

MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

AD A110089

NUSC Technical Report 6343A  
15 January 1982

12

# A Numerical Treatment of the Dynamic Motion of a Zero Bending Rigidity Cylinder in a Viscous Stream

Ding Lee  
Surface Ship Sonar Department

Robert M. Kennedy  
Submarine Sonar Department

DTIC FILE COPY

DTIC  
JAN 26 1982  
A



**Naval Underwater Systems Center**  
Newport, Rhode Island / New London, Connecticut

Approved for public release, distribution unlimited.

01 26 82 026

**Preface**

This report was prepared under NUSC Project No. J68028, "Towed Array Improvement Program," Principal Investigator, Dr. Robert M. Kennedy (Code 3231). The sponsoring activity was Naval Sea Systems Command, Project Manager, Commander R. Welsh (Code 63D).

The Technical Reviewer was Dr. Sung-Hwan Ko, (Code 3233).

**Reviewed and Approved: 10 February 1981**



**J. W. KYLE**

**Head, Submarine Sonar Department**



**D. WALTERS**

**Head, Surface Ship Sonar Department**

The authors of this report is located at the  
New London Laboratory, Naval Underwater Systems Center,  
New London, Connecticut 06320.



20. (Cont'd)

Fourier sense. This paper presents a general purpose numerical treatment formulated to overcome these difficulties. The numerical approach is based on finite difference schemes applied in conjunction with powerful numerical ordinary differential equation methods. The theory is examined with respect to consistency, stability, and convergence of these numerical procedures. A numerical example is included to demonstrate the validity of the treatment. Although an explicit boundary condition is absent from this study, a derived boundary condition is demonstrated to be adequate.

TABLE OF CONTENTS

	Page
1. INTRODUCTION . . . . .	1
2. POSED PROBLEMS . . . . .	3
3. THE NUMERICAL TREATMENT . . . . .	5
3.1 Finite Difference Discretization . . . . .	6
3.1.1 Consistency . . . . .	7
3.2 Ordinary Differential Equation Solution . . . . .	9
3.2.1 Stability . . . . .	12
3.2.2 Consistency . . . . .	13
3.2.3 Convergence . . . . .	13
4. BOUNDARY CONDITIONS . . . . .	13
5. A NUMERICAL EXAMPLE . . . . .	18
6. CONCLUSIONS . . . . .	22
7. REFERENCES . . . . .	23
8. APPENDIX-COMPUTER PROGRAMS STRUCTURE AND COMPUTER LISTING . . . . .	25



A

A NUMERICAL TREATMENT OF THE DYNAMIC MOTION OF A  
ZERO BENDING RIGIDITY CYLINDER IN A VISCOUS STREAM

1. INTRODUCTION

Paidoussis<sup>1</sup> worked out a solution to the dynamic motion problem. Ortloff and Ives<sup>2</sup> studied a special case of the same problem and expressed their solution in the form of an infinite series involving Gamma and Bessel functions. Both the orders and the arguments of Bessel functions are generally complex and can be large in magnitude. Furthermore, evaluation of a Bessel function of complex order is laborious and time-consuming, and accuracy cannot be assured. When the solution proposed by Ortloff and Ives is applied to the nonhomogenous problem where the "upstream" end of the cylinder is forced, a harmonic time dependence is assumed; this means that "forcing" the system by an arbitrary time function will require multiple solutions combined in the Fourier sense.

To overcome these difficulties, a general purpose numerical approach is introduced. This approach discretizes  $\eta_\xi$ ,  $\eta_{\xi r}$ , and  $\eta_{\xi\xi}$  by backward and central differences. This discretization brings the dynamic motion equation into a system of second order ordinary differential equations. This system is decomposed into a system of first order ordinary differential equations. A feasible numerical ordinary differential equation method is then used to solve this system with optimal efficiency.

There are many advantages to using a numerical method to solve the problem of dynamic motion. The theory is well developed with respect to consistency, stability, and convergence. Numerical methods are systematic to implement,

TR 6343A

and effective techniques can be used to accurately accelerate computations. When a numerical approach is used, the laborious evaluation of special functions is bypassed, maximizing accuracy and efficiency.

This report begins with a description of the dynamic motion problem and the associated initial and boundary conditions. A numerical approach is introduced and the supporting theory and mathematical formulation are discussed. An example is given to demonstrate the validity of our numerical solution to a well posed dynamic motion problem. The computer programs are included in the appendix.

## 2. POSED PROBLEMS

The motion of a wire suspended in a fluid stream, considered by Ortloff and Ives,<sup>2</sup> can be described mathematically by the partial differential equation,

$$\begin{aligned}
 EI \frac{\partial^4 y}{\partial x^4} + (M + m) \frac{\partial^2 y}{\partial t^2} + MU^2 \frac{\partial^2 y}{\partial x^2} + 2MU \frac{\partial^2 y}{\partial t \partial x} \\
 - \frac{\partial}{\partial x} \left[ \frac{C_T}{2} \frac{M}{D} U^2 (L-x) \frac{\partial y}{\partial x} \right] \\
 + \frac{1}{2} C_N \frac{M}{D} U \frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x} = 0,
 \end{aligned} \tag{2.1}$$

where

$EI$  = bending rigidity,

$M$  = lateral virtual mass of fluid per unit length of wire accelerated by the accelerating wire,

$m$  = mass of the wire per unit length,

$U$  = velocity of the free stream,

$C_T$  = drag coefficient due to pressure acting on the wire surface,

$D$  = wire diameter,

$L$  = total wire length, and

$C_N$  = drag coefficient due to shear forces acting on the wire surface.

The special case of a zero bending rigidity (or an infinitely flexible) cylinder is realized by setting  $EI = 0$ . To express the above equation in dimensionless terms, use

$$\begin{aligned} \tau &= (t/L)U, & \beta &= M/(M+m), & \xi &= x/L, \\ \epsilon &= L/D, & n &= y/L. \end{aligned}$$

The above equation becomes

$$\begin{aligned} \frac{\partial^2 n}{\partial \tau^2} + \frac{\partial^2 n}{\partial \xi^2} &+ \beta \left[ 1 - \frac{1}{2} C_T \epsilon (1 - \xi) \right] + \frac{\partial n}{\partial \xi} \frac{(C_T + C_N)}{2} \epsilon \beta \\ &+ 2\beta \frac{\partial^2 n}{\partial \xi \partial \tau} + \frac{1}{2} C_N \epsilon \beta \frac{\partial n}{\partial \tau} = 0. \end{aligned} \quad (2.2)$$

The associated initial boundary conditions are described by

$$n = 0 \quad \xi = 0 \text{ (fixed end condition);} \quad (2.3a)$$

$$|n| \text{ is finite,} \quad \xi = 1 \text{ (bounded free end deflection);} \quad (2.3b)$$

$$n = \eta_1(\xi) \quad \tau = 0 \text{ (prescribed initial deflection);} \quad (2.3c)$$

and

$$\left. \frac{\partial n}{\partial \tau} \right|_{\tau=0} = 0, \quad \tau = 0 \text{ (zero initial velocity).} \quad (2.3d)$$

Ortloff and Ives solved the problem posed by equation (2.2) using conditions described in equation (2.3). Their solution is expressed in terms of Bessel functions.

The initial boundary value problem, equation (2.3), for the partial differential equation (2.2) is said to be well posed in the sense of Hadamard<sup>3</sup> if and only if its solution exists, is unique, and depends continuously on the data assigned. After the problem is formulated using finite difference and ordinary differential equations, it will be seen that the problem is well posed. We will seek a unique solution by means of the numerical techniques presented in the next section. When the boundary conditions become uncertain, there is not enough information available to solve equation (2.2); we term this problem ill posed. However, a derived boundary condition is developed, which is shown to be adequate for our problem.

### 3. THE NUMERICAL TREATMENT

In search for a general purpose, accurate solution to the well posed problem (2.2), subject to conditions described in equation (2.3), the method of attack is to discretize  $u_{\xi\xi}$ ,  $u_{\xi\tau}$ , and  $u_{\xi}$  by central and backward finite differences and then to transform equation (2.2) into a system of second order ordinary differential equations (known as the method of lines<sup>4</sup>). We discovered that Generalized Adams Bashforth (GAB) methods<sup>5,6</sup> can be used to solve this system efficiently.

Expressing equation (2.2) in short form and writing  $u$  as  $\eta$  gives

$$u_{\tau\tau} + a(\xi) u_{\xi\xi} + bu_{\xi} + 2B(u_{\xi})_{\tau} + cu_{\tau} = 0, \quad (3.1)$$

where

$$\begin{aligned} a(\xi) &= \beta \left[ 1 - \frac{1}{2} C_T \epsilon (1 - \xi) \right], \\ b &= \frac{1}{2} (C_T + C_N) \epsilon \beta, \text{ and} \\ c &= \frac{1}{2} \beta C_T \epsilon. \end{aligned}$$

### 3.1 FINITE DIFFERENCE DISCRETIZATION

Applying the second order central and backward finite difference discretization to equation (3.1) in the  $\xi$  direction, we obtain

$$\begin{aligned} (u_m)_{\tau\tau} + a(\xi_m) \frac{u_{m+1} - 2u_m + u_{m-1}}{h^2} + \frac{b}{h} (u_m - u_{m-1}) \\ + \frac{2\beta}{h} (u_m - u_{m-1})_{\tau} + c(u_m)_{\tau} = 0, \end{aligned} \quad (3.2)$$

where  $h = \Delta\xi$  for index  $m = 1, 2, \dots$

A simplification of equation (3.2) gives

$$\begin{aligned} (u_m)_{\tau\tau} + \left( \frac{2\beta}{h} + c \right) (u_m)_{\tau} - \frac{2\beta}{h} (u_{m-1})_{\tau} + \frac{a(\xi_m)}{h^2} u_{m+1} \\ + \left[ \frac{b}{h} - \frac{2a(\xi_m)}{h^2} \right] u_m - \left( \frac{b}{h} - \frac{a(\xi_m)}{h^2} \right) u_{m-1} = 0. \end{aligned} \quad (3.3)$$

Equation (3.3) is a difference equation, representing a system of second order ordinary differential equations and is an approximate equation to equation (2.2).

### 3.1.1 Consistency

Before we apply the GAB method to equation (3.3), let us examine the consistency of our finite difference operator  $\mathcal{L}_h [u;h]$ . First, expressing equation (2.2) in true operator form, we obtain

$$\mathcal{L} [u] = \left( \frac{\partial^2}{\partial \tau^2} + a(\xi) \frac{\partial^2}{\partial \xi^2} + b \frac{\partial}{\partial \xi} + 2\beta \frac{\partial^2}{\partial \xi \partial \tau} + c \frac{\partial}{\partial \tau} \right) u = 0. \quad (3.4)$$

Next, expanding  $u_{m+1}$  and  $u_{m-1}$  in powers of  $h$  and keeping the first two principal terms, we obtain

$$u_{m+1} = u_m + h (u_m)' + \frac{h^2}{2} (u_m)'' + \dots, \text{ and}$$

$$u_{m-1} = u_m - h(u_m)' + \frac{h^2}{2} (u_m)'' - \dots$$

Therefore,

$$\frac{u_{m+1} - 2u_m + u_{m-1}}{h^2} \cong (u_m)_{\xi\xi} + \frac{1}{12} h^2 (u_m)_{\xi\xi\xi\xi} \quad (3.5)$$

and

$$\frac{u_m - u_{m-1}}{h} = (u_m)_\xi - \frac{h^2}{2} (u_m)_{\xi\xi}. \quad (3.6)$$

Substituting the power expansions of  $u_{m+1}$  and  $u_{m-1}$  into equation (3.2) and using equations (3.5) and (3.6), we find that equation (3.2) is then expressed in a difference operator form,

$$\begin{aligned} \mathcal{L}_h[u;h] &= \left( \frac{\partial^2}{\partial \tau^2} + a(\xi) \left[ \frac{\partial^2}{\partial \xi^2} + \frac{h^2}{12} \frac{\partial^4}{\partial \xi^4} \right] \right. \\ &+ b \left[ \frac{\partial}{\partial \xi} - \frac{h}{2} \frac{\partial^2}{\partial \xi^2} \right] + 2b \frac{\partial}{\partial \tau} \left[ \frac{\partial}{\partial \xi} - \frac{h^2}{2} \frac{\partial^2}{\partial \xi^2} \right] + c \frac{\partial}{\partial \tau} \left. \right) u = 0 \end{aligned} \quad (3.7)$$

$$\mathcal{L}[u] - \mathcal{L}_h[u;h]$$

$$= \left( -a(\xi) \frac{h^2}{12} \frac{\partial^4}{\partial \xi^4} + \frac{h}{2} b \frac{\partial^2}{\partial \xi^2} + 2b \frac{h^2}{2} \frac{\partial^2}{\partial \xi^2} \right) u.$$

It is seen that

$$\lim_{h \rightarrow 0} (\mathcal{L}[u] - \mathcal{L}_h[u;h]) \rightarrow 0.$$

Therefore, the difference operator is consistent with the true operator in the sense of Keller<sup>7</sup>. Thus, the consistency requirement is established.

Now that we have a consistent difference operator, we seek stable numerical ordinary differential equation methods to solve equation (3.2).

### 3.2 ORDINARY DIFFERENTIAL EQUATION SOLUTION

To seek the solution to equation (3.2), refer to equation (3.3).

Write  $\frac{\partial u_m}{\partial \tau} = w_m$

$$\frac{\partial^2 u_m}{\partial \tau^2} = w'_m = - \left( \frac{2\beta}{h} + c \right) w_m + \frac{2\beta}{h} w_{m-1} - \frac{a(\xi_m)}{h^2} u_{m+1} - \left[ \frac{b}{h} - \frac{2a(\xi_m)}{h^2} \right] u_m + \left( \frac{b}{h} - \frac{a(\xi_m)}{h^2} \right) u_{m-1} \quad (3.8)$$

Equation (3.8) is a set of equations that represent equation (3.3) as a system of first order ordinary differential equations. For illustration, using  $m = 1, 2$ , we can obtain

$$\frac{du_1}{d\tau} = w_1$$

$$\frac{dw_1}{d\tau} = - \frac{2\beta}{h} + c w_1 + \frac{2\beta}{h} w_0 - \frac{a(\xi_1)}{h^2} u_2 - \left[ \frac{b}{h} - \frac{2a(\xi_1)}{h^2} \right] u_1 + \left( \frac{b}{h} - \frac{a(\xi_1)}{h^2} \right) u_0$$

$$\frac{du_2}{d\tau} = w_2$$

$$\frac{dw_2}{d\tau} = -\left(\frac{2\beta}{h} + c\right) w_2 + \frac{2\beta}{h} w_1 - \frac{a(\xi_2)}{h^2} u_3 - \left[\frac{b}{h} - \frac{2a(\xi_2)}{h^2}\right] u_2 + \left(\frac{b}{h} - \frac{a(\xi_2)}{h^2}\right) u_1.$$

In matrix form, the equation becomes

$$\begin{bmatrix} \frac{du_1}{d\tau} \\ \frac{dw_1}{d\tau} \\ \frac{du_2}{d\tau} \\ \frac{dw_2}{d\tau} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\left[\frac{b}{h} - \frac{2a(\xi_1)}{h^2}\right] & -\left(\frac{2\beta}{h} + c\right) & -\frac{a(\xi_1)}{h^2} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{b}{h} - \frac{a(\xi_2)}{h^2} & \frac{2\beta}{h} & -\left[\frac{b}{h} - \frac{2a(\xi_2)}{h^2}\right] & -\left(\frac{2\beta}{h} + c\right) \end{bmatrix} \begin{bmatrix} u_1 \\ w_1 \\ u_2 \\ w_2 \end{bmatrix} + [g] \quad (3.9)$$

where

$$[g] = \left\{ 0, \frac{2\beta}{h} w_0 + \left(\frac{b}{h} - \frac{a(\xi_1)}{h^2}\right) u_0, 0, -\frac{a(\xi_2)}{h^2} u_3 \right\}^T,$$

which is in the form

$$u' = A(\xi)u + g(\xi, \tau, u). \quad (3.10)$$

The elements contained in the components of the g-vector have the following meanings.

$u_0$  = top boundary fixed end condition:  $u_0 = 0, \xi = 0$ ;

$w_0$  = initial condition:  $u_r = 0, r = 0$ ; and

$u_3$  = bottom boundary bounded free-end deflection condition:

$|u_3|$  is finite at  $\xi = 1$ .

The matrix elements,  $A_{ij}$ , of matrix A can be determined by the following setups in which we define  $A_{i,i-j} = 0$  if  $i-j \leq 0$  for  $j = 1, 2, 3$ .

When index  $i$  is odd,  $A_{i,i+1} = 1$ . When index  $i$  is even,

$$A_{i,i-3} = \frac{b}{h} - \frac{a(\xi)}{h^2},$$

$$A_{i,i-2} = \frac{2\beta}{h},$$

$$A_{i,i-1} = \frac{2a(\xi)}{h^2} - \frac{b}{h},$$

$$A_{i,i} = -\left(\frac{2\beta}{h} + c\right)$$

where  $a(\xi)$  is evaluated at  $a(\xi_l), l = \frac{i}{2}$ .

Now, the problem is to select an effective numerical ordinary differential equation method to solve equation (3.8). A close examination suggests that the Generalized Adams-Bashforth (GAB) method offers an efficient solution. In

the present application, because a low order GAB method can do the job, high order GAB methods are not necessary; hence, first order GAB methods were developed into computer programs in FORTRAN language. However, the program package is flexible so that high order GAB methods can be incorporated when required.

We introduce the first order GAB

$$u^{n+1} = e^{Ah} u^n + h\phi_{1,0}(Ah) g_n \quad (3.11)$$

to solve equation (3.10), where

$$\phi_{1,0}(Ah) = -(Ah)^{-1} (I - e^{Ah}). \quad (3.12)$$

The theory with respect to consistency, stability, and convergence has been very well developed for Nonlinear Multistep (NLMS) methods.<sup>8</sup> The GAB method is a member of the NLMS family. We summarize the theory below.

NLMS methods take the expression.

$$\sum_{i=0}^k \alpha_i e^{Ah(k-i)} u_{n+i} = h \sum_{i=0}^k \phi_{ki}(Ah) g_{n+i}. \quad (3.13)$$

### 3.2.1 Stability

The characteristic polynomial of NLMS is defined by

$$\rho(\lambda, s) = e^{\lambda Ah} \sum_{i=0}^k \alpha_i s^i. \quad (3.14)$$

Using the GAB method, the selection of  $\alpha_i$  is such that

$\alpha_k = 1$ ,  $\alpha_{k-1} = -1$ . We see that the root of  $\rho(\lambda, \tau)$  has unity and is simple; therefore, method (3.11) is stable.

### 3.2.2 Consistency

The GAB method, equation (3.14) satisfies the consistency condition

$$\lim_{h \rightarrow 0} \left\| \sum_{i=0}^k \alpha_i e^{Ah(k-i)} u_{n+i} - h \sum_{i=0}^k \phi_{ki}(Ah) g_{n+i} \right\| = 0 \quad (3.15)$$

for  $k = 1$ ,  $\alpha_k = 1$  and  $\alpha_{k-1} = -1$ . Therefore, GAB method is consistent.

### 3.2.3 Convergence

According to the convergence theorem of NLMS methods, "A stable and consistent NLMS method is convergent." Therefore the GAB method applied to problem (3.10) is a convergent method.

## 4. BOUNDARY CONDITIONS

In real applications, at  $\xi = 1$ , the bounded free end deflexion boundary condition is expected to be such that  $n(1, \tau)$  is finite. However, the appropriate function  $n(1, \tau)$  to be used for the boundary condition appears uncertain in reality. This lack of information defines problem (2.2) as an ill posed problem. For general partial differential equations, it is always difficult to formulate correct conditions leading to a well posed problem. Problems may look reasonable, yet cannot be solved.<sup>3</sup> It is hoped that the

bounded free end deflexion boundary condition may be obtained through experimentation, but the exact mathematical expression for  $\eta(1, \tau)$  must still be worked out. We will attempt to change the ill posed problem to a well posed one so that a solution exists and can be solved by the numerical techniques we have developed.

In the theory of second order partial differential equations there exists a class of well posed problems, such as the Cauchy problem for wave equations, the Dirichlet condition for Laplace equations, and the mixed initial boundary value problem for heat equations. Our first step is to examine the most general boundary conditions. Let  $u_N$  denote the normal derivative. The first boundary value problem of the Dirichlet type indicates

$$u = f \quad (4.1)$$

on the boundary. The second boundary value problem of the Neumann type indicates

$$u_N = f \quad (4.2)$$

on the boundary. The third boundary value problem of the mixed type indicates

$$u_N + \alpha u = f \quad (4.3)$$

on the boundary. Note that the third boundary value problem is well posed only for the restricted choice of  $\alpha$ . We will assume that the free end deflexion boundary condition takes the expression

$$\lambda u_N + \alpha u = f. \quad (4.4)$$

When

- $\lambda = 0, \alpha = 1$ , (4.4) reduces to (4.1);
- $\lambda = 1, \alpha = 0$ , (4.4) reduces to (4.2); and
- $\lambda = 1, \alpha$  arbitrary, (4.4) reduces to (4.3).

In our application, as given by the numerical example in the next section,

$$n(\xi, \tau)_N = n(\xi, \tau), \quad \lambda = 0, \alpha = 1 \text{ gives}$$

$n(\xi, \tau) = f(\xi, \tau)$  and  $\{f(1, \tau)\}$  is finite. This gives Ortloff's and Ives' bounded free end deflexion boundary condition.

The procedure to be followed here for determining a free end boundary condition is to derive an approximate boundary condition and then to use that boundary condition to compare the solution with a direct application of the Ortloff and Ives solution.<sup>2</sup> We develop a form of the boundary condition for the second order partial differential equation by following the approach used

by Paidoussis for his fourth order partial differential equation; that is, by integrating the transverse momentum equation over a short tapered end which is attached to the free end in order to generate the required boundary condition. Paidoussis assumed that the cross sectional area tapers smoothly from  $S$  to zero in a distance ( $l$ ) sufficiently short that the forces acting on the tapered end can be lumped and considered in appropriate boundary conditions. For our present problem the boundary condition is obtained from

$$\begin{aligned}
 f \int_{L-l}^L \left( \frac{\partial}{\partial t} + U \frac{\partial}{\partial x} \right) [M(x)v] dx + \int_{L-l}^L F_N dx - \int_{L-l}^L \frac{\partial}{\partial x} \left( T(x) \frac{\partial y}{\partial x} \right) dx \\
 + \int_{L-l}^L m(x) \frac{\partial^2 y}{\partial t^2} dx = 0
 \end{aligned} \tag{4.5}$$

where the parts of the equation express rate of change of fluid momentum, hydrodynamic forces, and cylinder inertia, respectively, and where  $f$  is a factor introduced by Paidoussis to account for the intractable flow conditions at the free end and  $v$  is the transverse velocity of the fluid relative to the cylinder. Therefore,

$$v = \frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x}, \tag{4.6}$$

$$F_N = \frac{1}{2} \left( \frac{M}{D} \right) U C_N V, \quad (4.7)$$

$$T(x) = T(L) + \frac{1}{2} \left( \frac{M}{D} \right) U^2 C_T (L-x); \quad (4.8)$$

and  $T(L)$  is a consequence of form drag at the free end.

An important assumption necessary to perform the integration is that the length of the tapered section ( $l$ ) is small enough that the lateral velocity ( $V$ ) may be considered constant over  $l$ . We find

$$\begin{aligned} & fM \frac{l}{2} \left( \frac{\partial^2 y}{\partial t^2} + U \frac{\partial^2 y}{\partial x \partial t} \right) + fUM \left( \frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x} \right) \\ & + \frac{UC_N M}{4} \frac{l}{D} \left( \frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x} \right) + \frac{1}{2} \left( \frac{M}{D} \right) U^2 C_T l \left( \frac{\partial y}{\partial x} \right) \\ & - T(L) l \frac{\partial^2 y}{\partial x^2} + \frac{m}{2} l \frac{\partial^2 y}{\partial t^2} + O(l^2) = 0 \end{aligned} \quad (4.9)$$

for  $x = L$ , all  $t$ .

After nondimension of this equation as before and neglecting terms of order  $(l^2)$  and  $\frac{l}{L}$ , we have

$$\left[ f + \frac{C_N}{4} \left( \frac{l}{D} \right) \right] \frac{\partial n}{\partial \tau} + \left[ f + \frac{C_N}{4} \left( \frac{l}{D} \right) + \frac{C_T}{2} \left( \frac{l}{D} \right) \right] \frac{\partial n}{\partial \xi} = 0 \quad (4.10)$$

for  $\xi = 1$ , all  $\tau$

On physical grounds it is reasonable to neglect  $C_T$  relative to  $C_N/4$ ,<sup>9</sup> making the final boundary condition

$$\frac{\partial \eta}{\partial \xi} + \frac{\partial \eta}{\partial \tau} = 0 \text{ for } \xi = 1, \text{ all } \tau, \quad (4.11)$$

which amounts to a "radiation condition"; that is, no reflected energy exists. In the following sections, we refer to boundary condition (4.11) as Kennedy boundary condition.

## 5. A NUMERICAL EXAMPLE

The test example is obtained through linearization of a fourth order nonlinear cable equation,<sup>10</sup> which is given by

$$m \frac{\partial^2 y}{\partial t^2} + M \left( \frac{\partial}{\partial t} + U \frac{\partial}{\partial x} \right)^2 y = \frac{\partial}{\partial x} \left( T \frac{\partial y}{\partial x} \right) - \bar{C}_N \left( \frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x} \right) \quad (5.1)$$

where  $m$ ,  $M$ ,  $U$  take the same definitions as given in section 2.  $T$ ,  $\bar{C}_N$  are defined as

$$T = \frac{T_1}{L} (L - x), \quad (5.2)$$

$$T_1 = \bar{C}_T \pi D U^2 L, \quad (5.3)$$

$$\bar{C}_T = \frac{1}{2} \frac{1}{\pi} \frac{1}{D^2} C_T, \text{ and} \quad (5.4)$$

$$\bar{C}_N = \frac{1}{2} \frac{M}{D} C_N U, \quad (5.5)$$

where  $C_N$  and  $C_T$  satisfy the definitions given in section 2.

Assuming  $\frac{\partial}{\partial x}$  and  $\frac{\partial}{\partial t}$  commute, expanding equation (5.1) gives

$$m \frac{\partial^2 y}{\partial t^2} + M \left\{ \frac{\partial^2 y}{\partial t^2} + 2U \frac{\partial^2 y}{\partial t \partial x} + U^2 \frac{\partial^2 y}{\partial x^2} \right\} = T \frac{\partial^2 y}{\partial x^2} + \frac{\partial T}{\partial x} \frac{\partial y}{\partial x} - \bar{C}_N \left( \frac{\partial y}{\partial t} + U \frac{\partial y}{\partial x} \right) \quad (5.6)$$

Performing  $\frac{\partial T}{\partial x}$  and using definitions (5.2) through (5.6), we get

$$\frac{\partial^2 y}{\partial t^2} \frac{M+m}{M} + \frac{\partial^2 y}{\partial x^2} \left[ U^2 - \frac{1}{2} C_T \frac{U^2}{D} (L-X) \right] + \frac{\partial y}{\partial x} \frac{U^2}{2D} (C_T + C_N) + 2U \frac{\partial^2 y}{\partial x \partial t} + \frac{1}{2} C_N \frac{U}{D} \frac{\partial y}{\partial t} = 0. \quad (5.7)$$

Equation (5.7) is the same as Ortluff's equation (2.1) before nondimensionalization.<sup>2</sup>

Select

$$D = \frac{1}{2} \text{ inch},$$

$$M = m = 0.00273$$

$$U = 15 \text{ ft/sec}$$

TR 6343A

$$L = 2000 \text{ ft}$$

$$C_T = \frac{1.8}{35}$$

$$C_N = 1.1259.$$

Then,

$$\epsilon = 48,000$$

$$\beta = \frac{1}{2}.$$

The solution to equation (5.7) is expressed by

$$y(t, x) = e^{i\omega t} J_\gamma(x),$$

where  $\gamma$  is approximately 21.89 and  $J_\gamma(x)$  initial values are calculated using a UNIVAC 1108 Bessel function subroutine.<sup>10</sup>

The fixed end boundary condition initially is zero. The free end boundary condition uses  $n(1, \tau) = e^{i\omega\tau} J_\gamma(1)$ .

This problem was tested again using Kennedy's free end boundary condition. Results are surprisingly in agreement with the known solution. The test results seem to show that the Kennedy free end boundary condition is adequate. Results are presented in graphic form. Two sets of graphs are given: one displays  $|n(\xi, \tau)|$  versus  $\tau$ , the other displays the real  $\{n(\xi, \tau)\}$  versus  $\tau$ . Both plots are constructed at  $\xi = 0.2, 0.8$ .

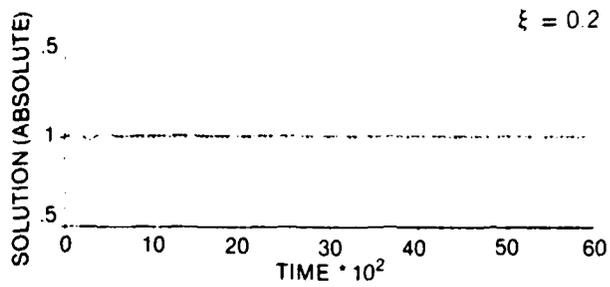


Figure 1: Solution magnitude vs time

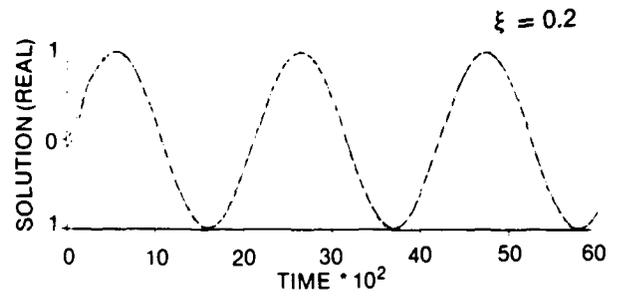


Figure 2: Real part solution vs time

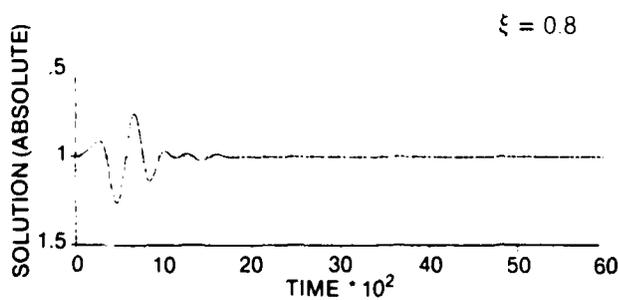


Figure 3: Solution magnitude vs time

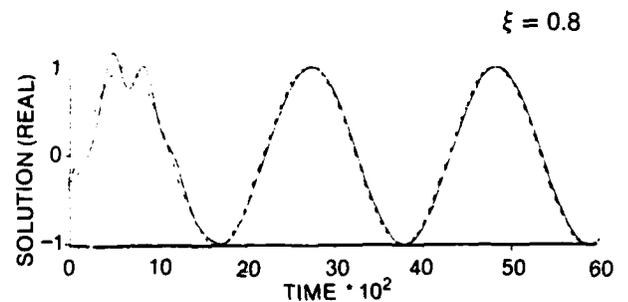


Figure 4: Real part solution vs time

## 6. CONCLUSIONS

A numerical solution to the dynamic motion of a zero bending rigidity cylinder in a viscous stream has been introduced. The numerical procedures developed to obtain the solution are theoretically convergent and computationally accurate.

For given appropriate boundary conditions and accurate initial values, this model will produce an accurate unique solution. For uncertain boundary conditions, this model can be used as a tool to study the boundary effects and possible to construct the ad hoc boundary conditions.

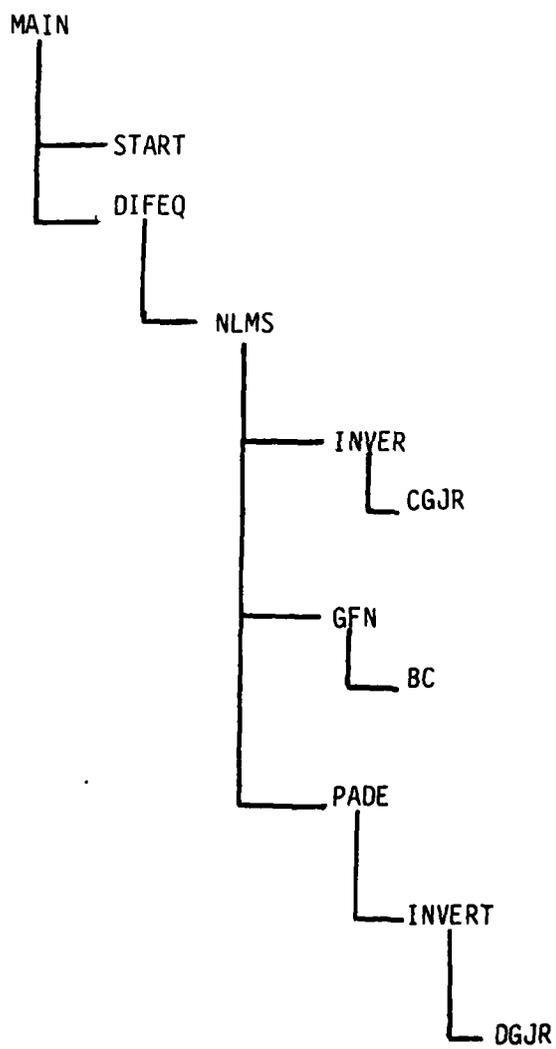
## REFERENCES

1. M. P. Paidoussis, "Dynamics of Flexible Slender Cylinders in Axial Flow," Journal of Fluid Mechanics, vol. 26, part 4, 1966, pp. 717-751
2. C. R. Ortloff and J. Ives, "On the Dynamic Motion of a Thin Flexible Cylinder in a Viscous Stream," Journal of Fluid Mechanics, vol. 38, part 4, 1969, pp. 713-720.
3. P. R. Garabedian, Partial Differential Equations, John Wiley & Sons, Inc., NY, 1964.
4. W. F. Ames, Numerical Methods for Partial Differential Equations, Academic Press, NY, 1977.
5. D. Lee and S. Preiser, "A Class of Nonlinear Multistep A-Stable Numerical Methods for Solving Stiff Differential Equations," Journal of Computers and Mathematics With Applications, vol. 4, no. 1, 1978, pp. 43-51.
6. D. Lee, Generalized Adams Methods, NUSC Technical Report No. 6011, Naval Underwater Systems Center, New London, CT, 1979.
7. H. B. Keller, Numerical Methods for Two-Point Boundary-Value Problem, Blaisdell Publishing Company, Waltham, Massachusetts, 1968.
8. D. Lee, Nonlinear Multistep Methods for Solving Initial Value Problems in Ordinary Differential Equations, NUSC Technical Report No. 4774, Naval Underwater Systems Center, New London, CT, 24 May 1974.
9. S. F. Hoerner, "Fluid-Dynamic Drag," published by the author, 1965.
10. J. Patel, personal communication, 1979.

APPENDIX

COMPUTER PROGRAMS STRUCTURE AND COMPUTER LISTING

COMPUTER PROGRAM STRUCTURE



TR 6343A

ACRONYMS

MAIN    main program which controls the setup of inputs and the preparation of outputs

START    supplies the initial values

DIFEQ    controls the present  $\tau$ -step and calls for NLMS(GAB) method

NLMS    1st order Generalized Adams-Bashforth method

INVER    calls for complex matrix inversion

CGJR    complex matrix inversion using Gauss-Jordan reduction

GFN    calculates the g-vector

BC    fixed end and free end boundary conditions

PADE    a rational function approximation for matrix exponentials

INVERT    calls for double precision matrix inversion

DGJR    double precision matrix inversion using Gauss-Jordan reduction

The user needs to deal with MAIN, START, GFN, and BC. The user need not be concerned with DIFEQ, NLMS, INVER, INVERT, CGJR, and DGJR.

## COMPUTER PROGRAM LISTING

**MAIN**

```

COMMON PEPS,PCT,PCN,PBETA,FQ,PA,PB,PC,PD,DZ,IBND
COMMON A(18,18),T(3)
DIMENSION Y(2,18),YZERO(18),YNEW(18),EXACT(18)
COMPLEX A,Y,YZERO,YNEW,XX,OMGA,SAVE,EXACT
C ***** THIS PACKAGE SOLVED A 2ND ORDER P.D.E. BY THE METHOD OF LINES AND
D
C          GENERALIZED ADAMS-BASHFORTH METHODS
C ***** REFERENCE: ORTLOFF AND IVES
C ***** INPUT PARAMETERS HAVE THE FOLLOWING DEFINITIONS:
C ***** N          = NUMBER OF 2ND ORDER ODE
C ***** TMAX       = MAXIMUM TAO
C ***** TINT       = EVERY TAO INTERVAL TO BE PRINTED OUT
C ***** PXI        = AT THIS XI, THE OUTPUT IS REQUESTED
C ***** FQ         = FREQUENCY, NONDIMENSIONAL OMEGA
C ***** IBND       = BOUNDARY CONDITION INDICATOR
C                   = 1 BUILT-IN KENNEDY BOUNDARY CONDITION
C                   = 2 USER-SUPPLIED BOUNDARY CONDITION
C ***** PEPS      = EPSILON
C ***** PCT       = C SUB T
C ***** PCN       = C SUB N
C ***** PBETA     = BETA
C ***** H         = TAO STEP SIZE
C ***** READ INPUTS HERE
          READ(5,*) N,TMAX,TINT,PXI,FQ,IBND
          READ(5,*) PEPS,PCT,PCN,PBETA,H
          N=2*(N-1)
          DZ=1.0/(FLOAT(N/2+1))
          PB=0.5*PBETA*PCT*PEPS
          PA=PBETA*(1.0-PB)
          PD=0.5*PCN*PEPS*PBETA
          PC=PB+PD
C ***** TO SET-UP MATRIX A
          DO 20 I=1,N
          DO 20 J=1,N
20  A(I,J)=CMPLX(0.0,0.0)
          BH=PC/DZ
          TBH=2.*PBETA/DZ
          DG=-(TBH+PB)
          DO 28 I=1,N
          IF(I .GE. 4) GO TO 26
          IF(I .EQ. 2) GO TO 25
          A(I,I+1)=CMPLX(1.0,0.0)
          GO TO 28
25  J=I/2
          X=PBETA*(1,-.5*PCT*PEPS*(1.-J*DZ))
          A(I,I-1)=CMPLX(2.*X/(DZ*DZ)-BH,0.0)
          A(I,I)=CMPLX(DG,0.0)
          A(I,I+1)=CMPLX(-X/(DZ*DZ),0.0)
          GO TO 28
26  IF(MOD(I,2) .EQ. 0) GO TO 27
          A(I,I+1)=CMPLX(1.0,0.0)
          GO TO 28

```

TR 6343A

```

27 J=I/2
   X=PBETA*(1.-.5*PDT*PBETA*(1.-J*DZ))
   A(I-I-3)=CMPLX(BH-X/DZ**2,0.0)
   A(I,I-3)=CMPLX(TBH,0.0)
   A(I,I-1)=CMPLX(D.*X**DEADZ,-BH,0.0)
   A(I,I)=CMPLX(BE,0.0)
   IF(I .EQ. N) GO TO 28
   A(I,I+1)=CMPLX(-X.(DZ*DEZ),0.0)
28 CONTINUE
   WRITE(5,*) (A(I,L),L=1,8)
   T(1)=0.0
C ***** TO OBTAIN INITIAL VALUES FROM 'START'
   CALL START(N,T,YZERO)
   TX=0.0
   SAVE=YZERO(N-1)
10 CONTINUE
   TX=TX+TINT
   CALL DIFEQ(H,N,TK,Y,YZERO,Z,YNEW,SAVE)
C ***** RESULTS IN YNEW(I)
C ***** Z CONTAINS PRESENT TAO
C ***** USER USES ABOVE INFORMATION FOR HIS PLOT
   SAVE=YNEW(N-1)
   IPT=IPT+1
   IF(IPT .LE. 4) GO TO 8
   IF(MOD(IPT,1) .EQ. 0) GO TO 8
   GO TO 100
8 CONTINUE
   WRITE(5,1) H,Z
1  FORMAT(/10X,'H =',E15.8,5X,'T =',E15.8/)
2  FORMAT(3X,F8.2,10X,E15.8,10X,E15.8)
C ***** PRINT OUT EXACT SOLUTION
C ***** THIS PORTION IS FOR TEST EXAMPLE ONLY
   CALL START(N,T,EXACT)
   DO 100 I=1,N,2
   NN=(I+1)/2
   DLZ=NN*DZ
   WRITE(5,2) DLZ, YNEW(I)
   WRITE(5,3) EXACT(I)
3  FORMAT(21X,E15.8,10X,E15.8/)
100 CONTINUE
   IF(TX .GE. TMAX) STOP
   GO TO 10
END
```

## NLMS

```

SUBROUTINE NLMS(H,Y,N,YN,IS,SAVE)
COMMON PEPS,PCT,PCN,PBETA,FQ,PA,PB,PC,PD,DZ,IBND
PARAMETER KM=18
COMMON A(KM,KM),T(3)
DIMENSION AH(KM,KM),EAH(KM,KM),G(KM)
DIMENSION P1(KM,KM),UNIT(KM,KM)
DIMENSION Y(2,KM),YN(KM)
DIMENSION FE(KM,KM),A1(KM,KM)
COMPLEX A,AH,EAH,G,P1,PH,UNIT,Y,YN,SAVE
COMPLEX FE,A1
DATA IND/0/
IF(IND.GT. 0) GO TO 14
DO 2 I=1,N
  DO 2 J=1,N
2   AH(I,J)=H*A(I,J)
  IF(N-1) 7,8,7
8   A1(1,1)=1./AH(1,1)
  GO TO 9
7   CALL INVER(AH,N,A1)
9   IF(N-1) 10,11,10
11  EAH(1,1)=CEXP(AH(1,1))
  GO TO 14
10  CALL PADE(A,H,EAH,N)
  IND = IND + 1
14  IF(IS.GT. 1) GO TO 100
  DO 6 I=1,N
    DO 6 J=1,N
      P1(I,J)=CMPLX(0.0,0.0)
      UNIT(I,J)=CMPLX(0.0,0.0)
6   CONTINUE
1   UNIT(I,I)=CMPLX(1.0,0.0)
100 CONTINUE
C *****
C *   NONLINEAR MULTISTEP STARTS HERE. *
C *   BEGINNING SECTION DOES INITIALIZATION *
C *****
DO 132 I=1,N
132 YN(I)=CMPLX(0.0,0.0)
  IF(IS.GT.1) GO TO 131
  DO 103 I=1,N
    DO 103 J=1,N
103 P1(I,J)=-EAH(I,J)+UNIT(I,J)
C *****
C *   1ST ORDER GAB *
C *   DO LOOP 105 CALCULATES PHI(1,0) *
C *   LOOP 108 OR 110 COMPUTES FINAL Y(N+1) *
C *****
DO 106 I=1,N
  DO 106 J=1,N
  PH=CMPLX(0.0,0.0)
  DO 105 K=1,N
105 PH=PH-A1(I,K)*P1(K,J)
  FE(I,J)=PH
106 CONTINUE
131 CALL GEN(G,H,N,Y,1,T,A,SAVE)
  DO 108 I=1,N
    DO 108 J=1,N
108 YN(I)=YN(I)+EAH(I,J)*Y(I,J)+H*FE(I,J)*G(J)
  RETURN

```

TR 6343 A

```
DO 9 I=1,N
DO 10 J=1,N
C(I,J)=AA(I,J)*H/2.0
PP(I,J)=0.DO
10 CONTINUE
C(I,I)=C(I,I)+1.DO
9 CONTINUE
DO 12 I=1,N
DO 13 J=1,N
DO 14 K=1,N
PP(I,J)=PP(I,J)+B(I,K)*C(K,J)
14 CONTINUE
13 CONTINUE
12 CONTINUE
IF(M .EQ. 0) GO TO 40
C *****
C * NORM(AH).GT.(.1), EXP(A) =EXP(A/2**M)**(2**M) *
C *****
DO 24 I=1,N
DO 25 J=1,N
B(I,J)=0.DO
25 CONTINUE
24 CONTINUE
DO 36 K=1,M
DO 27 I=1,N
DO 28 J=1,N
DO 29 L=1,N
B(I,J)=B(I,J)+PP(I,L)*PP(L,J)
29 CONTINUE
28 CONTINUE
27 CONTINUE
DO 31 I=1,N
DO 32 J=1,N
BB=B(I,J)
P(I,J)=CMLX(BB,0.0)
B(I,J)=0.DO
32 CONTINUE
31 CONTINUE
36 CONTINUE
H=HAVE
RETURN
20 H=H/2.0
M=M+1
DO 54 I=1,N
DO 55 J=1,N
PP(I,J)=0.DO
55 CONTINUE
54 CONTINUE
GO TO 30
40 H=HAVE
RETURN
END
```

# START

```

SUBROUTINE START(N,T,YZERO)
COMMON PEPS,PCT,PCN,PBETA,FQ,PA,PB,PC,PD,DZ,IBND
COMPLEX YZERO(1)
DIMENSION T(1)
C ***** USER SHALL REVISE THIS PORTION TO INCORPORATE HIS INITIAL VALUE
S
C ..... YZERO(I)   CONTAINS THE FUNCTION
C ..... YZERO(I+1) CONTAINS THE DERIVATIVE
DO 29 LP=1,N,2
DLZ=DZ*(LP+1)/2
IF(LP.EQ.1) YZERO(LP)=CEXP(CMPLX(0.0,FQ*T(1)))
*
*          *CMPLX(.99546739,-.73021506E-01)
IF(LP.EQ.3) YZERO(LP)=CEXP(CMPLX(0.0,FQ*T(1)))
*
*          *CMPLX(.98602435,-.14547502)
IF(LP.EQ.5) YZERO(LP)=CEXP(CMPLX(0.0,FQ*T(1)))
*
*          *CMPLX(.97172089,-.21705468)
IF(LP.EQ.7) YZERO(LP)=CEXP(CMPLX(0.0,FQ*T(1)))
*
*          *CMPLX(.96262427,-.28745766)
YZERO(LP+1)=YZERO(LP)*CMPLX(0.0,FQ)
29 CONTINUE
RETURN
END

```

# GFN

```

SUBROUTINE GFN(G,H,N,Y,U,T,A,SAVE)
C ***** THIS HANDLES THE G VECTOR
C ***** G VECTOR CONTAINS BOUNDARY INFORMATION
COMMON PEPS,PCT,PCN,PBETA,FQ,PA,PB,PC,PD,DZ,IBND
COMPLEX A(18,18),Y(2,18),G(18),SURF,BOTT,XA,XB,SAVE,WZ
DIMENSION T(1)
DATA PI,ZERO,ONE/3.1415926535,0.0,1.0/
DO 1 I=1,N
1 G(I)=CMPLX(0.0,0.0)
C ***** FIRST ARGUMENT 0, CALLS FOR FIXED END CONDITION
CALL BC(0,N,H,T,Y,SAVE,WZ,SURF)
G(2)=PBETA*(1.0-0.5*PCT*PEPS*(1.0-DZ))
G(2)=2.0*PBETA*WZ/DZ+(PC/DZ-G(2)/DZ**2)*SURF
WRITE(5,*)G(2)
C ***** FIRST ARGUMENT 1, CALLS FOR FREE-END CONDITION
CALL BC(1,N,H,T,Y,SAVE,WZ,BOTT)
X=-PBETA*(1.-.5*PCT*PEPS*(1.-(N/2)*DZ))/(DZ*DZ)
G(N)=CMPLX(X,0.0)*BOTT
WRITE(5,*)G(N)
RETURN
END

```

# INVERT

```

SUBROUTINE INVERT(A,N,ANS)
C *****
C * MATRIX INVERSION SUBROUTINE, CALLED BY PADE OR NLMS *
C * A CONTAINS THE ORIGINAL ELEMENTS AND REMAINS UNALTERED *
C * ANS CONTAINS THE A**(-1) *
C * THIS SET-UP IS USING UNIVAC 1108 MATHPK EXISTING DOUBLE *
C * PRECISION GAUSS-JORDAN REDUCTION *
C * THIS PROGRAM IS REPLACEABLE BY THE USER *
C *****
DOUBLE PRECISION A(18,18),ANS(18,18),V(2)
DIMENSION JC(18)
DATA NR/18/,NC/18/
V(1)=1.DO
DO 1 I=1,N
DO 1 J=1,N
1 ANS(I,J)=A(I,J)
CALL DGJR(ANS,NR,NC,N,N,MDEX,JC,V)
IF(MDEX .EQ. 1) GO TO 10
RETURN
10 WRITE (4,2)
2 FORMAT(3X,22HMATRIX INVERSION ERROR)
RETURN
END

```

# INVER

```

SUBROUTINE INVER(A,N,ANS)
PARAMETER NDIM=18
COMPLEX A(NDIM,NDIM),ANS(NDIM,NDIM),V(2)
DIMENSION JC(NDIM)
DATA NR/NDIM/,NC/NDIM/
V(1)=CPLX(1.0,0.0)
DO 1 I=1,N
DO 1 J=1,N
1 ANS(I,J)=A(I,J)
CALL CGJR(ANS,NC,NR,N,N,MDEX,JC,V)
IF(MDEX .EQ. 1) GO TO 10
RETURN
10 WRITE (4,2)
2 FORMAT(3X,11HMAT INV ERR)
RETURN
END

```

# DGJR

```

SUBROUTINE DGJR(A,NC,NR,N,MC,MDEX,JC,V)
DIMENSION JC(N),V(2)
DIMENSION A(NR,NC)
DOUBLE PRECISION A,V,X,DLOG
IW=V(1)
M=1
S=1.
L=N+(MC-N)*(IW/4)
KD=2- MOD(IW/2,2)
IF(KD.EQ.1) V(2)=0.
KI=2- MOD(IW,2)
GO TO (5,20),KI
5  DO 10 I=1,N
10  JC(I)=I
20  DO 91 I=1,N
    GO TO (22,21),KI
21  M=I
22  IF (I.EQ.N) GO TO 60
    X=-1.
    DO 30 J=I,N
    IF (X.GT.ABS(A(J,I))) GO TO 30
    X=ABS(A(J,I))
30  K=J
    CONTINUE
    IF(K.EQ.I) GO TO 60
    S=-S
    V(1)=-V(1)
    GO TO (35,40),KI
35  MU=JC(I)
    JC(I)=JC(K)
    JC(K)=MU
40  DO 50 J=M,L
    X=A(I,J)
    A(I,J)=A(K,J)
50  A(K,J)=X
60  IF (ABS(A(I,I)).GT.0.) GO TO 70
    IF(KD.EQ.1) V(1)=0.
    JC(1)=I-1
    RETURN
70  GO TO (71,72),KD
71  IF(A(I,I).LT.0.) S=-S
    V(2)=V(2) + DLOG(ABS(A(I,I)))
72  X=A(I,I)
    A(I,I)=1.
    DO 80 J=M,L
    A(I,J)=A(I,J)/X
    CALL ERRST(72,MDEX)
    IF(MDEX.EQ.1) GO TO 150
80  CONTINUE

```

TR 6343A

```
DO 91 K=1,N
IF (K.EQ.I) GO TO 91
X=A(K,I)
A(K,I)=0.
DO 90 J=M,L
A(K,J)=A(K,J)-X*A(I,J)
CALL ERRST(72,MDEX)
IF(MDEX.EQ.1) GO TO 150
90 CONTINUE
91 CONTINUE
GO TO (95,140),KI
95 DO 130 J=1,N
IF(JC(J).EQ.J) GO TO 130
JJ=J+1
DO 100 I=JJ,N
IF(JC(I).EQ.J) GO TO 110
100 CONTINUE
110 JC(I)=JC(J)
DO 120 K=1,N
X=A(K,I)
A(K,I)=A(K,J)
120 A(K,J)=X
130 CONTINUE
140 JC(1)=N
IF(KD.EQ.1) V(1)=S
RETURN
150 JC(1)=1-I
IF(KD.EQ.1) V(1)=S
RETURN
END
```

## CGJR

```

SUBROUTINE CGJR(A,NC,NR,N,MC,IFL,JC,V)
DIMENSION JC(1)
COMPLEX CLOG,V,XC,A(18,18)
COMPLEX Z
INTEGER*2 NERR
NERR=72
IFL=0
IW=V
V=(0.,0.)
IBIT=0
M=1
L=N+(MC-N)*(IW/4)
KD=2-MOD(IW/2,2)
KI=2-MOD(IW,2)
GO TO (5,20),KI
5   DO 10 I=1,N
10  JC(I)=I
20  DO 91 I=1,N
    GO TO (22,21),KI
21  M=I
22  IF (I.EQ.N) GO TO 60
    X=-1.
    DO 30 J=I,N
    ANORM=ABS(REAL(A(J,I)))+ABS(AIMAG(A(J,I)))
    IF(X.GT.ANORM) GO TO 30
    X=ANORM
    K=J
30  CONTINUE
    IF(K.EQ.I) GO TO 60
    IBIT=IBIT+1
    GO TO (35,40),KI
35  MU=JC(I)
    JC(I)=JC(K)
    JC(K)=MU
40  DO 50 J=M,L
    XC=A(I,J)
    A(I,J)=A(K,J)
50  A(K,J)=XC
60  ANORM=ABS(REAL(A(I,I)))+ABS(AIMAG(A(I,I)))
    IF(ANORM.GT.0) GO TO 70
    V=(0.,0.)
    JC(1)=I-1
    RETURN

```

TR 6343A

```
70 GO TO (71,72),KD
71 V=V+CLOG(A(I,I))
   Z=CLOG(A(I,I))
72 XC=A(I,I)
   A(I,I)=(1.,0.)
   DO 80 J=M,L
   A(I,J)=A(I,J)/XC
   CALL ERRST(NERR,IFL)
   IF(IFL.EQ.1) GO TO 150
80 CONTINUE
   DO 91 K=1,N
   IF (K.EQ.I) GO TO 91
   XC=A(K,I)
   A(K,I)=(0.,0.)
   DO 90 J =M,L
   A(K,J)=A(K,J)-XC*A(I,J)
   CALL ERRST(NERR,IFL)
   IF(IFL.EQ.1) GO TO 150
90 CONTINUE
91 CONTINUE
   GO TO (95,140),KI
95 DO 130 J=1,N
   IF (JC(J).EQ.J) GO TO 130
   JJ=J+1
   DO 100 I=JJ,N
   IF (JC(I).EQ.J) GO TO 110
100 CONTINUE
110 JC(I)=JC(J)
   DO 120 K=1,N
   XC=A(K,I)
   A(K,I)=A(K,J)
120 A(K,J)=XC
130 CONTINUE
140 JC(1)=N
   V=V+(0.,3.14159265)*CMPLX(FLOAT(MOD(IBIT,2)),0.)
   RETURN
150 JC(1)=1-I
   RETURN
   END
```

## BC

```

SUBROUTINE BC(D,N,H,T,SAVE,IX,0)
C ***** THIS SUBROUTINE SUPPLIES THE FREE-END BOUNDARY CONDITION
C ***** USER HAS AN OPTION TO SUPPLY HIS BOUNDARY CONDITION
C ***** IN THIS CASE, STATEMENT AFTER 2 SHOULD BE REPLACED BY USER'S
C ***** CONDITION
COMMON PEPS,PCT,PCN,PBETA,FQ,PA,PB,PC,PD,DZ,IBND
DIMENSION Y(1)
COMPLEX Y(2,18),X,NX
IF(LD .GT. 0) GO TO 10
C ***** FIXED END BOUNDARY CONDITION
X=DEXP(CMPLX(0.0,FQ*T(1)))
X=CMPLX(0.0,FQ)*X
RETURN
C ***** CONTINUE
C ***** TO 1,2,3,IBND
C ***** P.N. DERIVED BOUNDARY CONDITION
C ***** *****
C +      Q +P =0      A
C +      T      Z      K
C ***** *****
C ***** CONTINUE
X=(H*(1,N-1)/DZ+SAVE)/(1.0+H/DZ)
RETURN
C ***** USER SUPPLIED BOUNDARY CONDITION
2 CONTINUE
X=DEXP(CMPLX(0.0,FQ*T(1))) * CMPLX(.92986198,-.33670799)
X=CMPLX(1.0,0.0)
RETURN
3 CONTINUE
IF(PBETA .NE. 1.0) GO TO 31
X=Y(1,N-1)
RETURN
31 XYZ=1.+(PCT*PEPS*PBETA*DE)/(2.*(1.-PBETA))
X=(CMPLX(1./XYZ,0.0))*(CMPLX(1.+XYZ,0.0)*Y(1,N-1)-Y(1,N-3))
RETURN
END

```

## DIFEQ

```

SUBROUTINE DIFEQ(H,N,TMAX,Y,YZERO,ANORM,YN,SAVE)
COMMON PEPS,PCT,PCN,FBETA,FQ,PA,FB,PC,PD,DZ,IBND
COMMON A(18,18),T(3)
COMPLEX Y(2,18),YN(18),YOLD(2,18),YZERO(18),A,SAVE
DO 40 I=1,N
40 Y(1,I)=YZERO(I)
   IH=0
   TZERO=T(1)
60 TEA=T(1)+H
   IF(TEA.GT.TMAX) H=TMAX-T(1)
   IF(TEA.GT.TMAX) IH=0
   IF(TEA.GT.TMAX) GO TO 60
   IH=IH+1
   IF(IH .GE. 32767) IH=2
   T(2)=T(1)+H
   IMP=2
   DO 62 J=1,IMP
   DO 62 I=1,N
62 YOLD(J,I)=Y(J,I)
   CALL NLMS(H,YOLD,N,YN,IH,SAVE)
59 DO 66 I=1,N
   Y(2,I)=YN(I)
66 YOLD(2,I)=YN(I)
   *****
   *   RESULTS Y(TEA) IN YN(I) AND Y(2,I)   *
   *****
ANORM=TEA
DO 85 I=1,N
Y(1,I)=Y(2,I)
YZERO(I)=Y(1,I)
85 CONTINUE
T(1)=T(2)
TZERO=T(1)
IF(ABS(TEA-TMAX).LE.(.1E-5)) RETURN
GO TO 60
END

```

# PADE

```

SUBROUTINE PADE(A,H,P,N)
C ***** A RATIONAL APPROXIMATION OF MATRIX EXPONENTIALS
C ***** DOUBLE PRECISION IS NEEDED FOR REQUIRED ACCURACY
PARAMETER NM=18
COMPLEX A(NM,NM),P(NM,NM)
DOUBLE PRECISION AA(NM,NM),PP(NM,NM),B(NM,NM),C(NM,NM),HAVE
DOUBLE PRECISION CC(NM),COL,XNORM
HAVE=H
DO 2 I=1,N
  DO 1 J=1,N
    B(I,J)=0.DO
    C(I,J)=0.DO
    PP(I,J)=0.DO
    AA(I,J)=DBLE(REAL(A(I,J)))

1    CONTINUE
2    CONTINUE
  DO 17 I=1,N
    COL=0.DO
    DO 16 J=1,N
      COL=DMAX1(COL,DABS(DBLE(REAL(A(I,J)))))
16   CONTINUE
    CC(I)=COL
17   CONTINUE
    XNORM=CC(1)
    DO 18 I=1,N
      IF(XNORM .GT. CC(I)) GO TO 18
    XNORM=CC(I)
18   CONTINUE
C    *****
C    * COLUMN NORM IS USED TO SEE WHETHER EXP(A) NEEDS REDUCTION *
C    *****
M=0
30  IF(XNORM*H - 0.98) 3,20,20
C    *****
C    * EXP(A)=(I-.5*A)**(-1)*(I+.5*A) *
C    *****
3  DO 6 I=1,N
  DO 5 J=1,N
    DO 4 K=1,N
      PP(I,J)=PP(I,J)+AA(I,K)*AA(K,J)
4    CONTINUE
    C(I,J)=-AA(I,J)*H/2.0
5    CONTINUE
    C(I,I)=C(I,I)+1.DO
6  CONTINUE
CALL INVERT(C,N,B)

```

## INITIAL DISTRIBUTION LIST

Addressee	No. of Copies
NAVSEASYSKOM, SEA-62R (C. Walker, D. Porter), -63Y42 (P. Baumann)(6)	8
NOSC, Code 6322 (Dr. M. Shensa)	1
DTNSRDC, Code 1548 (Dr. P. P. Rispin)	1
NAVPGSCOL	1
DTIC	12
Dunn & Associates, (Dr. S. Dunn) P.O. Box 2408, Boca Raton, FL 33432	1
E.G.&G/WASC, (F. Galleta, P. BullWinkle), 1650 SW 39th St., Ft. Lauderdale, FL 33315	2
Hydrotronics, Inc., (Dr. S. Berlin), 631 S. Brookhurst St., Anaheim, CA 92804	1
MAR, Inc., (D. Wagner), 22 Pennsylvania Ave., Niantic, CT 06357	1
TASC, (Dr. J. D'Appolito), 6 Jacob Way, Reading, MA 01867	1

