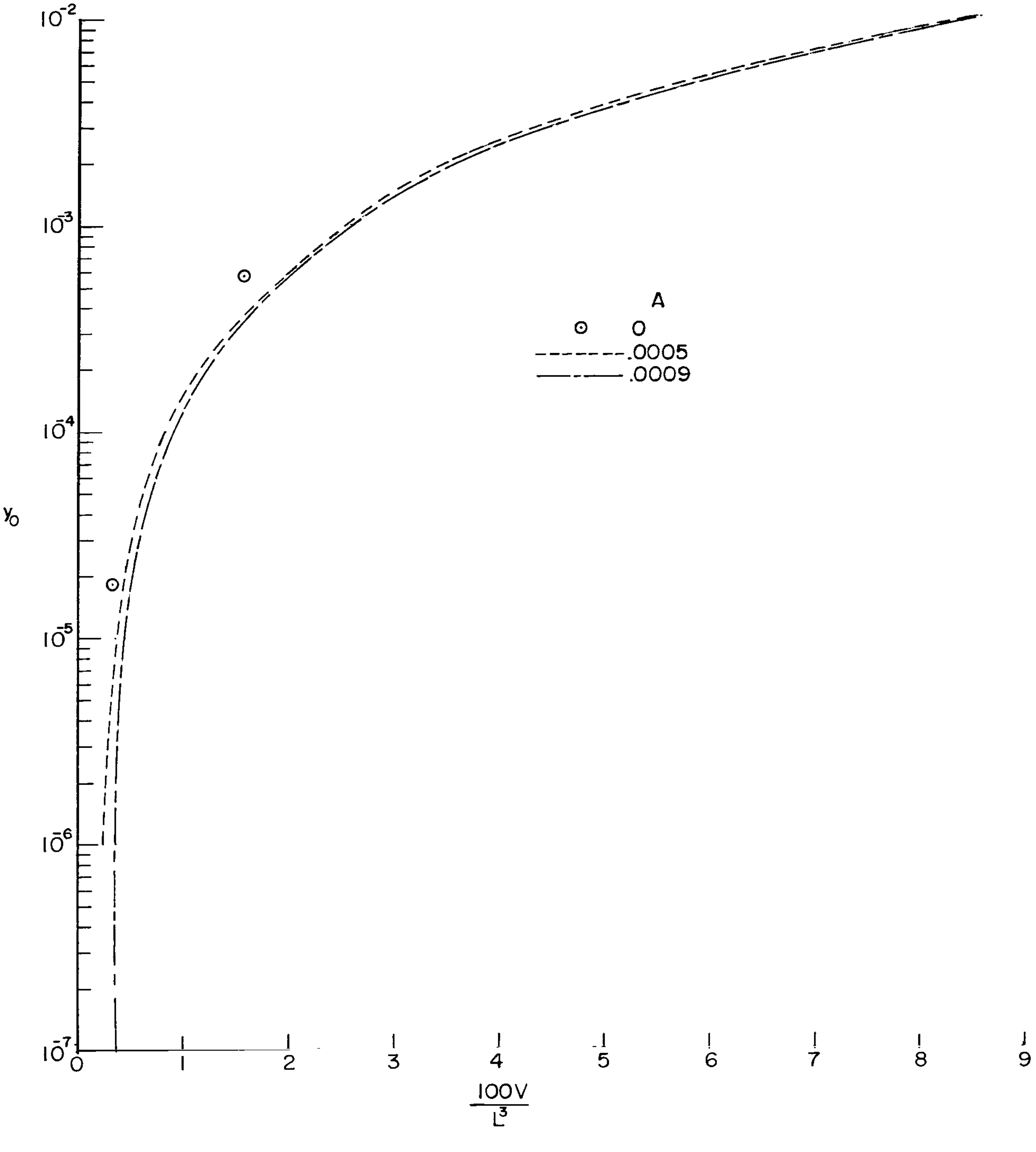
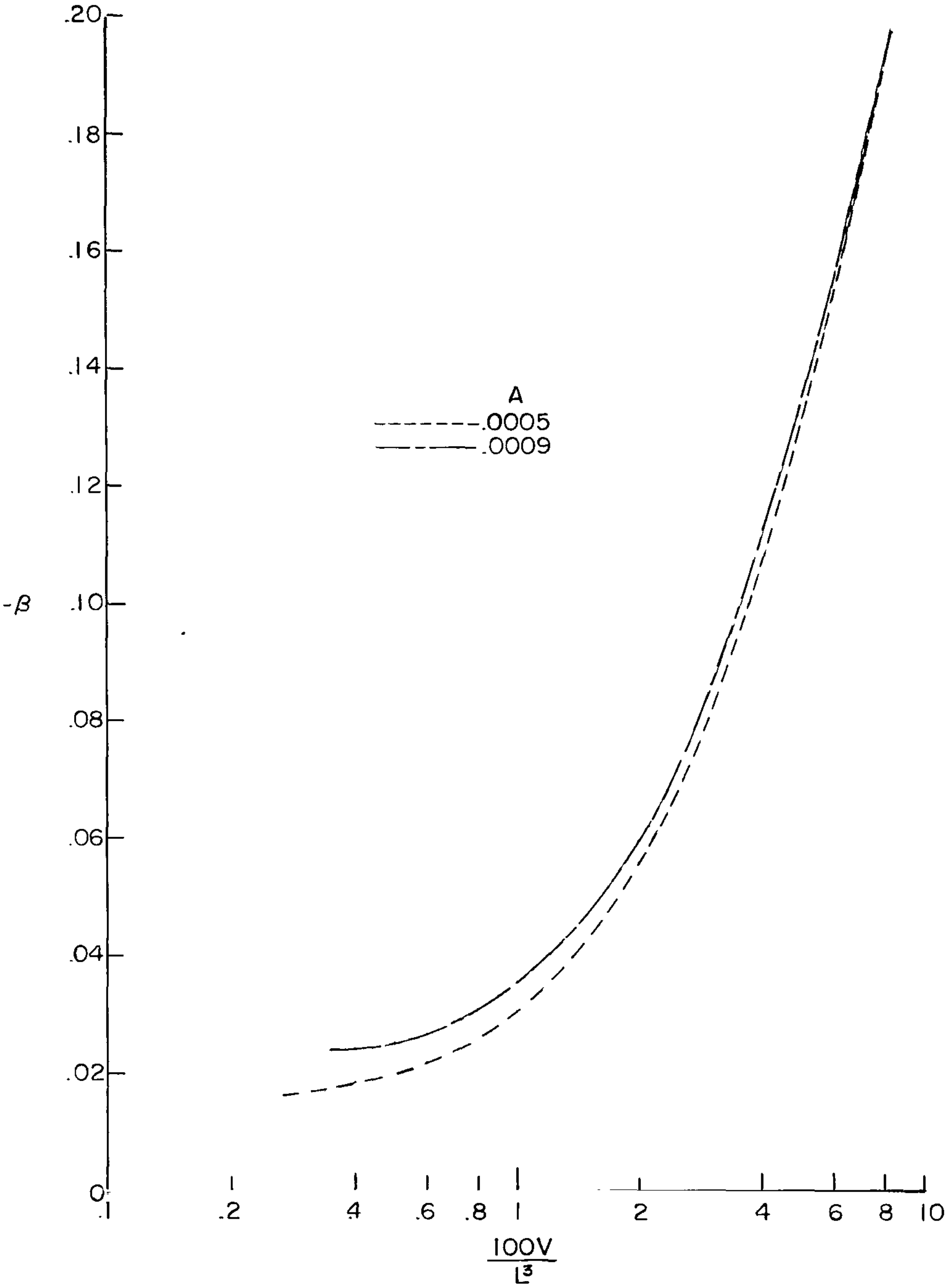


Figure 4.- Ratio of friction drag coefficient to total drag coefficient for given fineness ratio. σ = 0.01.



(a) Radius at nose.

Figure 5.- Radius at nose and Lagrange multiplier for given length and volume. $\sigma = 0.01$.



(b) Lagrange multiplier.

Figure 5.- Concluded.

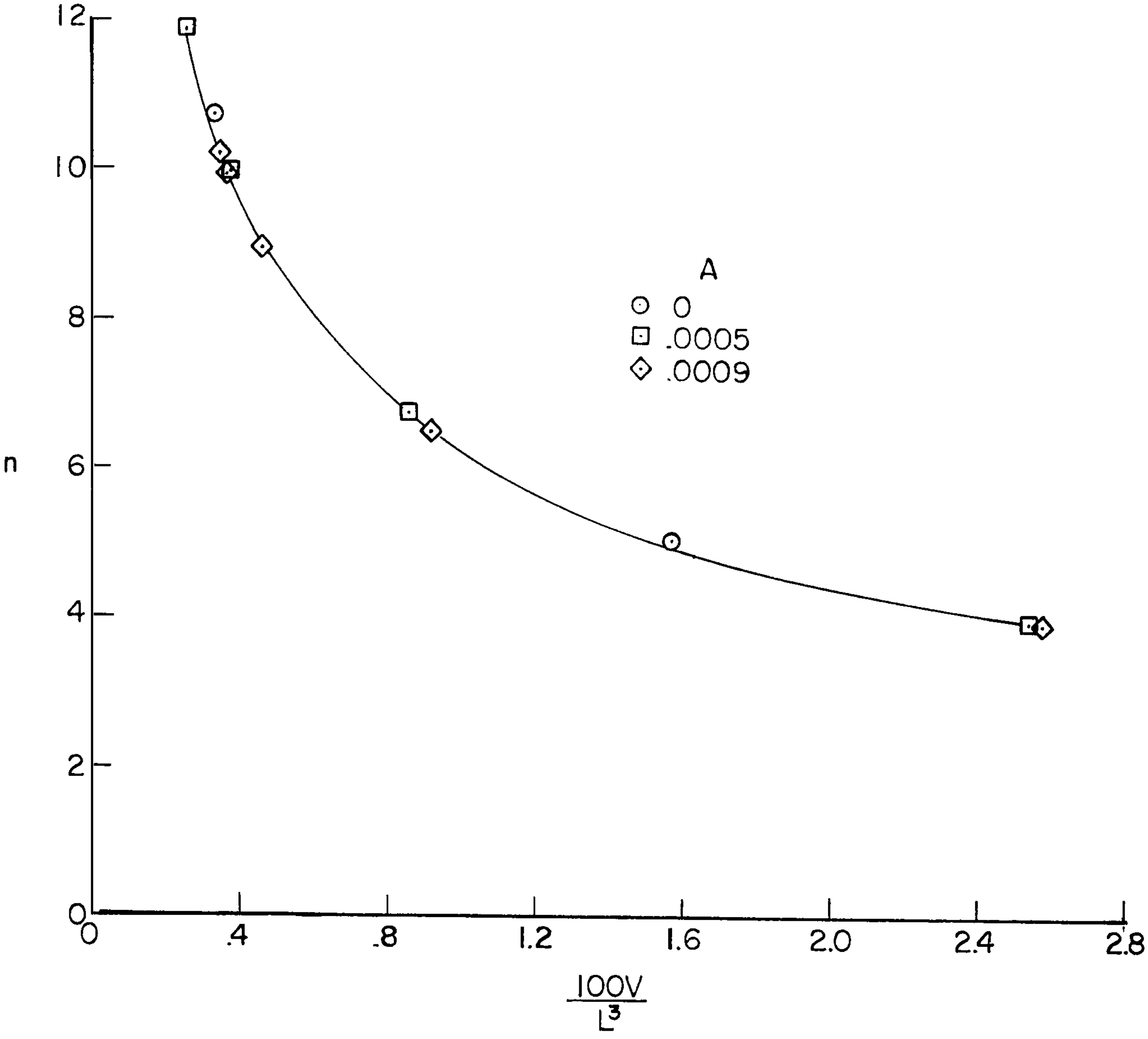


Figure 6.- Body fineness ratio for given length and volume. $\sigma$ = 0.01.

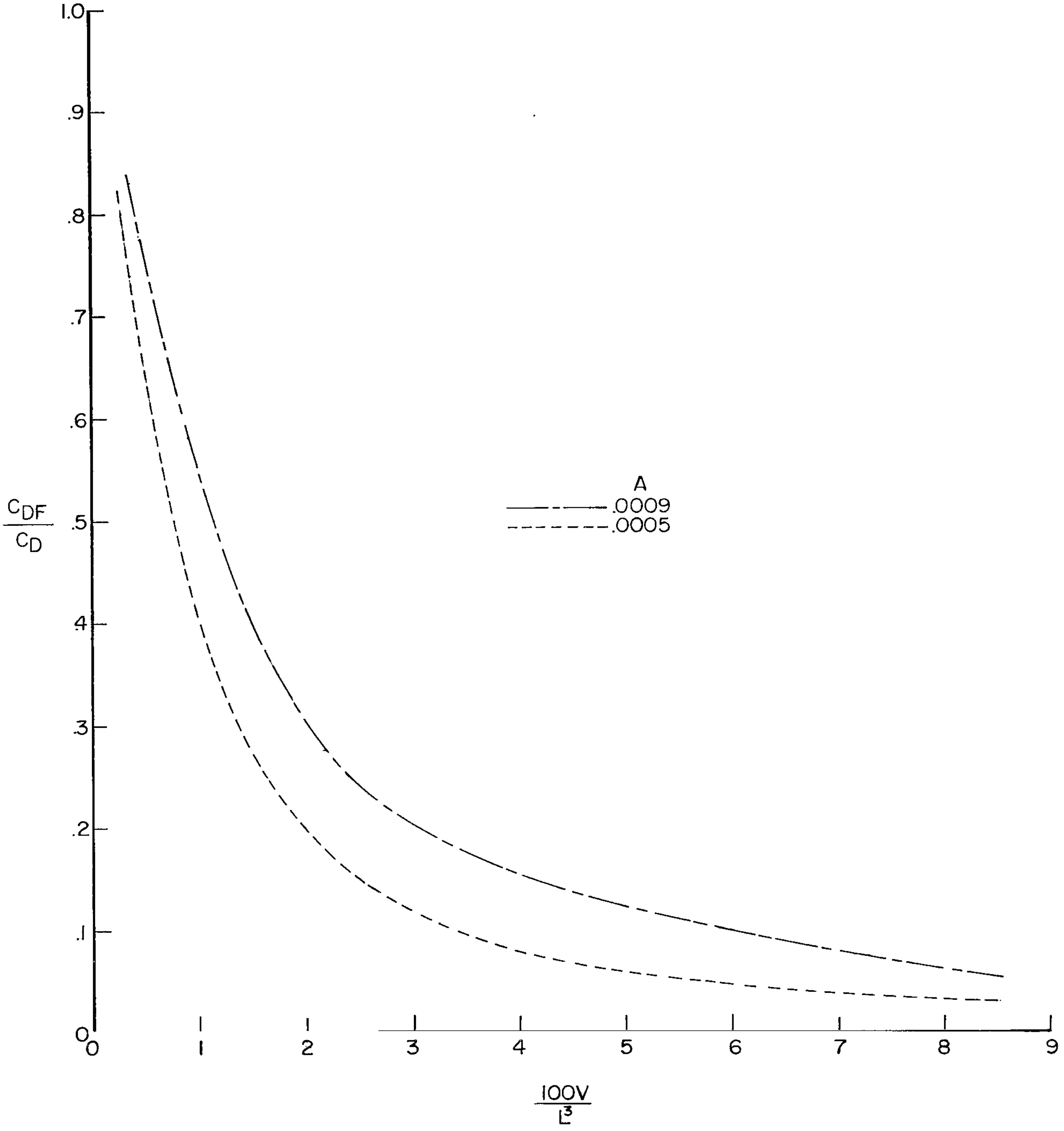
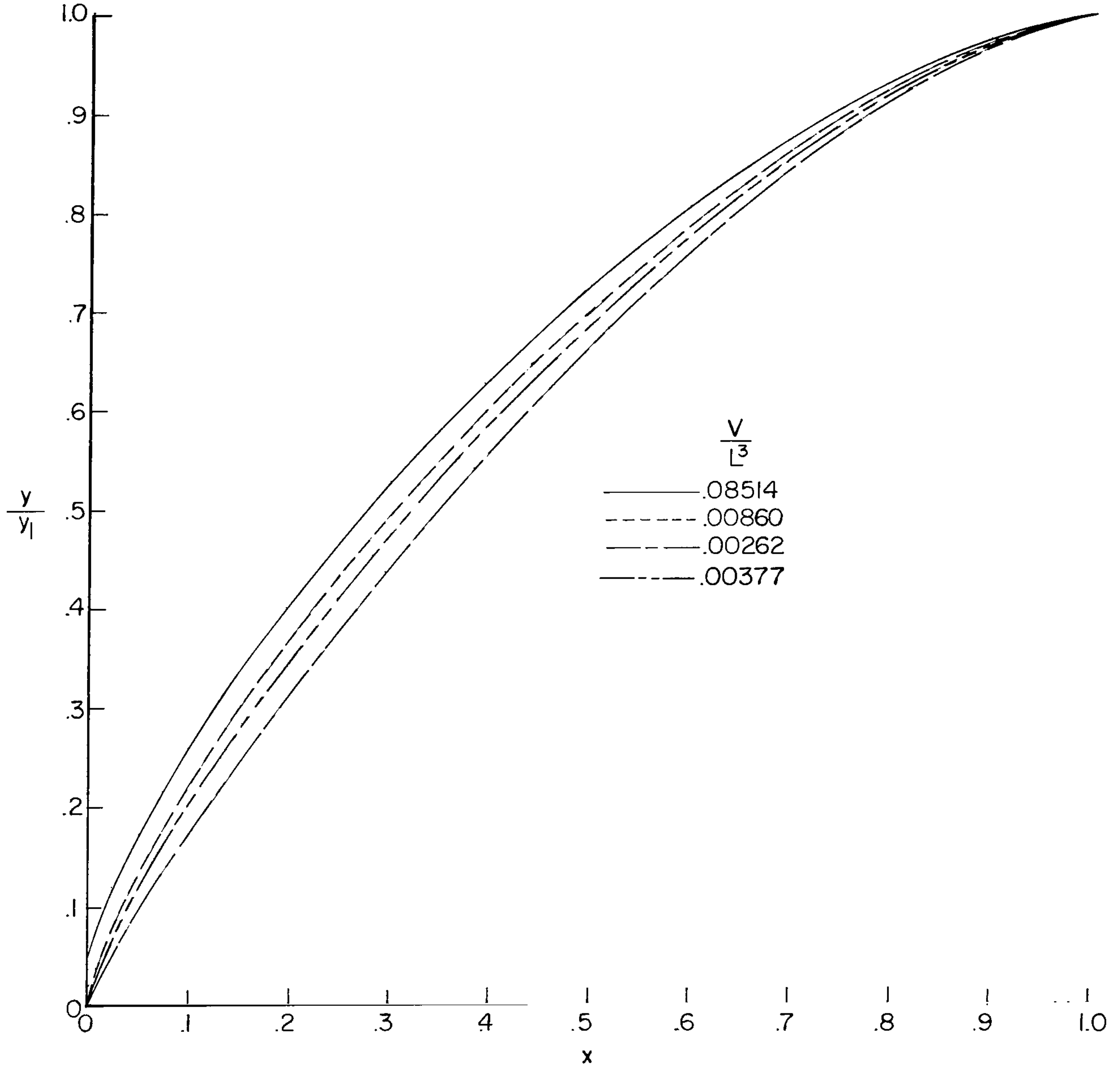
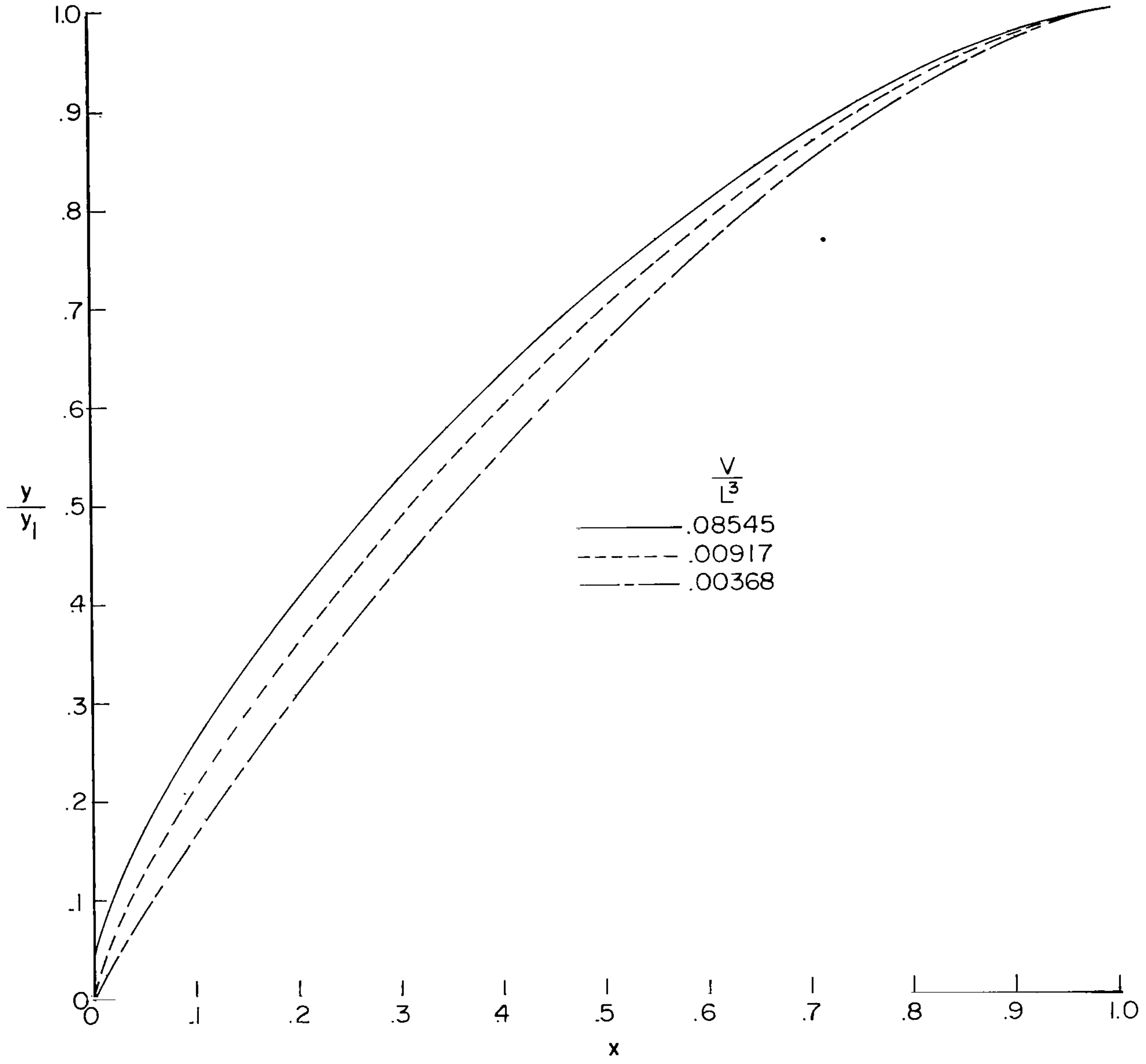


Figure 7.- Drag coefficient ratio of minimum-drag bodies with given length and volume. σ = 0.01.



(a) A = 0.005; α = 0.2.

Figure 8.- Minimum-drag body profiles. Length and volume are given; σ = 0.01.



(b) A = 0.0009; α = 0.2.

Figure 8.- Concluded.

NASA-Langley, 1969 —— 1 L-6451

*"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."*

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

# NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

*Details on the availability of these publications may be obtained from:*

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546