NASA Technical Paper 2418

February 1985

Combined Aerodynamic and Structural Dynamic Problem Emulating Routines (CASPER)

Theory and Implementation

William H. Jones

NASA

NASA
Technical
Paper
2418

1985

Combined Aerodynamic and Structural Dynamic Problem Emulating Routines (CASPER)

Theory and Implementation

William H. Jones

Lewis Research Center
Cleveland, Ohio

National Aeronautics
and Space Administration

Scientific and Technical
Information Branch

Summary

The Lewis Research Center of the National Aeronautics and Space Administration has, since 1950, dealt with a variety of turbomachinery airflow problems that have challenged current analytical abilities. Typically these flows are subsonic or transonic, time varying, and three dimensional. Furthermore the physical structures affecting these flows, usually turbomachinery blading, are also time varying above and beyond simple rotations about the axis of the machine. Typical flow problems include inlet distortion, aeroelastic instability (e.g., flutter), and, most recently, nonrecoverable stall.

In the mid 1970's the author saw the need for a computational tool to transcend these flow field difficulties. Although a number of computer programs exist to deal with subsets of these problems, a quick review at that time showed that almost all such programs withered away to oblivion in the combined presence of these difficulties. The author therefore set out to produce a computational tool to deal successfully with all facets of such problems at the same time.

Two key rules were established at the outset. First, data input was not to include empirical information relevant to a particular engine, blade, flow field, or otherwise. This requirement has been the downfall of many other programs. Such programs tend to be "tweaked" for a particular engine or configuration and then fail to predict well for a different situation.

The second key rule was that computational cost was not to be considered as a limiting factor (although this was not to preclude programming efficiency when possible). Historical perspective shows that programs that appear unthinkably large in one decade usually turn out to be trivially small for the computing equipment of the next decade. Since the computational tool to be developed was expected to have a long lifetime, it was regarded as unwise to accept current computational power as a guideline in sizing the program.

This report presents the theoretical basis of the resulting computational tool and discusses (briefly) the current implementation of the tool. No results are reported since only incomplete results were obtained before the project was discontinued for nontechnical reasons. Because of the scope and volume of the work, other facets of the work (usage instructions, problem statement details, etc.) are held in abeyance for future reporting.

Introduction

CASPER is a collection of computer routines that provide a solution or, more precisely, a simulation of Navier-Stokes flow through and around arbitrarily shaped bodies. Flows calculated are truly three dimensional, time varying, and viscous. No symmetry is required. Flow boundaries may also be arbitrary, nonsymmetric, and time varying.

Traditional solutions to the airflow problem are usually Eulerian in nature: a grid is defined within the flow field and the flow parameters at each grid point are calculated. Usually this approach, which is appealing for field theory conceptualizations, is appalling for boundary satisfaction. Arbitrary, time-varying bodies will often foil the Eulerian scheme completely because of either the inability to fit a grid to the flow field or the inability to transform the problem (or its solution) to or from some solution space.

The fundamental solution concept in CASPER is the Lagrangian aerodynamic element (aeroelement). This simple approach of "observing" a vast number of "air balls" as they proceed through a test area exchanges the nightmare of field theory computation for the nightmare of data-base management and computation. The solution validity hinges on the assumption that a large collection of small-scale, discrete Navier-Stokes solutions involving an aeroelement and several of its nearest neighbor aeroelements (also referred to just as neighbors) will closely approximate a large-scale, continuous solution of a Navier-Stokes flow. If this assumption is valid (which current experience would seem to indicate is the case), CASPER becomes mostly an effort in bookkeeping and computational effectiveness.

CASPER follows each individual Lagrangian aeroelement as it moves through the defined boundaries, maintaining a complete supply of information about the aeroelement. This information includes the position, velocity, acceleration, temperature, pressure (including a history of recent aeroelement pressures), volume (including a history of recent aeroelement volumes), mass, and list of nearest neighbors. Motion through the defined boundaries is in short, iterative steps. Each such step involves a complete data-base revision sequence, resulting finally in revised values for the acceleration of each aeroelement. (CASPER performs these calculations in a consistent set of mass-length-time units; that is, regardless of what unit system is used to supply information to CASPER, it is assumed that a unit force is equal to a unit mass times a unit length divided by a unit time squared.)

Another key feature of CASPER is that the data for each aeroelement are accessible individually from those of all other aeroelements. By arranging the calculation sequence into phases that segregate inputs from outputs, CASPER can be worked on in parallel during any given phase of calculation. The Parallel, Asynchronous Executive (PAX) provides the management services necessary to calculate results in parallel (ref. 1).

CASPER is currently resident on a UNIVAC 1100/42 computer system operating under the EXEC VIII operating system. The problem used for CASPER testing required some 100 million bytes of mass storage for holding aeroelement data and problem description information. This test problem was considered small in comparison with real problems of current interest in the aeropropulsion industry.

This report details the theoretical basis of CASPER, suggests the basic Navier-Stokes simulation iteration, and touches upon some implementation details. Although the basic algorithms appeared to be operational, work was terminated before complete results could be obtained that would support any success or failure judgments. Therefore no computational results are presented in this report.

Symbols

A presumed surface area of an aeroelement

a acceleration

b second-order tensor transformation coefficients

C coefficient relating fluid normal stress to fluid normal strain rate in a Newtonian fluid

C_{HT} coefficient of interaeroelement conductive heat transfer

c_v coefficient of specific heat at constant volume

e fluid strain rate

f generic functions

g acceleration of gravity

h gradient direction of gravitational potential

J Jacobian operator (i.e., the matrix of second partial derivatives)

P_{HT} power of interaeroelement conductive heat transfer

p static pressure

R ideal-gas constant

S fluid stress

T aeroelement static temperature

t time

U aeroelement internal energy

u velocity

W_c work of aeroelement compression

W_d work of aeroelement viscous distortion

x coordinate direction, or position in a coordinate frame

δ_{ij} Kronecker delta function ($\delta=1$ for $i=j$; $\delta=0$ for $i \neq j$)

θ sum of fluid normal strain rates

λ coefficient relating fluid normal stress to fluid shear strain rate in a Newtonian fluid

μ first coefficient of viscosity

μ' second coefficient of viscosity

ρ static density

Subscripts represent coordinate directions unless otherwise noted.

Theoretical Basis

Overview

CASPER exists not as a single program but as a library of aeroelement data-base modification modules. Each module provides one modification of the aeroelement data base. It is only when the user causes these modules to be executed in succession that a Navier-Stokes flow simulation occurs.

The following sections highlight the theoretical basis behind some of the modules. The discussions are oriented toward individual computational goals and occasionally overlap several software modules. The implementation of the theory in actual modules is discussed in a later section.

Aeroelement Accelerations

Aeroelement accelerations are based on the Navier-Stokes equation derived below. The derivation is drawn from reference 2.

Consider the aeroelement mass shown in figure 1. The net elemental fluid stress in the x_1 direction is

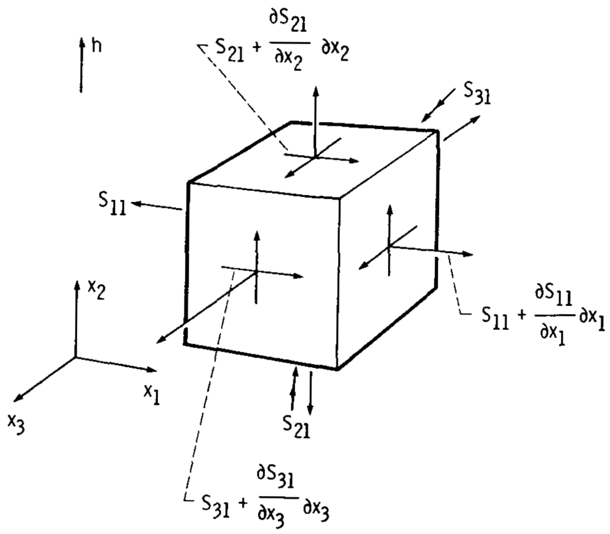

Figure 1.—Elemental fluid mass.

$$\left(\frac{\partial S_{11}}{\partial x_1} dx_1\right) dx_2 dx_3 + \left(\frac{\partial S_{21}}{\partial x_2} dx_2\right) dx_3 dx_1 + \left(\frac{\partial S_{31}}{\partial x_3} dx_3\right) dx_1 dx_2 = \frac{\partial S_{i1}}{\partial x_i} dx_1 dx_2 dx_3 \tag{1}$$

The force of gravity always acts in the negative h direction. The force of gravity in the x_1 direction is then

$$-g\rho \frac{\partial h}{\partial x_1} dx_1 dx_2 dx_3 \tag{2}$$

where g is the acceleration of gravity and ρ is the density of the aeroelement. By equating force to mass times acceleration (and thus assuming constant aeroelement mass), the following result is obtained:

$$\rho dx_1 dx_2 dx_3 a_1 = -g\rho \frac{\partial h}{\partial x_1} dx_1 dx_2 dx_3 + \frac{\partial S_{i1}}{\partial x_i} dx_1 dx_2 dx_3 \tag{3}$$

or

$$a_1 = -g \frac{\partial h}{\partial x_1} + \frac{1}{\rho} \frac{\partial S_{i1}}{\partial x_i}$$

where a_1 is the acceleration of the aeroelement in the first coordinate direction. This equation can be generalized for each axis as

$$a_j = -g \frac{\partial h}{\partial x_j} + \frac{1}{\rho} \frac{\partial S_{ij}}{\partial x_i} \tag{4}$$

The fluid is assumed to be Newtonian in nature, which confers upon it the following properties:

(1) The fluid is isotropic.

(2) The pressure p contributes directly to normal stresses.

(3) Shear stresses are linearly related to shear strain rates.

If the coordinate system is taken to be that of the principal stresses and strains (which coincide for isotropic fluids), it is then assumed that

$$\left.\begin{aligned} S_{11} &= -p + Ce_{11} + \lambda e_{22} + \lambda e_{33} \\ S_{22} &= -p + \lambda e_{11} + Ce_{22} + \lambda e_{33} \\ S_{33} &= -p + \lambda e_{11} + \lambda e_{22} + Ce_{33} \end{aligned}\right\} \tag{5}$$

where S_{ii} is the principal fluid stress in the x_i coordinate direction and e_{ij}, the fluid strain rate, is

$$e_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$$

These equations can be rewritten as

$$\left.\begin{aligned} S_{11} &= -p + \lambda\theta + 2\mu e_{11} \\ S_{22} &= -p + \lambda\theta + 2\mu e_{22} \\ S_{33} &= -p + \lambda\theta + 2\mu e_{33} \end{aligned}\right\} \tag{6}$$

where

$$2\mu = C - \lambda$$

$$\theta = e_{11} + e_{22} + e_{33}$$

These equations can in turn be generalized as

$$S_{mn} = -p\delta_{mn} + \lambda\theta\delta_{mn} + 2\mu e_{mn} \tag{7}$$

Transforming to a primed, nonprincipal coordinate system gives

$$\begin{aligned} S'_{ij} &= b_{mi} b_{nj} S_{mn} \\ &= b_{mi} b_{nj}(-p\delta_{mn} + \lambda\theta\delta_{mn} + 2\mu e_{mn}) \\ &= (-p\delta_{mn} + \lambda\theta\delta_{mn}) b_{mi} b_{nj} + 2\mu b_{mi} b_{nj} e_{mn} \\ &= -p\delta'_{ij} + \lambda\theta\delta'_{ij} + 2\mu e'_{ij} \end{aligned} \tag{8}$$

By dropping the primes the general relation

$$S_{ij} = -p\delta_{ij} + \lambda\theta\delta_{ij} + 2\mu e_{ij} \tag{9}$$

is obtained. The second coefficient of viscosity, $\mu' = \lambda + 2/3\,\mu$, is introduced and the relation becomes

$$S_{ij} = -p\delta_{ij} + \left(\mu' - \frac{2}{3}\mu\right)\theta\delta_{ij} + 2\mu e_{ij} \tag{10}$$

Substituting this expression into equation (4) produces

$$\rho a_j = -\rho g\frac{\partial h}{\partial x_j} - \frac{\partial p}{\partial x_j} + \frac{\partial}{\partial x_j}\left[\left(\mu' - \frac{2}{3}\mu\right)\theta\right] + \frac{\partial}{\partial x_i}\left[\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right] \tag{11}$$

This equation is the Navier-Stokes relation and is used to obtain accelerations a_j based on the velocities and pressures of the aeroelements.

Calculation of Derivatives

CASPER calculates the derivatives of scalar quantities by inverting a Taylor series approximation to the function. The approximation is centered about the aeroelement location at which the derivative values are desired. The multidimensional Taylor series for a scalar function is

$$f(\vec{x} + \Delta\vec{x}) = f(\vec{x}) + \vec{\nabla} f(\vec{x})\Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T J[f(\vec{x})]\Delta\vec{x} + \ldots \tag{12}$$

where

$$J\{f(\vec{x})\} = \begin{bmatrix} \frac{\partial^2 f(\vec{x})}{\partial x_1^2} & \frac{\partial^2 f(\vec{x})}{\partial x_1 \partial x_2} & \frac{\partial^2 f(\vec{x})}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f(\vec{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 f(\vec{x})}{\partial x_2^2} & \frac{\partial^2 f(\vec{x})}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f(\vec{x})}{\partial x_3 \partial x_1} & \frac{\partial^2 f(\vec{x})}{\partial x_3 \partial x_2} & \frac{\partial^2 f(\vec{x})}{\partial x_3^2} \end{bmatrix}$$

This expression can be restated to consider the change in function value between two points in space as

$$f(\vec{x} + \Delta\vec{x}) - f(\vec{x}) = \vec{\nabla} f(\vec{x})\Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T J[f(\vec{x})]\Delta\vec{x} + \ldots \tag{13a}$$

or

$$\Delta f = \vec{\nabla} f(\vec{x})\Delta\vec{x} + \frac{1}{2}\Delta\vec{x}^T J\{f(\vec{x})\}\Delta\vec{x} + \ldots \tag{13b}$$

In CASPER, it is arbitrarily assumed that terms beyond the second order are negligible. Thus the truncated equation can be written in its expanded form as

$$\Delta f = \left\{\frac{\partial f}{\partial x_1}\ \frac{\partial f}{\partial x_2}\ \frac{\partial f}{\partial x_3}\right\} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{Bmatrix} + \frac{1}{2}\{\Delta x_1\ \Delta x_2\ \Delta x_3\} \times \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{bmatrix} \begin{Bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \end{Bmatrix} \tag{14}$$

In the following equation (as well as in subsequent equations of this section only), the subscript of the function value and the second subscript of position coordinates refer to values obtained with respect to the identified point in space. The first subscript of position coordinates identifies the coordinate direction as in previous equations. Thus x_{2i} represents the change in value in the second coordinate direction from the aeroelement position to the point "i" in space.

With this new subscript form in mind, equation (14) can be further expanded by using values from aeroelement neighbor "i" and the aeroelement itself as

$$\Delta f_i = \frac{\partial f}{\partial x_1}\Delta f_{1i} + \frac{\partial f}{\partial x_2}\Delta x_{2i} + \frac{\partial f}{\partial x_3}\Delta x_{3i} + \frac{1}{2}\{\Delta x_{1i}\,\Delta x_{2i}\,\Delta x_{3i}\} \times \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2}\Delta x_{1i} + \frac{\partial^2 f}{\partial x_1 \partial x_2}\Delta x_{2i} + \frac{\partial^2 f}{\partial x_1 \partial x_3}\Delta x_{3i} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1}\Delta x_{1i} + \frac{\partial^2 f}{\partial x_2^2}\Delta x_{2i} + \frac{\partial^2 f}{\partial x_2 \partial x_3}\Delta x_{3i} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1}\Delta x_{1i} + \frac{\partial^2 f}{\partial x_3 \partial x_2}\Delta x_{2i} + \frac{\partial^2 f}{\partial x_3^2}\Delta x_{3i} \end{bmatrix} \tag{15a}$$

ORIGINAL PAGE IS
OF POOR QUALITY

or

$$\Delta f_i = \frac{\partial f}{\partial x_1}\Delta x_{1i} + \frac{\partial f}{\partial x_2}\Delta x_{2i} + \frac{\partial f}{\partial x_3}\Delta x_{3i} + \frac{1}{2}\left\{ \frac{\partial^2 f}{\partial x_1^2}\Delta x_{1i}^2 + \frac{\partial^2 f}{\partial x_1 \partial x_2}\Delta x_{1i}\Delta x_{2i} + \frac{\partial^2 f}{\partial x_1 \partial x_3}\Delta x_{1i}\Delta x_{3i} \right.$$

$$\left. + \frac{\partial^2 f}{\partial x_2 \partial x_1}\Delta x_{2i}\Delta x_{1i} + \frac{\partial^2 f}{\partial x_2^2}\Delta x_{2i}^2 + \frac{\partial^2 f}{\partial x_2 \partial x_3}\Delta x_{2i}\Delta x_{3i} + \frac{\partial^2 f}{\partial x_3 \partial x_1}\Delta x_{3i}\Delta x_{1i} + \frac{\partial^2 f}{\partial x_3 \partial x_2}\Delta x_{3i}\Delta x_{2i} + \frac{\partial^2 f}{\partial x_3^2}\Delta x_{3i}^2 \right\} \quad (15b)$$

If nine neighbor positions were available, the following matrix equation could be written:

$$\begin{Bmatrix} \Delta f_1 \\ \Delta f_2 \\ \Delta f_3 \\ \Delta f_4 \\ \Delta f_5 \\ \Delta f_6 \\ \Delta f_7 \\ \Delta f_8 \\ \Delta f_9 \end{Bmatrix} = \begin{bmatrix} \Delta x_{11} & \Delta x_{21} & \Delta x_{31} & \tfrac{1}{2}\Delta x_{11}^2 & \tfrac{1}{2}\Delta x_{21}^2 & \tfrac{1}{2}\Delta x_{31}^2 & \Delta x_{11}\Delta x_{21} & \Delta x_{21}\Delta x_{31} & \Delta x_{31}\Delta x_{11} \\ \Delta x_{12} & \Delta x_{22} & \Delta x_{32} & \tfrac{1}{2}\Delta x_{12}^2 & \tfrac{1}{2}\Delta x_{22}^2 & \tfrac{1}{2}\Delta x_{32}^2 & \Delta x_{12}\Delta x_{22} & \Delta x_{22}\Delta x_{32} & \Delta x_{32}\Delta x_{12} \\ \Delta x_{13} & \Delta x_{23} & \Delta x_{33} & \tfrac{1}{2}\Delta x_{13}^2 & \tfrac{1}{2}\Delta x_{23}^2 & \tfrac{1}{2}\Delta x_{33}^2 & \Delta x_{13}\Delta x_{23} & \Delta x_{23}\Delta x_{33} & \Delta x_{33}\Delta x_{13} \\ \Delta x_{14} & \Delta x_{24} & \Delta x_{34} & \tfrac{1}{2}\Delta x_{14}^2 & \tfrac{1}{2}\Delta x_{24}^2 & \tfrac{1}{2}\Delta x_{34}^2 & \Delta x_{14}\Delta x_{24} & \Delta x_{24}\Delta x_{34} & \Delta x_{34}\Delta x_{14} \\ \Delta x_{15} & \Delta x_{25} & \Delta x_{35} & \tfrac{1}{2}\Delta x_{15}^2 & \tfrac{1}{2}\Delta x_{25}^2 & \tfrac{1}{2}\Delta x_{35}^2 & \Delta x_{15}\Delta x_{25} & \Delta x_{25}\Delta x_{35} & \Delta x_{35}\Delta x_{15} \\ \Delta x_{16} & \Delta x_{26} & \Delta x_{36} & \tfrac{1}{2}\Delta x_{16}^2 & \tfrac{1}{2}\Delta x_{26}^2 & \tfrac{1}{2}\Delta x_{36}^2 & \Delta x_{16}\Delta x_{26} & \Delta x_{26}\Delta x_{36} & \Delta x_{36}\Delta x_{16} \\ \Delta x_{17} & \Delta x_{27} & \Delta x_{37} & \tfrac{1}{2}\Delta x_{17}^2 & \tfrac{1}{2}\Delta x_{27}^2 & \tfrac{1}{2}\Delta x_{37}^2 & \Delta x_{17}\Delta x_{27} & \Delta x_{27}\Delta x_{37} & \Delta x_{37}\Delta x_{17} \\ \Delta x_{18} & \Delta x_{28} & \Delta x_{38} & \tfrac{1}{2}\Delta x_{18}^2 & \tfrac{1}{2}\Delta x_{28}^2 & \tfrac{1}{2}\Delta x_{38}^2 & \Delta x_{18}\Delta x_{28} & \Delta x_{28}\Delta x_{38} & \Delta x_{38}\Delta x_{18} \\ \Delta x_{19} & \Delta x_{29} & \Delta x_{39} & \tfrac{1}{2}\Delta x_{19}^2 & \tfrac{1}{2}\Delta x_{29}^2 & \tfrac{1}{2}\Delta x_{39}^2 & \Delta x_{19}\Delta x_{29} & \Delta x_{29}\Delta x_{39} & \Delta x_{39}\Delta x_{19} \end{bmatrix} \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \frac{\partial f}{\partial x_3} \\ \frac{\partial^2 f}{\partial x_1^2} \\ \frac{\partial^2 f}{\partial x_2^2} \\ \frac{\partial^2 f}{\partial x_3^2} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} \end{Bmatrix} \quad (16)$$

Given the aeroelement positions and the function values involved, this matrix equation could be inverted to yield all of the first and second derivatives of the scalar function at the location of the target aeroelement. It is assumed that only the derivatives

$$\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \frac{\partial^2 f}{\partial x_1^2}, \text{ and } \frac{\partial^2 f}{\partial x_2 \partial x_3}$$

are nonzero. Thus the matrix equation is reduced to

$$\begin{Bmatrix} \Delta f_1 \\ \Delta f_2 \\ \Delta f_3 \\ \Delta f_4 \\ \Delta f_5 \end{Bmatrix} = \begin{bmatrix} \Delta x_{11} & \Delta x_{21} & \Delta x_{31} & \tfrac{1}{2}\Delta x_{11}^2 & \Delta x_{21}\Delta x_{31} \\ \Delta x_{12} & \Delta x_{22} & \Delta x_{32} & \tfrac{1}{2}\Delta x_{12}^2 & \Delta x_{22}\Delta x_{32} \\ \Delta x_{13} & \Delta x_{23} & \Delta x_{33} & \tfrac{1}{2}\Delta x_{13}^2 & \Delta x_{23}\Delta x_{33} \\ \Delta x_{14} & \Delta x_{24} & \Delta x_{34} & \tfrac{1}{2}\Delta x_{14}^2 & \Delta x_{24}\Delta x_{34} \\ \Delta x_{15} & \Delta x_{25} & \Delta x_{35} & \tfrac{1}{2}\Delta x_{15}^2 & \Delta x_{25}\Delta x_{35} \end{bmatrix} \begin{Bmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \frac{\partial f}{\partial x_3} \\ \frac{\partial^2 f}{\partial x_1^2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_3} \end{Bmatrix} \quad (17)$$

This matrix, when inverted, yields two second-order derivatives that are not generally needed. Therefore CASPER algorithms invert only the first three rows of the matrix; however, this does not prohibit future implementations from performing the full inversion of the 9-by-9 matrix, should conditions warrant this.

CASPER provides an inversion (as specified above) of equation (17) for every aeroelement. Application of the inverted-matrix result quickly provides the gradient of any scalar quantity whose values are known on an aeroelement-by-aeroelement basis.

Some CASPER algorithms must calculate second derivatives of quantities. Since the CASPER derivative calculation procedure does not compute the information necessary to obtain these second derivatives directly, an alternative scheme is followed. In these cases the first derivatives of the quantity are calculated for all aeroelements in the normal manner; subsequently, the derivatives of the calculated first derivatives (in effect, the second derivatives of the quantity) are calculated by the same procedure.

Fluid Element Movement and Physical Boundaries

The most difficult task within CASPER is the movement of aeroelements through realistic physical boundaries. Two simplifying assumptions have been made. First, physical boundaries must be describable as mathematical entities rather than as collections of points in space. Second, the parameters governing these mathematical entities must be explicitly known during the period of time that motion is to take place.

The first assumption leads naturally to the formulation that any boundary is described as the locus of points in space-time for which

$$f(\vec{x},t) = 0 \tag{18}$$

If

$$f(\vec{x},t) > 0$$

the point in space-time is in the portion of space that is not occupied by the physical object behind the boundary. If

$$f(\vec{x},t) \leq 0$$

the point in space-time is in the portion of space occupied by that physical object.

Clearly, finding one such function that will suffice for any real problem (e.g., an axial-flow turbomachine) is a practical impossibility. Thus CASPER allows the range of influence of a given boundary to be limited by other described surfaces (known as surface truncations) and for many such truncated surfaces to be, effectively, concatenated into any desired combination. Mathematically, truncation can be stated as follows: If for all indicated truncation surfaces i,

$$f_i(\vec{x},t) \geq 0$$

then

$$f(\vec{x},t) = f(\vec{x},t) \tag{19}$$

otherwise

$$f(\vec{x},t) = 1$$

As currently implemented, truncation services can be nested to any practical depth. This allows truncation surfaces, themselves, to be truncated.

The concatenation of surfaces is a byproduct of surface truncation rather than a true mathematical conjunction mechanism. Since a truncated surface will always return a safe (nonviolating) surface value outside its range of influence, the problem description is free to supply another surface in that area of space-time. Thus, by limiting the range of influence of each surface, many surfaces can be provided to describe complex shapes.

Another key feature of the surface truncation mechanism is that the distinction between safe surface values obtained by truncation and those obtained by nontruncated surface evaluation is retained and available to the user. This feature is useful in defining other spatial structures.

Actual surfaces defined in CASPER also provide the gradient of the surface function and the velocity and acceleration of any point in space that is moving in free-body synchronism with the surface. These values are available everywhere in the space-time continuum with the possible exception of points of singularity for the surface function. These additional surface characteristics are always evaluated without regard to any described surface truncations. It is up to the invoking algorithm to establish the validity (or influence) of a given surface before using these additional characteristics.

The second major assumption, explicit knowledge of parameters, is necessary for the segregation of inputs and outputs. One of the goals of CASPER was a high degree of interaction between aerodynamic forces and structural motion; however, to include such interactions at the incremental aeroelement motion level would have given rise to a logical loop without resolution.

Thus the shape of all objects in a CASPER problem must be explicitly known at the outset of each incremental motion calculation. Time variations are allowed provided that they are explicitly known, as other parameters are, at the outset of computation for the incremental period. (Indeed, within the CASPER surface evaluation package, time variation parameters are indistinguishable from all other parameters; only within the incremental motion package is a space-versus-time distinction made.)

This restriction does not inhibit relating aerodynamic force and structural motion. Instead, such considerations are moved to the point in time (truly a point, i.e., a span of zero duration) between successive incremental motion periods. At that point, surface parameters can be set to new values according to any desired scheme. Typically the newly revised aerodynamic load history would be used to project a new structural motion for the next incremental motion period. Presumably this new motion would affect the aerodynamics of the upcoming incremental motion period.

The process of aeroelement motion begins simply with the integration of the aeroelement accelerations. These accelerations are assumed to be constant through the time period. Thus

$$u_i(t + \Delta t) = u_i(t) + \int_t^{t+\Delta t} a_i dt$$

$$= u_i(t) + a_i \Delta t \tag{20}$$

and

$$x_i(t + \Delta t) = x_i(t) + \int_t^{t+\Delta t} u_i(t) dt$$

$$= x_i(t) + u_i(t)\Delta t + \frac{1}{2} a_i \Delta t^2 \tag{21}$$

At this point the constraints of boundaries must be accommodated. This is first enforced by requiring

$$f[\vec{x}(t + \Delta t), t + \Delta t] \geq 0 \tag{22}$$

for each appropriate boundary function f. Since boundaries that do not apply will presumably fall to surface truncations and report values for f of $+1$, this condition is sufficient. However, it soon becomes clear that for realistic problems the calculation burden of comparing millions of aeroelement positions to thousands of surfaces would best be avoided.

CASPER reduces this effort by requiring the definition of flow zones in the problem statement and the listing of any and all boundaries resident (in whole or in part) in each flow zone. Flow zones are defined by surfaces (usually truncated) in the same manner as physical boundaries. The condition for residence in a flow zone is

$$f(\vec{x}, t) \geq 0 \tag{23}$$

where f is appropriate for the particular flow zone and is not determined by truncation of the surface (i.e., if the surface evaluation encounters truncation and returns the automatic value of $+1$, the point of evaluation is not considered to be in the described flow zone). CASPER determines and retains for each aeroelement the current flow zone of residence. By reviewing the resident boundary list for the appropriate flow zone, the number of comparisons necessary to satisfy the conditions of equation (22) can be reduced.

Additional information establishing the neighbor flow zones for each flow zone is used to narrow the range of search when an aeroelement flows out of one flow zone and into another. Current implementations of CASPER do not support automatic, time-varying alterations of the information on flow-zone neighbors or flow-zone resident boundaries. However, there is nothing in the CASPER algorithms that prohibits the change of such information on a dynamic basis.

When aeroelement motion results in boundary violation (as in fig. 2), the point of violation is found by a simple bisection iteration, where the bisected variable is time and position is treated as parametric in time. When this point of violation is found, corrective measures are applied to the aeroelement datum to alter its path through space from the point of violation. First, the velocity of the aeroelement is reset in a calculation that equates the angle of incidence to the angle of reflection. Thus the aeroelement bounces elastically at the point of violation. This calculation is carried out with respect to the motion of the surface at the point of violation. In this way, if the aeroelement is motionless and the boundary is moving, velocity is imparted to the aeroelement to move it out of the way of the boundary. Second, if boundary violation occurs without apparent motion of the aeroelement normal to the boundary, the acceleration of the aeroelement normal to the boundary is set to zero. This corrects problems that can occur when the aero-

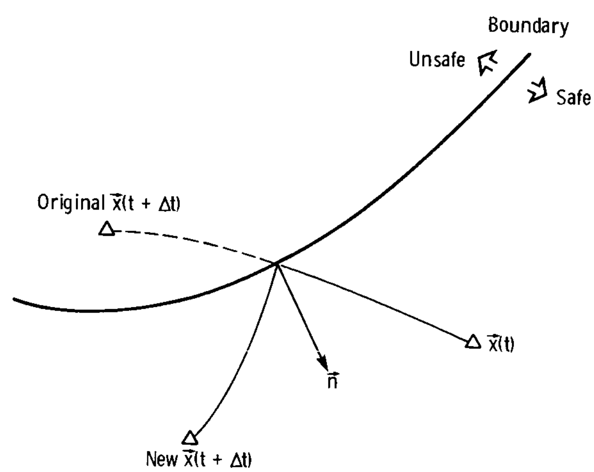

Figure 2.—Boundary violation.

element is differentially close to the boundary and has an acceleration that, at least in part, is into the boundary. (In this instance it can become computationally impossible to locate any point forward in time, even after velocity alteration, that does not produce a boundary violation.)

This formulation of boundary violation checking works well for surfaces with large object depths; however, for thin objects (such as corners and edges) a phenomenon referred to as "passthrough" becomes a problem. Regardless of the duration of the incremental motion period, sharp corners will fail to deflect some aeroelement paths that should be deflected (fig. 3). To reduce this effective dulling of sharp edges, the initial urge is to shorten the motion time, thus reducing the distance traveled between boundary checks. Although this is effective, it can also be quite costly since it brings with it increased resolution in the entire Navier-Stokes simulation process. CASPER offers an alternative by allowing the calculation of a user-specified number of intermediate aeroelement positions, which are all based on the original position, velocity, and acceleration information.

The intermediate-positions option allows the user to set the flow field resolution and the edge sharpness independently, provided that the edge sharpness is the more restrictive constraint. It is, of course, desirable not to oversharpen edges, since this still requires additional surface comparisons. It is expected that future implementations of CASPER will adjust edge sharpness dynamically. Probably a lower sharpness limit will be set

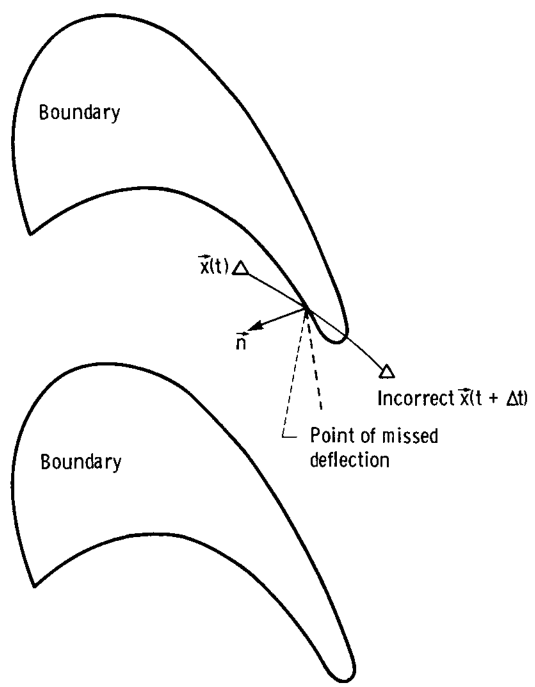

Figure 3.—Boundary passthrough phenomenon.

and will be increased from that point in proportion to the proximity of the boundary to the aeroelement.

Finally CASPER provides the ability, on an aeroelement-by-aeroelement basis, to disable the standard aeroelement motion algorithm from repositioning an aeroelement. In this event it becomes the user's responsibility to provide and invoke routines that move (or do not move, as appropriate) the affected aeroelements. This feature is intended, principally, for the simulation of boundary layer flows by attaching some aeroelements to the boundaries while other aeroelements flow past the attached aeroelements.

Thermodynamic Effects

All thermodynamic calculations in CASPER result in power quantities that, when multiplied by the time increment duration, give the amount of work transferred into a particular aeroelement. CASPER computes for each aeroelement the power of compression, the power of viscous aeroelement distortion, and the rate of heat flow (adjusted to units of power). In all cases flow of energy into the aeroelement is taken to be positive. At an appropriate time in the Navier-Stokes simulation loop, these powers are summed, multiplied by the time increment duration, and used to alter a total, internal energy value for each aeroelement. This internal energy is then related to aeroelement temperature through a simple coefficient-of-specific-heat relation.

From fundamental thermodynamics the compressive work done on an aeroelement is

$$W_c = \int p\,dv \tag{24}$$

The CASPER data base records the five most recent pressure and volume values for each aeroelement. The three most recent of each of these values (the most recent of which is the current value for the aeroelement) are used to establish parabolic approximations for pressure and volume as functions of time. These approximations are substituted into the expression for the work of compression, and the result is parametrically integrated from the time of the previous pressure and volume value to the time of the current values. Although currently the pressure and volume histories must be equally spaced in time, this constraint can be removed, if necessary, by simple programming (i.e., no other CASPER module depends on the spacing of pressure and volume histories). Finally the work of compression is divided by the time-increment duration to establish an equivalent power.

The power of viscous distortion arises from the resistance of the fluid to deformation. As delineated in reference 3, given the Newtonian fluid assumption of the Navier-Stokes equation, the distortion work can be calculated as

$$W_d = v\left[\frac{\partial}{\partial x_1}(u_1S_{11}+u_2S_{12}+u_3S_{13}) + \frac{\partial}{\partial x_2}(u_1S_{21}+u_2S_{22}+u_3S_{23}) + \frac{\partial}{\partial x_3}(u_1S_{31}+u_2S_{32}+u_3S_{33})\right] = v\left[\frac{\partial}{\partial x_i}(u_jS_{ij})\right] \quad (25)$$

From equation (10) this becomes

$$W_d = v\frac{\partial}{\partial x_i}\left\{u_j\left[-p\delta_{ij}+\left(\mu'-\frac{2}{3}\mu\right)\theta\delta_{ij}+2\mu e_{ij}\right]\right\} \quad (26)$$

Again, the computed work is divided by the duration of the time increment to obtain a power-of-viscous-distortion value.

Finally simple conductive heat flow between neighbor aeroelements is considered. Heat flow per unit time is simply taken to be proportional to the difference in the temperatures of the two aeroelements, the average of the heat transfer coefficients, the average presumed surface areas of the aeroelements, and the inverse of the distance between them. Thus

$$P_{\mathrm{HT}} = \frac{(T_j - T_i)(C_{\mathrm{HT}_j}+C_{\mathrm{HT}_i})(A_j+A_i)}{4D_{ij}} \quad (27)$$

where the subscripts i and j denote distinct aeroelements and D_{ij} is the distance between those aeroelements. This computation yields a power term directly. It is only important to assure that the units of the coefficient of heat transfer are such as to yield a number consistent with the previously determined powers of compression and viscous distortion.

When all of the sources of energy are accounted for, the aeroelement temperature is revised on the basis of the assumption

$$\Delta U = c_v \Delta T \quad (28)$$

where c_v is the coefficient of specific heat at constant volume.

Convective heat transfer is not considered explicitly in the thermodynamic model since aeroelements are not constrained against intermixing. Aeroelements are not algorithmically required to maintain the same neighbors throughout the iterative loop process. Aeroelements are free to follow independent paths through space, intermix with new neighbor aeroelements, and transfer heat with these new neighbors. This provides a crude convective heat transfer mechanism.

Pressure and Volume Calculations

CASPER provides the ability to calculate the current pressure and volume of aeroelements from current temperature, mass (which never changes throughout the computational lifetime of an aeroelement), and proximity of neighbors. Fundamental to this calculation is the estimation of volume. CASPER begins the volume estimation process by averaging the distances from the subject aeroelement to each of the nearest neighbors of the aeroelement. This value is then used as a radius for a simple spherical volume determination.

It was forseen that a correction would be needed to accommodate aeroelements with a vacuum on one side and fluid on the other side. In this case an aeroelement with a large volume on one side would be miscalculated as having a normal volume on the basis of the proximity of neighbors on the other side. Therefore a correction based on the solid angle subtended by the nearest neighbors of the aeroelement was implemented. This angle is compared with the average solid angle for all aeroelements (usually obtained during the previous simulation cycle). As the solid angle of a particular aeroelement deviates below average, the initial volume estimate for that aeroelement is increased. Solid angle deviations that are above the average cause the aeroelement volume estimate to be decreased. However, the algorithm is such that the response to above-average solid angles is much less than to those below the average. Above-average solid angles would seem only to indicate an unusual dispersion of nearest neighbors about an aeroelement that is, in fact, fully immersed in the fluid continuum.

It is clear that these initial estimates will not be correct in an absolute sense but will be proportionately correct on an aeroelement-to-aeroelement basis. Thus a final correction is begun by summing all of the aeroelement volume estimates and comparing the result to the known total volume of the flow zones of the problem. A simple multiplicative correction factor is then produced and applied to the volume estimate of each aeroelement to convert proportionately correct values to absolutely correct values.

With the volume result available, the calculation of the density of each aeroelement (given the mass of that aeroelement) is trivial. Calculation of the pressure of each aeroelement (given the previously determined temperature of the aeroelement) is done through the ideal-gas law

$$p = \rho RT \quad (29)$$

CASPER supplies this gas law as a subroutine. Thus alternative gas laws (including table-lookup style laws) can be substituted merely by supplying an appropriate subroutine of the same name. In this way, real gases can be supported by the ingenious CASPER user.

Establishing Neighbor Aeroelements

Many CASPER algorithms hinge on knowing the identification of several (currently five) nearest-neighbor aeroelements for each aeroelement. This is primarily a bookkeeping chore and represents no theoretical insight at all. The criterion for selecting nearest neighbors is a straightforward center-to-center distance comparison of the aeroelements. The principal goal of nearest-neighbor-finding algorithms is to limit the computational cost by limiting the number of nearest-neighbor candidates. CASPER offers three nearest-neighbor-finding algorithms: (1) a full exhaustive search, (2) a search limited by flow zone relations, and (3) a search through the current nearest-neighbor tree.

The exhaustive-search algorithm for finding nearest neighbors of a particular aeroelement (referred to as the target aeroelement) is very simple. The distance from the target aeroelement to every other aeroelement in the population is computed and compared with the current list of nearest neighbors. If the new distance is shorter than the distance to one of the current nearest neighbors, a simple ordered-list replacement (which discards the furthest nearest neighbor and inserts the new nearest neighbor at the appropriate point in the list) is performed. This nearest-neighbor finder may produce dubious results in an algorithmic sense since no consideration is given to the possibility that boundaries may exist between the aeroelement and its selected neighbors. This is called bogus-nearest-neighbor-syndrome (BNNS).

The flow-zone-related search algorithm limits the number of nearest-neighbor candidates by requiring each candidate to be resident either in the flow zone of the target aeroelement or in a flow zone that is a neighbor to the flow zone of the target aeroelement. This approach can eliminate the problem of bogus neighbors being selected from flow areas that are physically close but problematically far. The improvement in search time is clear but may not be sufficient if there are still large numbers of aeroelements in each flow zone. If this nearest-neighbor search algorithm is to be used extensively, it is vital that the problem statement be designed with a feasible average number of aeroelements per flow zone.

The nearest-neighbor-tree search allows the user to specify an arbitrary number of candidates for inclusion in the nearest-neighbor search, thereby providing firm control of the search time. Candidates are drawn from the current nearest-neighbors tree for the target aeroelement (i.e., the current neighbors of the target aeroelement, the current neighbors of the neighbors of the target aeroelement, the current neighbors of the neighbors of the neighbors, and so on). Checks are made to ensure that redundant entries are not made in the neighbor tree and that the target aeroelement does not become its own nearest-neighbor candidate. This algorithm attempts to restrict BNNS by requiring that candidate aeroelements reside either in the flow zone of the target aeroelement or in a flow zone neighbor to the flow zone of the target aeroelement. This approach also assumes that the initial neighbor tree was an essentially valid, well-cross-linked representation of the true neighbor structure (in particular that, if followed far enough, the neighbor tree would eventually involve all aeroelements in the population). If the initial tree were not valid (for instance, if the target aeroelement and its five nearest neighbors all listed each other as neighbors), the neighbor-tree search algorithm would do nothing to improve the validity of the nearest-neighbor list of the target aeroelement.

Aeroelement Relocation

Most problems involve the flow of fluid from a source area through a test area to an exhaust area. To provide a supply of new aeroelements for the source area, CASPER extracts all aeroelements from the designated exhaust area, combines them with all aeroelements currently resident in the source area, and reinitializes the entire set of aeroelements (provided it is nonnull) in the source area. This requires a complete resetting of the data base of each such aeroelement, including new position, velocity, mass, and temperature. The user is required to provide problem-specific routines that generate realistic numbers for this process. This facility can be disabled to allow closed-circuit or closed-box systems to be modeled. In this case, the user-supplied routines, if called, will simply take no action.

Note that during this relocation phase, the user has the opportunity to simulate various interesting conditions and to effect problem control adjustments. For instance, in providing information about aeroelements being relocated to the inlet, the user could model inlet distortions in both pressure and temperature. Also, the user could perform acts such as adjusting the temperatures of aeroelements in combustor flow zones to achieve or maintain a desired combustor exit temperature schedule. The possibilities for user manipulation are endless.

Flow Field Simulation Sequence

Figure 4 presents a possible sequence of CASPER module utilization to simulate a fluid flow. The figure does not represent information specific to current

1.0 Initialize data base and establish starting conditions.

2.0 Establish nearest-neighbor relationships.

3.0 Calculate
  3.1 Aeroelement volumes
  3.2 Aeroelement densities
  3.3 Aeroelement pressures

4.0 Establish derivative calculation matrix for each aeroelement.

5.0 Calculate Navier-Stokes accelerations for each aeroelement.

6.0 Calculate
  6.1 Aeroelement powers of compression
  6.2 Aeroelement powers of viscous distortion
  6.3 Aeroelement conductive heat transfers
  6.4 Aeroelement temperature changes

7.0 Adjust boundary parameters to reflect boundary conditions for next aeroelement motion period.

8.0 Move aeroelements through space by integrating accelerations.

9.0 Remove aeroelements from the exhaust zone, insert them in the inlet zone, and reinitialize inlet zone aeroelement data.

10.0 Go to step 2.0.

Figure 4.—Possible flow simulation sequence.

CASPER implementation but rather covers only the fundamental logic of the approach. The most striking feature of the sequence is that there is no exit. CASPER does not search for any convergence condition. Instead it continues to produce flow results without end. A settling-in period exists when problem simulation first begins; however, it is up to the user to identify the end of this period since CASPER cannot recognize it on its own.

Two additional problems intertwine themselves in CASPER problems: inducing flow, and isolating the test area from unintended inlet and exhaust area distortions. Figure 5 shows a diagram of a simple flow problem that illustrates simple solutions to both these problems.

First, a fluid does not start flowing by itself. At the very least, a pressure gradient or gravitational force must be applied to obtain continuous fluid flow. Simply placing aeroelements in a test area does not accomplish this. One could supply sufficiently massive aeroelements to the inlet zone of the example problem during each aeroelement relocation phase to maintain a given inlet plenum pressure or a given inlet pressure ratio between the inlet and exhaust plenums. This would not provide independent control of the two plenum pressures (which could be a problem requirement). Furthermore the validity of flow-to-vacuum simulations is in some doubt. This could cause aberrations of the results upstream of the vacuum face.

A second alternative could be for a user routine to selectively remove mass from exhaust plenum aeroelements. This would provide independent control of plenum pressures. The difficulty with this approach is that the rationale for selecting the mass reductions is not nearly so clear as that for initializing aeroelements in the inlet plenum. Should all exhaust plenum aeroelements be reduced in mass or only those toward the rear of the plenum? Should a uniform pressure distribution or a pressure gradient result? Should temperature or velocity be adjusted too? Individual users may wish to assert and implement their own answers to these questions (which would be entirely possible within the existing CASPER framework); however, the author sought a third alternative.

The flow-induction alternative the author selected was to model an additional part of the apparatus that would be used in an actual experiment: a centrifugal exhauster. This approach strikes a blow both at inducing flow and at isolating poor flow simulations from the test area. In using this approach, it is necessary to admit that flows in the exhauster may not be realistic. However, the author feels that expecting flows upstream of the exhauster to be realistic is a much more comfortable idea. Furthermore there are no questions to ask about mass removal. Aeroelements are driven around in a circle by the exhauster blading until centrifugal force drags them out into the exhaust zone, where they disappear into the aeroelement relocation algorithms. Also, exhaust plenum pressure is controlled by adjusting the rotational speed of the exhauster model.

Implementation

Overview

The implementation of key CASPER modules is presented here to add corroborative detail in the absence

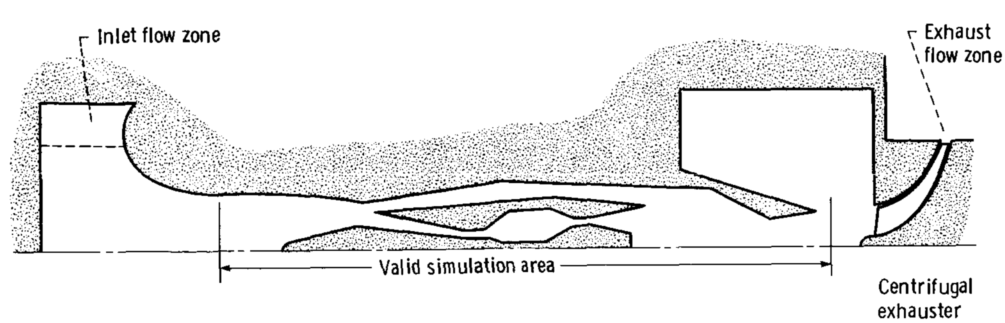

Figure 5.—Cross section of a simple flow problem for CASPER.

of computational results. The listings presented are actual, compiled CASPER code modules. The reader is cautioned that the comments contained in these listings, while generally accurate and up to date, are not guaranteed to represent the exact truth of the code presented. As always, the final arbiter of computational truth is the language statements and the interpretation thereof by the language compiler. The reader is also advised that the symbols previously used in this report are not (always) used by the CASPER codes presented in the listings. In the following discussion of the listings the symbols used by the code will be used as they appear in the listing.

Some elements of the CASPER code presented may appear confusing because of the parallel-processing nature of CASPER. For instance, because of the requirements of parallel processing, results are often stored in scratch slots for later copying back to the formal result slot. It is not within the scope of this report to provide a tutorial on parallel processing and thus these oddities will be ignored.

Finally no specifics about the use of these codes for actual computation are provided. Information such as program building instructions, program use instructions, and data preparation is beyond the scope of this report.

Calculation of Derivatives

The inversion of a Taylor series to provide derivative calculation information is provided by the CASPER module INTF presented in listing 1. This module, in turn, calls the subroutines MATRIX, DET4, and DET3 (listings 2, 3, and 4, respectively). The fundamental flow of control is to invoke the matrix construction and inversion subroutine, MATRIX, for each aeroelement in the aeroelement identification range provided to INTF (i.e., aeroelement identification numbers in the range IL to IH). If MATRIX reports a matrix singularity fault, the event is logged (by calling ERROR2, an environmental service provided by PAX), and an attempt to clear up the problem (by revising the nearest-neighbor list to provide a different matrix for inversion) is made by calling the subroutine CLNINT (which is not presented as a listing in this report).

MATRIX (listing 2) constructs the 5-by-5 matrix of spatial differentials on the basis of the proximity of the listed nearest neighbors for the identified aeroelement (lines 79 to 84). The partial inversion of the matrix proceeds in lines 85 to 105 by the transform-of-cofactor method. This method relies on the fact that the inverse of a matrix is the quotient of the transform of the cofactor matrix divided by the determinant of the matrix. This method was selected since it allows partial inversions to be calculated. Only the determinant and the cofactors of the diagonally reflected term desired need to be calculated to determine the inverse matrix entry for any particular term.

Evaluation proceeds in MATRIX by constructing 4-by-4 matrices (and then, a 3-by-3 matrix in DET4) whose determinants are evaluated to provide the cofactors of the desired terms. The determinant of the original 5-by-5 matrix is evaluated at the first opportunity (lines 99 to 101) and a check is made to see if a singularity problem exists before any further effort is expended. If a singularity problem does exist, the aeroelement is marked as not having a current derivative calculation matrix (lines 108 and 109) and the problem is reported to the calling routine (line 107). If a singularity problem is not detected, the final construction and storage of a 3-row-by-5-column derivative calculation matrix is performed by lines 114 to 119 of the listing. This process includes clearing the derivative matrix not-available flag for the particular aeroelement.

DET4 and DET3 (listings 3 and 4) show the determinant calculation process reduced to the point where the determinant can be calculated directly by formula. Although the entire computational strategy is crude, it has the advantage of working and already being coded and debugged. If CASPER were to be continued into a computational production phase, these algorithms could be improved remarkably.

Calculation of Aeroelement Acceleration

STOK1 and STOK2, the CASPER modules that calculate aeroelement accelerations, are presented in listings 5 and 6. The computation of the accelerations must proceed in two phases since derivatives of terms involving derivatives must be calculated. This requires that the terms involving derivatives must be calculated for all aeroelements before the derivatives of the derivative terms can be determined by the application of the derivative calculation matrix. Note that each routine avoids the acceleration computation entirely if the derivative calculation matrix is not available (see line 104 of listing 5 and line 114 of listing 6). Also, note that various aeroelement information is stored in variables local to the subroutine to avoid the overhead expense of repeated access to values in shared, software-virtualized storage. Finally note that these routines calculate the contribution of all of the terms of equation (11) except the gravitational term.

The first phase of computation, STOK1 (listing 5), proceeds to calculate the derivatives of velocity and the fluid strain rate matrix by applying the derivative calculation matrix (lines 113 to 129). The full strain rate matrix is stored in the aeroelement record for use by other CASPER modules. An additional term (whose derivative must be determined in the second phase of the acceleration calculation) involving viscosity and principal fluid strains is calculated and stored in the aeroelement record (lines 130 to 135).

The second phase of computation, STOK2 (listing 6), completes the calculation of aeroelement acceleration by the Navier-Stokes equation. A preliminary check of aerolement density for approximate equality to zero is made (line 117) to avoid a possible divide fault (line 165). Again, a number of localizations of aeroelement information are performed to avoid shared-data access overheads. The gradients of the pressure (lines 139 to 144) and of the viscosity-and-principal-fluid-strain term (lines 145 to 150) at the aeroelement are calculated by using the standard derivative calculation procedure. An implied tensor sum of the derivatives of the product of viscosity and fluid strains is calculated by lines 151 to 163. Finally, on lines 164 to 166, these three principal terms are combined to form the aeroelement accelerations in each coordinate direction. These accelerations are then stored in the aeroelement record.

This two-phase process completes the computation of aeroelement acceleration through the Navier-Stokes equation. The only term neglected is the gravitational component. It is important to note that, given the data base available for each aeroelement, especially the derivative calculation matrix, which allows gradients to be inferred from the differences in scalar function values, the calculation of accelerations through the Navier-Stokes equation is almost trivial.

Aeroelement Positioning and Physical Boundaries

MOVEL, SURF, and SA100 (listings 7, 8, and 9) present key components of the aeroelement positioning package. Although the actual logic and programming of the aeroelement positioning process (which simply integrates acceleration and velocity to establish position) can appear to be quite involved, it must be borne in mind that underlying it all is a basically simple vision of motion through space subject to real boundaries.

Subroutine MOVEL (listing 7) is the main component of the aeroelement positioning module of CASPER. It implements the aeroelement acceleration integration algorithm and invokes the various boundary evaluation services to enforce the boundary constraints of the problem. An argument supplied to the subroutine (NINC) specifies the number of time subincrement steps to be taken to sharpen boundary edges. Again, aeroelement information is localized to minimize shared-data-access overheads.

The principal aeroelement positioning sequence occurs in lines 304 to 307 of listing 7. The internal subroutine SPC (defined by lines 459 to 482 of the listing) projects the course of the aeroelement on the basis of the current aeroelement position, velocity, and acceleration from formulas based on the classic double integration of acceleration. The internal subroutine YNM (defined by lines 483 to 536) checks to see if the new position violates any boundaries in the problem. (Note that the subroutine establishes the flow zone of residence for the new position and then checks the position against all boundaries (1) in flow zone 1 (always) and (2) in the flow zone of residence for the new position.) Assuming that no problems have occurred in tracking the flow zone of residence (line 306), a test is performed (line 307) to see if a boundary violation has occurred. Assuming that no such violation has occurred (and that corrective measures left over from the resolution of such a violation are not pending, as determined in line 308), the time subincrement loop is repeated until all of the intermediate aeroelement positions have been determined and tested. The final aeroelement position and flow zone are stored in the aeroelement record by lines 453 to 456.

The handling of a detected boundary violation is begun with lines 375 to 445. A simple bisection loop is implemented to determine the position of violation to within a small variation (since real computations are very seldom exact). To begin this process, lines 375 to 402 are concerned with establishing points on both sides of the boundary (i.e., one point in violation and one point not in violation). Lines 404 to 423 perform a simple bisection search, where the bisected variable is time and where position is treated as parametric in time. Note that, as a safety precaution, this bisection process will always terminate after a fixed number of iterations even if the required resolution in time (line 405) has not been reached. Lines 442 to 445 effect adjustments to subroutine variables to cause the next principal position loop (lines 304 to 307) to position the aeroelement just short of the violated boundary (by adjusting the time span variable T), and to cause the course correction process to occur when (and if) that position has been successfully attained.

The course correction process is implemented by lines 309 to 355. The velocity of the aeroelement relative to the surface is established (lines 323 to 325), the velocity is adjusted by an elastic reflection method (lines 326 to 336), and the result is converted back to the absolute frame of reference (lines 337 and 338). If necessary (line 339), the component of aeroelement acceleration into the boundary (normal to the boundary, relative to the acceleration of the boundary) is set to zero by lines 340 to 350. The need to zero the normal acceleration component is determined if, during the bisection process that established the point of violation, the nonviolating point was never replaced with an updated value. Although no theoretical statement exists on this subject, this condition is presumed to indicate that the aeroelement is differentially close to the boundary already and cannot get up the necessary velocity to bounce off the boundary.

Thus, to move the aeroelement forward in time, it is necessary to disable its ability to violate the boundary by at least turning off its acceleration into the boundary.

This completes the discussion of the principal features of the aeroelement positioning process. The boundary description services have, for the most part, been submerged in calls to the YNM subroutine and in a few discrete calls to other boundary services. Listings of many of these boundary services are not included in this report since they are trivial invocations and recombinations of the fundamental boundary service features presented in listing 8.

Listing 8 presents the control point of all boundary services, the subroutine SURF. All requests to evaluate surface values, gradients, velocities, and accelerations are handled by this one, universal, fits-all routine. Control within this routine falls into three major categories: (1) specification of the desired result and transfer of control (lines 547 to 615), (2) handling of the surface truncation process (lines 622 to 719), and (3) fundamental surface evaluations (lines 724 to 752).

The desired-result-specification-and-transfer-of-control sections simply set various internal variables to turn various features of the SURF subroutine on or off. The variable ISW selects the computational results of the fundamental surface evaluation functions (of which listing 9, subroutine SA100, is an example). The variable NRTN controls the return of control from the fundamental surface evaluation section to the appropriate invoker. The surface gradient, velocity, and acceleration requests all ignore the surface truncation mechanisms and thus transfer control directly to the fundamental-surface-evaluation section and receive control directly back from that section when surface evaluation is complete. The desired-result-specification-and-transfer-of-control sections also provide a conversion from surface identification number to the surface header pointer, IP.

The fundamental-surface-evaluation section (lines 724 to 752) provides a simple calling mechanism to invoke the appropriate surface evaluation routine and to obtain the desired fundamental result (accomplished by lines 724 to 727). Once a base result is obtained, certain optional modifications may be made to it as invoked by the requesting section (lines 734 to 739). Finally control is transferred back to the invoking code section.

The surface-truncation-handling section is, by far, the most complicated area of code since it is, in fact, a recursive code written in a nonrecursive language. If a recursion of the truncation process is, in fact, occurring, lines 622 to 637 save the current surface context on a (software implemented) stack and initiate a new surface context by setting the surface description header pointer IP. If further truncations have not been inhibited (usually by selecting a result-specification-and-transfer-of-control section), a check is made to see if truncations to this surface exist (line 640). If no truncations exist, control is transferred to the fundamental-surface-evaluation section and, upon completion of the surface evaluation, the truncation-and-surface-evaluation process is unwound.

If one or more truncations do exist, the truncation recursion process is begun at a new level. Lines 641 to 650 perform various initializations associated with the truncation process, including locating a list of truncation headers that include descriptive information about each truncation. The code then loops through each truncation header (lines 651 and 652), obtains appropriate information from each truncation header in turn (lines 653 to 665), and recursively invokes the truncation process (line 666) to establish the surface value for each truncation surface. The recursion process continues until a truncation surface is found that does not, itself, list any further truncation surfaces. Thus the mechanism of truncating truncations to build up complex truncation surfaces through spatial concatenation can occur.

When a truncation surface evaluation finally does occur, control is returned to line 672, where a final surface value modification obtained from the truncation header is applied. This allows the user to use existing surfaces by flipping them side to side. The modified truncation surface value is inspected by line 673 to determine if a truncation has occurred. Truncation occurs only if the truncation surface value is definitely less than zero. Equality with zero is not considered a truncation so that flow zones (which are defined by a primary surface and its truncations) can be defined by using common separating surfaces (one such being just the flip side of the other) without creating a locus of points that does not lie in either flow zone (by having a separating surface value of exactly zero).

If surface truncation does not occur, lines 674 to 689 skip over the truncation header and transfer control back to the truncation-header-evaluation loop. If surface truncation does occur, the recursion process is unwound to the next level (lines 695 and 696) and control is transferred either back to the desired-result-specification-and-transfer-of-control section or back to the truncation-recursion-control section. The effect of transfer back to the recursion control section is to declare that since the truncation surface was itself truncated, it is therefore nonapropos and is passed over in the process of truncating its own parent surface.

Subroutine SA100 (listing 9) is a representative fundamental surface evaluation subroutine. This is the type of routine that is invoked by subroutine SURF (listing 8) in the fundamental-surface-evaluation section. This particular subroutine provides a time-invariant sphere in space. Four results are always provided by subroutines of this class as specified by the switch variable ISW: (1) the surface value (line 91), (2) the surface gradient (line 96), (3) the surface velocity (lines 98 and 99), and (4) the surface acceleration (in this case, since the surface is time invariant, also provided by lines 98 and 99). As can be seen by this example, fundamental surfaces are easy to describe and add to the CASPER repertoire.

The separation of surface control parameters for the fundamental-surface-evaluation functions into various groups (as commented on in lines 54 to 57 in listing 9, in the column labeled "'IV' slot") is also an important feature of the CASPER surface description system. It allows the easy repetition of groups of parameters (in this case, spheres with the same center or spheres with the same radius) simply by providing addressing information to the same actual parameter images. For spheres the economy of this is doubtful, but for more exotic surfaces (e.g., a realistic cambered and twisted fan blade that will be repeated 30 or 40 times in the same rotor) this feature can be a great help in avoiding errors in repetitive data entry.

Calculation of Thermodynamic Effects

The power of compression is calculated by subroutine POWERC, which is presented as listing 10. This routine calculates a straightforward parametric integration of the pressure-versus-volume history by passing parabolic approximations through three pressure points and three volume points (performed by lines 87 to 109) and stores the result in the aeroelement record (line 110). As it turns out, once the characteristic parametric spacings (in time, in this instance) are known, the computation can be reduced to a matrix-by-vector multiply operation to determine parabolic approximation coefficients (matrix determined by lines 87 to 94, coefficients determined by lines 96 to 107). Thus the matrix formulation can be performed once outside the actual aeroelement-by-aeroelement computation loop.

The power of viscous distortion is calculated in two phases for much the same reason as were the aeroelement accelerations—in this case derivatives of terms not yet calculated are required. Thus the appropriate term must be calculated and stored for each aeroelement before the phase involving the derivative of the term can begin. This process is carried out by subroutines WRKFLD and WRKFLE (listings 11 and 12, respectively). These subroutines are straighforward calculations of the power-of-viscous-distortion equation outlined in the theory section of this report. Lines 29 to 33 of listing 11 calculate and store the power-of-viscous-distortion result (which is the aeroelement volume times the divergence of the key term calculated in subroutine WRKFLD) by the standard derivative calculation scheme outlined previously.

The conductive heat transfer between aeroelements is calculated by subroutine WRKFLF (listing 13). The computation is a simple one implementing the conductive heat transfer equations discussed in the theoretical section of this report. It is important to note that the heat transfer is calculated only if the nearest neighbor of the target aeroelement reciprocates with the target aeroelement (i.e., if the nearest neighbor of the target aeroelement lists the target aeroelement as one of its nearest neighbors). Since the CASPER algorithm will not modify the record of the nearest-neighbor aeroelement (for parallel-processing reasons), this reciprocity is necessary to ensure the conservation of energy. In this way, it is ensured that the same amount of energy transferred into the target aeroelement will be removed from the neighbor aeroelement (when its turn comes in the computational process) to conserve the total energy between the two aeroelements.

The cleanup hitter in the thermodynamic subroutine lineup is subroutine WRKFLG (listing 14). This routine accumulates all of the power terms for an aeroelement, multiplies the result by the time increment to obtain a net work, divides the result by the coefficient of specific heat at constant volume to determine a change in temperature, adjusts the previous aeroelement temperature by the calculated change in temperature, and stores the result in the aeroelement record.

Calculation of Aeroelement Volume and Pressure

Subroutine VOL (listing 15) estimates aeroelement volumes on the basis of the proximity of nearest neighbors to the target aeroelement. Lines 273 to 283 initialize for the solid-angle correction calculation and calculate the initial spherical volume estimate for the target aeroelement. The solid angle calculation is performed by forming a regular spherical pentagon and dividing it into three successive spherical triangles. For this process to work, the nearest neighbors must be ordered to produce a regular spherical pentagon. This is accomplished in lines 287 to 312 by establishing a spherical pole with neighbor 1, establishing a line of zero longitude with neighbor 2, and reordering neighbors 3, 4, and 5 to achieve monotonically increasing longitudinal values. Lines 318 to 363 use the reordered neighbors to calculate the solid angle of the three spherical triangles so subtended and accumulates the results to obtain the total solid angle subtended by all five nearest neighbors. Finally the original volume estimate is adjusted (on line 369) by the relationship of the calculated solid angle to the current average solid angle, and the result is stored in the target aeroelement record (on line 374).

Subroutine VSUM (listing 16) implements the volume-estimate-correction scheme described in the theoretical section of this report. The volume estimate for each aeroelement is added to a total volume value. In subroutines not included in this report this total volume estimate is compared with the known volume for the problem and a multiplicative correction factor is produced. This factor is applied to the volume estimate for each aeroelement by subroutine VCOR (listing 17).

Subroutine RHOPRS (listing 18) calculates aeroelement density and pressure. The aeroelement density calculation is a trivial operation performed on lines 27 and 28 by dividing aeroelement mass by aeroelement volume and storing the result in the aeroelement record. The calculation of pressure is a similarly trivial process (lines 29 to 31); however, in this instance, it is performed by invoking the subroutine GASLAW (which is presented as listing 19). This allows the user to easily alter the operating gas law by the simple expedient of replacing the GASLAW subroutine. Thus any gas law relating pressure to density and temperature (and indeed any other recorded aeroelement property since the aeroelement identification number is passed to GASLAW) can be easily supported.

Finding Nearest-Neighbor Aeroelements

Only one example of a nearest-neighbor aeroelement finder is provided in this report: subroutine FNBNE (listing 20). This is the neighbor-tree-search algorithm described in the theoretical section of this report and is the most frequently used method of finding nearest neighbors in current CASPER practice. The principal advantage of this method is that the user sets the number of candidate aeroelements inspected by supplying the variable JCC in the subroutine call.

The subroutine starts out (lines 153 to 228) by constructing a candidate tree from the current neighbor tree for the target aeroelement. This is done in a two-phase process of (1) placing successive layers of the neighbor tree onto a candidate possibility stack (lines 190 to 195) and (2) removing aeroelements from the candidate possibility stack and inserting them in the candidate aeroelement tree (lines 174 to 181). This insertion process ensures that redundant entries will not occur.

The candidate-aeroelement-tree-building process continues until either enough independent aeroelement candidates have been found (i.e., more than JCC candidates have been found) or the neighbor tree fails to produce any new aeroelement candidates. In this second event the subroutine will attempt to fill out the candidate aeroelement tree by acquiring aeroelements from the flow zone of the target aeroelement (and thence from neighbor flow zones). This process is carried out by lines 207 to 228.

Once a candidate aeroelement tree of at least the requested size has been constructed, a simple center-to-center distance test is used to identify five nearest neighbors from among the candidates in the candidate aeroelement tree (lines 257 to 264 with neighbor list replacement being performed by lines 265 to 270). Note that lines 253 to 255 perform a candidate screening operation that attempts to eliminate the bogus-nearest-neighbor syndrome by requiring that each candidate aeroelement tested reside in either the flow zone of the target aeroelement or in a flow zone that is a neighbor to the flow zone of the target aeroelement.

Flow Field Simulation Sequence

This report does not include a listing of the flow field simulation sequence since this involves PAX parallel-processing knowledge. The interested reader is referred to reference 1, which includes a listing of the current CASPER flow field simulation sequence.

Concluding Remarks

This report has detailed the theoretical basis and illustrated the implementation of CASPER. No new insight into fluid flow is involved. Rather CASPER uses a simple visualization of the fundamental physics of the problem and shifts the burden from insight to computation and management. The results of CASPER operation are a simulation of Navier-Stokes flow rather than a solution of the Navier-Stokes equation.

CASPER allows the following key features in fluid flow simulation:

(1) Three-dimensional flow
(2) Viscous flow
(3) Time-varying flow
(4) Three-dimensional boundaries
(5) Nonsymmetric boundaries
(6) Arbitrary, complex boundaries through the truncation and concatenation of simple boundaries
(7) Time-varying boundaries
(8) Boundary-flow interactions

As currently implemented, CASPER tasks can be executed asynchronously in parallel. A Parallel, Asynchronous Executive (PAX) exists to manage this feature. Details of PAX are provided in a referenced report.

Computational results (four complete iterative cycles) have been obtained from CASPER. These results are sufficient to prove the syntactical correctness of CASPER modules but do not provide the necessary base from which a technical judgment about the appropriateness of the CASPER approach can be made.

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio, September 21, 1984

References

1. Jones, William H.: Parallel, Asynchronous Executive (PAX): System Concepts, Facilities, and Architecture. NASA TP-2179, 1983.
2. Li, Wen-Hsiung; and Lam, Sau-Hai: Principles of Fluid Mechanics. Addison-Wesley Publishing Co., Inc., 1964.
3. Schlichting, Hermann (J. Kestin, transl.): Boundary-Layer Theory. Sixth ed. McGraw-Hill Book Company, 1968, p. 253.

```
 @FOR,MS  CASPER2.INTFD
FOR   4R1   E -04/13/84-15:39:41 (2,)
>@EOF

   SUBROUTINE INTF      ENTRY POINT 000057


   STORAGE USED: CODE(1) 000074; DATA(0) 000015; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   CHKLH
    0004   CHKTIM
    0005   MATRIX
    0006   ERROR2
    0007   CLNINT
    0010   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000020 116G      0001   000044 23L      0001   000046 24L      0000 I 000000 FLAG1      0000 I 000004 I
    0000 I 000002 ID        0000 I 000003 IE        0000   000006 INJP$    0000 I 000001 IS


    00101                                                                                     000000
    00101                                                                                     000000
    00101                                                                                     000000
    00101                                                                                     000000
    00101                                                                                     000000
    00101                                                                                     000000
    00101                    SUBROUTINE INTF (IL,IH)                                201A0010  000000
    00101           C                                                               20100020  000000
    00101           C                                                               20100030  000000
    00101           C        INTF       ****** A SUBROUTINE FOR CASPER ******       20100040  000000
    00101           C                   AUTHOR     WILLIAM HENRY JONES              20100050  000000
    00101           C                   V01-00     29 NOV 76                        20100060  000000
    00101           C                   V01-01     23 AUG 78                        20110061  000000
    00101           C                   V01-01A    06 FEB 79                        201A0062  000000
    00101           C                   V01-01B    11 FEB 80      SINGULAR MATRICIES TO SAFS  000000
    00101           C                                                               20100080  000000
    00101           C                                                               20100090  000000
    00101           C        ARGUMENTS PASSED IN SUBROUTINE CALL ******             20110233  000000
    00101           C                                                               20110234  000000
    00101           C        ARGUMENT   TYPE       DIMENSION      DESCRIPTION       20110235  000000
    00101           C        ---------  ---------  ---------      ----------------- 20110236  000000
    00101           C        IL         INTEGER    SCALAR         LOW AEROELEMENT ID LIMIT   201A0240  000000
    00101           C                                                               201A0242  000000
    00101           C        IH         INTEGER    SCALAR         HIGH AEROELEMENT ID LIMIT  201A0244  000000
    00101           C                                                               201A0246  000000
    00101           C                                                               20100260  000000
    00101           C        DESCRIPTION:                                           20100270  000000
    00101           C                                                               20100280  000000
    00101           C        1   CLEARS FLAG1                                       20100290  000000
    00101           C                                                               20100300  000000
    00101           C        2   FOR I1=IL,IH, CALLS 'MATRIX'                       201A0310  000000
    00101           C                                                               20100320  000000
    00101           C        3   EXITS                                              20100330  000000
    00101           C                                                               20100340  000000
    00101           C        REQUIRED SUBROUTINES:                                  20100350  000000
    00101           C                                                               20100360  000000
    00101           C        211        MATRIX                                      20110370  000000
    00101           C                                                               20100380  000000
    00101           C                                                               20100390  000000
    00103                    INTEGER FLAG1,IL,IH,IS,ID,IE                           201A0400  000000
    00104                    DATA ID/201/                                           201A0405  000000
    00106                    DATA IE/1/                                             201A0410  000000
    00110                    CALL CHKLH (IL,IH,IS,ID,IE)      @ CHECK AEROELEMENT ID RANGE  201A0420  000000
    00111                    IF (IS) 16,24,16                 @ VALID RANGE ?       201A0430  000006
    00114            16      FLAG1=0                          @ YES, ZERO FLAG      201A0440  000010
    00115                    DO 23 I=IL,IH,IS                 @ DO THE LOOP         201A0450  000020
    00120                    CALL CHKTIM (IL,IH,I)            @ KEEP AN EYE ON THE TIME       000020
    00121                    CALL MATRIX (I,FLAG1)            @ CALC A MATRIX                 000025
    00122                    IF (FLAG1) 20,23,20              @ SINGULARITY FAULT ?           000031
    00125            20      FLAG1=0                          @ YES, RESET THE FLAG           000033
    00126                    CALL ERROR2 (ID,5)               @ REPORT IT                     000034
    00127                    CALL CLNINT (I)                  @ FIX IT                        000040
    00130            23      CONTINUE                         @                               000046
    00132            24      RETURN                           @                               000046
    00133                    END                                                    20100480  000073
    END FOR
    >
```

Listing 1.—Subroutine INTF.

```
 @FOR,MS  CASPER2.MATRID
FOR   4R1   E -04/13/84-15:41:12 (0,)
>@EOF

   SUBROUTINE MATRIX    ENTRY POINT 000311

   STORAGE USED: CODE(1) 000335; DATA(0) 000174; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    ISIZEC 000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    X
    0005    NEI
    0006    DET4
    0007    STAT
    0010    STSTAT
    0011    STINTP
    0012    XPII
    0013    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000003 110G      0001    000006 113G      0001    000040 121G      0001    000052 127G      0001    000053 132G
    0001    000065 140G      0001    000075 143G      0001    000076 146G      0001    000113 154G      0001    000152 163G
    0001    000167 176G      0001    000226 203L      0001    000201 206G      0001    000234 223G      0001    000237 226G
    0001    000155 404L      0001    000200 406L      0001    000207 601L      0000 R 000000 A          0000 R 000031 COF
    0000 R 000132 DET        0006 R 000000 DET4       0000 R 000062 D4         0000 R 000102 D4I        0000 I 000126 I
    0000    000152 INJP$     0003    000000 ISIZE     0000 I 000127 J          0000 I 000130 K          0000 I 000131 L
    0005 I 000000 NEI        0000 I 000133 NNN        0000 R 000134 ODET       0000 R 000135 RRR        0007 R 000000 STAT
    0004 R 000000 X


00101                                                                                         000006
00101                                                                                         000006
00101                                                                                         000006
00101                                                                                         000006
00101                                                                                         000006
00101                                                                                         000006
00101         SUBROUTINE MATRIX (I1,FLAG1)                                        21150010    000006
00101    C                                                                        21100020    000006
00101    C                                                                        21100030    000006
00101    C    MATRIX      ****** A SUBROUTINE FOR CASPER ******                   21100040    000006
00101    C                AUTHOR      WILLIAM HENRY JONES                         21100050    000006
00101    C                V01-00      29 NOV 76                                   21100060    000006
00101    C                V01-01                                                  21110061    000006
00101    C                V01-02                                                  21120062    000006
00101    C                V01-03                                                  21130063    000006
00101    C                V01-04                                                  21140064    000006
00101    C                V01-05      23 AUG 78                                   21150065    000006
00101    C                                                                        21100080    000006
00101    C    ARGUMENTS IN CASPER 'CACHE' MEMORY ******                           21150084    000006
00101    C                                                                        21150086    000006
00101    C    ARGUMENT    TYPE        DIMENSIONS       DESCRIPTION                21100090    000006
00101    C    ---------   ---------   ---------       ------------------          21150095    000006
00101    C    X           REAL        1 TO ISIZE       X,Y,Z, COORDINATES OF      21100100    000006
00101    C                            1 TO 3             THE AEROELEMENTS         21100110    000006
00101    C                                                                        21100120    000006
00101    C    INTP        REAL        1 TO ISIZE       FIRST 3 ROWS OF INTERPOL21150130    000006
00101    C                            1 TO 3             ATOR FROM 2ND ORDER      21100140    000006
00101    C                            1 TO 5             TAYLOR SERIES            21100150    000006
00101    C                                                                        21100160    000006
00101    C    NEI         INTEGER     1 TO ISIZE       INTEGER LIST OF NEAREST    21100170    000006
00101    C                            1 TO 5             NEIGHBORS (NN)           21100180    000006
00101    C                                                                        21100190    000006
00101    C    STAT        INTEGER     1 TO ISIZE       AEROELEMENT STATUS LIST    21100200    000006
00101    C                                                                        21100210    000006
00101    C                                                                        21150213    000006
00101    C    ARGUMENTS PASSED IN SUBROUTINE CALL ******                          21150214    000006
00101    C                                                                        21150215    000006
00101    C    ARGUMENT    TYPE        DIMENSION     DESCRIPTION                   21150216    000006
00101    C    ---------   ---------   ---------     ---------------------         21150217    000006
00101    C    I1          INTEGER     SCALAR           POINTER TO CURRENT ELEM.21100220    000006
```

Listing 2.—Subroutine MATRIX.

```
00101   C                                                                      21100230   000006
00101   C     FLAG1        INTEGER     SCALAR           SINGULAR MATRIX FLAG1=1 21100240   000006
00101   C                                                                      21100250   000006
00101   C                                                                      21100260   000006
00101   C     DESCRIPTION:                                                     21100270   000006
00101   C                                                                      21100280   000006
00101   C     1   MATRIX FORMS THE ARRAY (CALLED A IN THE PROGRAM)             21100290   000006
00101   C                                                                      21100300   000006
00101   C         $X11       $X12       $X13       .5*$X11**2  $X12*$X13       21100310   000006
00101   C         $X21       $X22       $X23       .5*$X21**2  $X22*$X23       21100320   000006
00101   C         $X31       $X32       $X33       .5*$X31**2  $X32*$X33       21100330   000006
00101   C         $X41       $X42       $X43       .5*$X41**2  $X42*$X43       21100340   000006
00101   C         $X51       $X52       $X53       .5*$X51**2  $X52*$X53       21100350   000006
00101   C                                                                      21100360   000006
00101   C        WHERE $XKJ = X(NEI(I1,K),J) - X(I1,J)                         21100370   000006
00101   C                                                                      21100380   000006
00101   C     2   MATRIX CALCULATES THE APPROPRIATE COFACTORS AN THE           21100390   000006
00101   C           DETERMINANT OF A                                           21100400   000006
00101   C                                                                      21100410   000006
00101   C     3   IF THE DETERMINANT OF A = 0 THEN MATRIX SETS FLAG1=1, SETS   21140420   000006
00101   C           BIT 0 OF STAT AND EXITS                                    21100430   000006
00101   C                                                                      21100440   000006
00101   C     4   MATRIX CALCULATES INVERSION OF A (FIRST 3 ROWS ONLY), STORES 21100450   000006
00101   C           IT INTO INT(I1, , ), CLEARS BIT 0 OF STAT (THE INTERPOLATOR 21140460  000006
00101   C           NOT AVAILABLE BIT), AND EXITS                              21140470   000006
00101   C                                                                      21100480   000006
00101   C     REQUIRED SUBROUTINES ******                                      21150490   000006
00101   C                                                                      21150492   000006
00101   C     213        DET3       212        DET4                            21150494   000006
00101   C     401        AVIRI      402        X                               21150498   000006
00101   C     417        STSTAT     416        STAT                            21150502   000006
00101   C                           422        NEI                             21150506   000006
00101   C     425        STINTP                                                24450508   000006
00101   C                                                                      21100520   000006
00103         COMMON /ISIZEC/ISIZE                                             21100530   000006
00104         INTEGER I1,FLAG1                                                 21150550   000006
00105         DIMENSION A(5,5),COF(5,5),D4(4,4),D4I(5,4)                       21120570   000006
00106         DEFINE DX(I,J)=X(I,J)                                            21100580   000006
00107         DO 201 I=1,5                          @CALCULATE ARRAY A         21100590   000006
00112         DO 201 J=1,3                                                     21100600   000006
00115   201   A(I,J)=DX(NEI(I1,I),J)-X(I1,J)                                   21100610   000006
00120         DO 202 I=1,5                                                     21100620   000040
00123         A(I,4)=0.5*A(I,1)*A(I,1)                                         21100630   000040
00124   202   A(I,5)=A(I,2)*A(I,3)                                             21100640   000043
00126         DO 401 I=1,5                          @SETUP INTERMEDIATE STORE  21120650   000053
00131         DO 401 J=1,4                          @  FOR FIRST PASS          21120660   000053
00134   401   D4I(I,J)=A(I,J+1)                                                21120670   000053
00137         DO 408 J=1,3                          @BEGIN COLUMN LOOP         21120680   000065
00142         DO 402 K=1,4                          @SETUP 4X4 MATRIX FOR FIRST 21120690  000076
00145         DO 402 L=1,4                          @  PASS                    21120700   000076
00150   402   D4(K,L)=D4I(K+1,L)                                               21120710   000076
00153         DO 404 I=1,5                                                     21120720   000113
00156         COF(I,J)=DET4(D4)*((-1)**(I+J))       @CALCULATE COFACTORS       21120730   000121
00157         IF (I-5)     ,404,                    @IF I=5 SKP SETUP FOR NEXTPASS21120740 000141
00162         DO 403 L=1,4                          @SETUP 4X4 FOR NEXT PASS   21120750   000152
00165   403   D4(I,L)=D4I(I,L)                                                 21120760   000152
00167   404   CONTINUE                                                         21120770   000157
00171         IF (J-1) 406,    ,406                 @IF FIRST COLUMN COMPLETED 21120780   000157
00174         DET=0.0                               @  CALCULATE DETERMINANT ' ' 21120790 000162
00175         DO 405 K=1,5                                                     21120800   000167
00200   405   DET=DET+A(K,J)*COF(K,J)                                          21120810   000167
00202         IF (ABS(DET)-1.0E-10) 601,406,406                                21130820   000173
00205   406   DO 407 K=1,5                          @SWAP INTERNEDIATE COLUMN  21120830   000201
00210   407   D4I(K,J)=A(K,J)                                                  21120840   000201
00212   408   CONTINUE                              @EOL COLUMN LOOP           21120850   000205
00214         GO TO 203                                                        21120860   000205
00215   601   FLAG1=1                               @ERROR - SET FLAG1         21120870   000207
00216         NNN=OR(STAT(I1),2**0)                 @ SET STATUS BIT 0         21150880   000210
00217         CALL STSTAT (I1,NNN)                  @                          21150850   000216
00220         RETURN                                                           21120890   000222
00221   203   ODET=1.0/DET                                                     21121980   000226
00221   C     FORM INVERSE OF MATRIX BY DIVIDING TRANSFORM OF COFACTOR         21101990   000226
00221   C       MATRIX BY DETERMINANT                                          21102000   000226
00222         DO 204 I=1,5                                                     21102010   000237
00225         DO 204 J=1,3                                                     21102020   000237
00230         RRR=ODET*COF(I,J)                     @                          21152030   000237
00231   204   CALL STINTP (I1,J,I,RRR)              @                          21152035   000242
00234         NNN=AND(STAT(I1),COMPL(2**0))         @ CLR STATUS BIT 0         21152040   000257
00235         CALL STSTAT (I1,NNN)                  @                          21152045   000265
00236         RETURN                                                           21102050   000271
00237         END                                                              21102060   000334
END FOR
>
```

Listing 2.—Concluded.

```
 @FOR,MS  CASPER2.DET4D
FOR   4R1   E -04/13/84-15:43:57 (0,)
>@EOF

   FUNCTION DET4        ENTRY POINT 000123

   STORAGE USED: CODE(1) 000135; DATA(0) 000043; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   DET3
    0004   XPII
    0005   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000012 105G      0001   000013 110G      0001   000031 117G      0001   000075 126G      0001   000100 203L
    0003 R 000000 DET3       0000 R 000000 DET4      0000 R 000014 DE4       0000 R 000001 D3        0000 I 000012 I
    0000    000021 INJP$     0000 I 000013 J
```

```
00101                                                                                000013
00101                                                                                000013
00101                                                                                000013
00101                                                                                000013
00101                                                                                000013
00101                                                                                000013
00101              REAL FUNCTION DET4 (D4)                                21200010   000013
00101       C                                                             21200020   000013
00101       C                                                             21200030   000013
00101       C                ****** A REAL FUNCTION FOR CASPER ******     21200040   000013
00101       C                AUTHOR      WILLIAM HENRY JONES              21200050   000013
00101       C                DATE        13 DEC 76                        21200060   000013
00101       C                REVISION    ORIGINAL                         21200070   000013
00101       C                                                             21200080   000013
00101       C      ARGUMENT  TYPE        DIMENSION     DESCRIPTION        21200090   000013
00101       C                                                             21200100   000013
00101       C      D4        REAL        1 TO 4        ARBITRARY 4 X 4 ARRAY 21200110 000013
00101       C                            1 TO 4                           21200120   000013
00101       C                                                             21200130   000013
00101       C      DESCRIPTION:                                           21200140   000013
00101       C                                                             21200150   000013
00101       C      CALCULATES THE DETERMINANT OF THE INPUT MATRIX BY WEIGHTED 21200160 000013
00101       C        ACCUMULATION OF FUNCTION CALLS TO DET3 (3 X 3 DETERMINANT) 21200170 000013
00101       C                                                             21200180   000013
00101       C      REQUIRED SUBROUTINES:                                  21200190   000013
00101       C                                                             21200200   000013
00101       C                REAL FUNCTION DET3 (D)                       21200210   000013
00101       C                                                             21200220   000013
00101       C                                                             21200230   000013
00101       C                                                             21200240   000013
00103              REAL D4(4,4),D3(3,3)                                   21200250   000013
00104              DO 201 I=1,3                                           21200260   000013
00107              DO 201 J=1,3                                           21200270   000013
00112         201  D3(I,J)=D4(I+1,J+1)      @SETUP FIRST SUBMATRIX        21200280   000013
00115              DE4=0.0      @CLR ACCUMULATOR                          21200290   000024
00116              DO 203 J=1,4             @ENTER ACCUMULATOR LOOP       21200300   000031
00121              DE4=DE4+DET3(D3)*D4(1,J)*((-1)**(J+1))                 21200310   000042
00122              IF (J-4)     ,203,       @TST FOR SKP NEXT SUBMATRIX   21200320   000064
00125              DO 202 I=1,3             @SUBSTITUTE CHANGING COLUMN   21200330   000075
00130         202  D3(I,J)=D4(I+1,J)        @  FOR NEXT ACCUMULATOR PASS  21200340   000075
00132         203  CONTINUE                                               21200350   000102
00134              DET4=DE4                                               21200360   000102
00135              RETURN                                                 21200370   000104
00136              END                                                    21200380   000134
END FOR
```

Listing 3.—Subroutine DET4.

```
>@FOR,MS  CASPER2.DET3D
FOR   4R1   E -04/13/84-15:45:11 (0,)
>@EOF

   FUNCTION DET3        ENTRY POINT 000036

   STORAGE USED: CODE(1) 000041; DATA(0) 000006; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000 R 000000 DET3      0000    000001 INJP$

00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101                                                                          000000
00101            REAL FUNCTION DET3 (D)                              21300010  000000
00101       C                                                        21300020  000000
00101       C                                                        21300030  000000
00101       C*   DET3      ****** A REAL FUNCTION FOR CASPER ******  21300040  000000
00101       C              AUTHOR    WILLIAM HENRY JONES             21300050  000000
00101       C              DATE      13 DEC 76                       21300060  000000
00101       C              REVISION  ORIGINAL                        21300070  000000
00101       C                                                        21300080  000000
00101       C    ARGUMENT  TYPE      DIMENSIONS     DESCRIPTION      21300090  000000
00101       C                                                        21300100  000000
00101       C    D         REAL      1 TO 3         ARBITRARY 3 X 3 ARRAY 21300110  000000
00101       C                        1 TO 3                          21300120  000000
00101       C                                                        21300130  000000
00101       C    DESCRIPTION:                                        21300140  000000
00101       C                                                        21300150  000000
00101       C    FORMS THE DETERMINANT OF AN ARBITRARY 3 X 3 ARRAY   21300160  000000
00101       C                                                        21300170  000000
00101       C    REQUIRED SUBROUTINES:                               21300180  000000
00101       C                                                        21300190  000000
00101       C              NONE                                      21300200  000000
00101       C                                                        21300210  000000
00101       C                                                        21300220  000000
00101       C                                                        21300230  000000
00103            REAL D(3,3)                                         21300240  000000
00104            DET3=D(1,1)*(D(2,2)*D(3,3)-D(2,3)*D(3,2))-D(2,1)*(D(1,2)*D(3,3)-D(21300250  000000
00104           11,3)*D(3,2))+D(3,1)*(D(1,2)*D(2,3)-D(1,3)*D(2,2))   21300260  000000
00105            RETURN                                              21300270  000025
00106            END                                                 21300280  000040
END FOR
>
```

Listing 4.—Subroutine DET3.

```
 @FOR,MS  CASPER3.STOK1D
FOR   4R1    E -04/13/84-15:04:44 (4,)
>@EOF

   SUBROUTINE STOK1      ENTRY POINT 000271

   STORAGE USED: CODE(1) 000316; DATA(0) 000131; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   STAT
    0004   MU
    0005   MUTHET
    0006   INTP
    0007   U
    0010   EIJ
    0011   CHKLH
    0012   NEI
    0013   STEIJ
    0014   STMUTH
    0015   CHKTIM
    0016   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000017 132G      0001   000032 140G      0001   000042 144G      0001   000057 153G      0001   000072 161G
    0001   000074 164G      0001   000120 172G      0001   000122 175G      0001   000124 201G      0001   000202 205L
    0001   000147 210G      0001   000241 212L      0001   000152 213G      0001   000214 232G      0001   000250 701L
    0000 R 000044 DELTU     0000 R 000063 DUIDXJ    0010 R 000000 EIJ       0000 I 000075 I         0000 I 000001 ID
    0000 I 000002 IE        0000   000110 INJP$     0006 R 000000 INTP      0000 I 000000 IS        0000 I 000074 I1
    0000 I 000076 J         0000 I 000077 K         0000 R 000033 LEIJ      0000 R 000003 LINTP     0000 R 000025 LMUTHE
    0000 I 000026 LNEI      0000 R 000022 LU        0004 R 000000 MU        0005 R 000000 MUTHET    0012 I 000000 NEI
    0000 R 000100 RRR       0003 I 000000 STAT      0007 R 000000 U


00101                                                                                000000
00101                                                                                000000
00101                                                                                000000
00101                                                                                000000
00101                                                                                000000
00101                                                                                000000
00101                                                                                000000
00101          SUBROUTINE STOK1 (IL,IH)                                    301A0010  000000
00101      C                                                               30100020  000000
00101      C                                                               30100030  000000
00101      C               ****** A SUBROUTINE FOR CASPER ******           30100040  000000
00101      C               AUTHOR      WILLIAM HENRY JONES                 30100050  000000
00101      C               V01-00      28 DEC 76                           30100060  000000
00101      C               V01-01                                          30110061  000000
00101      C               V01-02                                          30120062  000000
00101      C               V01-03      24 AUG 78                           30130063  000000
00101      C               V01-03A     06 FEB 79     FIRST SPLIT OF 'STOK' 301A0064  000000
00101      C               V01-03B     11 FEB 80     TIME CHECK                      000000
00101      C               V01-03C     15 SEP 80     FUNCTION TYPE STATEMENTS        000000
00101      C               V01-03D     15 APR 83     FUNCTION TYPES, EFFICIENCY V01-03D  000000
00101      C                                                               30100080  000000
00101      C   ARGUMENTS IN CASPER 'CACHE' MEMORY ******                   30130084  000000
00101      C                                                               30130086  000000
00101      C   ARGUMENT    TYPE        DIMENSION       DESCRIPTION         30100090  000000
00101      C   ---------   ---------   ---------   ------------------      30130100  000000
00101      C   U           REAL        1 TO ISIZE      AEROELEMENT VELOCITIES 30100110  000000
00101      C                           1 TO 3                              30100120  000000
00101      C                                                               30100130  000000
00101      C   NEI         INTEGER     1 TO ISIZE      NEAREST NEIGHBOR LIST  30100140  000000
00101      C                           1 TO 5                              30100150  000000
00101      C                                                               30100160  000000
```

Listing 5.—Subroutine STOK1.

```
00101      C    INTP         REAL         1 TO ISIZE     INTERPOLATOR MATRICIES     30130170    000000
00101      C                              1 TO 3                                    30100180    000000
00101      C                              1 TO 5                                    30100190    000000
00101      C                                                                        30100200    000000
00101      C    STAT         INTEGER      1 TO ISIZE       STATUS LIST              30100210    000000
00101      C                                                                        30100220    000000
00101      C    MU           REAL         1 TO ISIZE       AEROELEMENT VISCOUSITIES30100270    000000
00101      C                              1 TO 2              1=FIRST, 2=SECOND     30100280    000000
00101      C                                                                        30100290    000000
00101      C    EIJ          REAL         1 TO ISIZE       STRAIN RATE MATRICIES    30100330    000000
00101      C                              1 TO 3              EIJ(I,J)=.5*(U(I),X(J)30100340    000000
00101      C                              1 TO 3                 +U(J),X(I))        30100350    000000
00101      C                                                                        30100360    000000
00101      C    MUTHET       REAL         1 TO ISIZE       SEE DESCRIPTION NOTE 3   30100370    000000
00101      C                                                                        301A0380    000000
00101      C    ARGUMENTS PASSED IN SUBROUTINE CALL ******                          301A0382    000000
00101      C                                                                        301A0384    000000
00101      C    ARGUMENT     TYPE         DIMENSION      DESCRIPTION                301A0386    000000
00101      C    ---------    ---------    ---------      ------------------         301A0388    000000
00101      C    IL           INTEGER      SCALAR         AEROELEMENT ID LOW LIMIT   301A0390    000000
00101      C                                                                        301A0392    000000
00101      C    IH           INTEGER      SCALAR         AEROELEMENT ID HIGH LIMIT  301A0394    000000
00101      C                                                                        301A0396    000000
00101      C                                                                        301A0398    000000
00101      C    DESCRIPTION ******                                                  301A0400    000000
00101      C                                                                        301A0402    000000
00101      C    THIS SUBROUTINE PERFORMS THE FOLLOWING OPERATIONS FOR EACH          30100410    000000
00101      C    AEROELEMENT IN THE RANGE 'IL' TO 'IH'.                              301A0420    000000
00101      C                                                                        30100430    000000
00101      C    1  A CHECK IS MADE FOR AVAILABILITY OF THE INTERPOLATOR MATRIX.     30100460    000000
00101      C       IF NOT AVAILABLE ALL SUCCEEDING OPERATIONS ARE SKIPPED FOR       30100470    000000
00101      C       THE AEROELEMENT DURING THE PASS                                  30100480    000000
00101      C                                                                        30100490    000000
00101      C    2  THE STRAIN RATE MATRIX IS CALCULATED (EIJ(I,J))                  30100500    000000
00101      C                                                                        30100510    000000
00101      C    3  THE QUANTITY (MU'-.6666667*MU)*(EIJ(1,1)+EIJ(2,2)+EIJ(3,3)) IS   30100520    000000
00101      C       CALCULATED                                                       30100530    000000
00101      C                                                                        30100540    000000
00101      C    REQUIRED SUBROUTINES ******                                         30130650    000000
00101      C                                                                        30130652    000000
00101      C    401          AVIRI                                                  30130654    000000
00101      C                              404            U                          30130656    000000
00101      C    407          STA                                                    30130658    000000
00101      C                              408            P                          30130660    000000
00101      C    417          STSTAT       416            STAT                       30130662    000000
00101      C                              422            NEI                        30130664    000000
00101      C                              424            INTP                       30130666    000000
00101      C    427          STEIJ        426            EIJ                        30130668    000000
00101      C                              428            MU                         30130670    000000
00101      C    431          STMUTH       430            MUTHET                     30130672    000000
00101      C                                                                        30100680    000000
00101      C                                                                        30100690    000000
00101      C                                                                        30100700    000000
00103           INTEGER IL,IH,IS,ID,IE                                          301A0710    000000
00104           INTEGER STAT                                                                000000
00105           REAL MU,MUTHET                                                              000000
00106           REAL INTP             @ INTERPOLATOR MATRIX ACCESS FUNCTION     V01-03D     000000
00107           REAL LINTP(5,3)       @ LOCAL COPY OF INTERPOLATOR              V01-03D     000000
00110           REAL LU(3)            @ LOCAL COPY OF ELEMENT VELOCITY          V01-03D     000000
00111           REAL LMUTHE           @ LOCAL COPY OF MUTHET FACTOR             V01-03D     000000
00112           INTEGER LNEI(5)       @ LOCAL COPY OF NEAREST NEIGHBOR IDS      V01-03D     000000
00113           REAL LEIJ(3,3)        @ LOCAL COPY OF STRAIN RATE MATRIX        V01-03D     000000
00114           DATA ID/301/                                                    301A0712    000000
00116           DATA IE/1/                                                      301A0714    000000
00120           REAL DELTU(5,3),DUIDXJ(3,3)                                     30110750    000000
00121           DEFINE DU(I,J)=U(I,J)                                           30100780    000000
00122           DEFINE DMUTHE(I)=MUTHET(I)                                      30100800    000000
00123           DEFINE DMU(I,J)=MU(I,J)                                         30100810    000000
00124           DEFINE DEIJ(I,J,K)=EIJ(I,J,K)                                   30100820    000000
00125           CALL CHKLH (IL,IH,IS,ID,IE)      @ GO CHECK AEROELEMENT RANGE   301A0830    000000
00126           IF (IS) 101,701,101              @ LEGAL RANGE ?                301A0832    000006
00131      101  DO 213 I1=IL,IH,IS               @ YES, DO EACH AEROELEMENT     V01-03D     000010
```

Listing 5.—Continued.

```
00134          IF (AND(STAT(I1),1)) 103,103,212     @ DO WE HAVE AN INTERPOLATOR ?V01-03D    000017
00137     103  DO 107 I=1,3                         @ YES, LOCALIZE             V01-03D    000032
00142          LU(I)=U(I1,I)                        @  AEROELEMENT VELOCITY     V01-03D    000032
00143          DO 107 J=1,5                         @                           V01-03D    000042
00146          LINTP(J,I)=INTP(I1,I,J)              @  TRANSPOSE OF INTERPOLATOR  V01-03D  000042
00147     107  CONTINUE                             @                           V01-03D    000057
00152          DO 110 I=1,5                         @                           V01-03D    000057
00155          LNEI(I)=NEI(I1,I)                    @  NEAREST NEIGHBOR LIST    V01-03D    000057
00156     110  CONTINUE                             @                           V01-03D    000074
00160          DO 201 I=1,5                         @CALCULATE VELOCITY DIFFERENCE30100850 000074
00163          DO 201 J=1,3                         @  FOR ALL 5 NN IN ALL 3    30120860   000074
00166     201  DELTU(I,J)=DU(LNEI(I),J)-LU(J)       @  COORDINATE DIRECTIONS    V01-03D    000074
00171          DO 202 I=1,3                         @CLR DERIVATIVE U(I),X(J)   30100880   000122
00174          DO 202 J=1,3                         @  STORAGE AND THEN         30100890   000122
00177          DUIDXJ(I,J)=0.0                      @  CALCULATE DERIVATIVE     30100900   000122
00200          DO 202 K=1,5                         @  BY INTERPOLATOR          30100910   000124
00203     202  DUIDXJ(I,J)=DUIDXJ(I,J)+(LINTP(K,J)*DELTU(K,I)) @                V01-03D    000124
00207          DO 205 I=1,3                         @ CALC STRAIN RATE MATRIX   V01-03D    000152
00212          DO 205 J=1,3                         @                           V01-03D    000152
00215          RRR=0.5*(DUIDXJ(I,J)+DUIDXJ(J,I))    @                           30130950   000152
00216          CALL STEIJ (I1,I,J,RRR)              @                           30130955   000156
00217          LEIJ(I,J)=RRR                        @ SAVE LOCAL COPY           V01-03D    000164
00220          IF (I-J) 203,205,203                 @ OFF-DIAGONAL ELEMENT ?    V01-03D    000166
00223     203  CALL STEIJ(I1,J,I,RRR)               @ YES, FORCE A SYMETRIC     V01-03D    000171
00224          LEIJ(J,I)=RRR                        @  RESULT                   V01-03D    000177
00225     205  CONTINUE                             @                           V01-03D    000210
00230          LMUTHE=0.0                           @ CALC QUANTITY OF NOTE 3   V01-03D    000210
00231          DO 209 I=1,3                         @                           V01-03D    000214
00234          LMUTHE=LMUTHE+LEIJ(I,I)              @                           V01-03D    000214
00235     209  CONTINUE                             @                           V01-03D    000217
00237          LMUTHE=LMUTHE*(MU(I1,2)-(0.666667*MU(I1,1))) @                   V01-03D    000217
00240          CALL STMUTH (I1,LMUTHE)              @                           V01-03D    000234
00241     212  CALL CHKTIM (IL,IH,I1)               @ KEEP AN EYE ON THE TIME   V01-03D    000241
00242     213  CONTINUE                             @                           V01-03D    000250
00244     701  RETURN                               @ BACK TO CALLER            301A1370   000250
00245          END                                                              30101380   000315
END FOR
>
```

Listing 5.—Concluded.

```
@FOR,MS  CASPER3.STOK2D
FOR   4R1   E -04/13/84-15:07:47 (4,)
>@EOF

   SUBROUTINE STOK2     ENTRY POINT 000424

   STORAGE USED: CODE(1) 000451; DATA(0) 000214; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   STAT
    0004   MU
    0005   MUTHET
    0006   INTP
    0007   U
    0010   P
    0011   EIJ
    0012   CHKLH
    0013   RHO
    0014   STSTAT
    0015   NEI
    0016   STA
    0017   CHKTIM
    0020   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000044 107L     0001   000020 140G     0001   000062 162G     0001   000075 170G     0001   000100 173G
    0001   000116 202G     0001   000121 205G     0001   000136 214G     0001   000372 220L     0001   000160 225G
    0001   000174 232G     0001   000177 236G     0001   000211 244G     0001   000225 251G     0001   000230 255G
    0001   000243 263G     0001   000246 266G     0001   000251 271G     0001   000310 300G     0001   000311 303G
    0001   000313 307G     0001   000342 316G     0001   000344 322G     0001   000355 330G     0001   000403 701L
    0000 R 000153 DEJDXI   0000 R 000052 DELT     0000 R 000065 DELTM    0000 R 000142 DMEDXI   0000 R 000057 DPDXJ
    0000 R 000062 DTDXJ    0011 R 000000 EIJ      0000 I 000157 I        0000 I 000001 ID       0000 I 000002 IE
    0000   000174 INJP$    0006 R 000000 INTP     0000 I 000000 IS       0000 I 000156 I1       0000 I 000160 J
    0000 I 000161 K        0000 R 000040 LEIJ     0000 R 000013 LINTP    0000 R 000032 LMU      0000 R 000051 LMUTHE
    0000 I 000003 LNEI     0000 R 000033 LNMU     0000 R 000011 LP       0000 R 000012 LRHO     0000 I 000010 LSTAT
    0004 R 000000 MU       0005 R 000000 MUTHET   0015 I 000000 NEI      0010 R 000000 P        0013 R 000000 RHO
    0000 R 000162 RRR      0003 I 000000 STAT     0007 R 000000 U


00101                                                                                  000000
00101                                                                                  000000
00101                                                                                  000000
00101                                                                                  000000
00101                                                                                  000000
00101                                                                                  000000
00101                                                                                  000000
00101              SUBROUTINE STOK2 (IL,IH)                                  302A0010  000000
00101       C                                                                30100020  000000
00101       C                                                                30100030  000000
00101       C                 ****** A SUBROUTINE FOR CASPER ******          30100040  000000
00101       C                 AUTHOR     WILLIAM HENRY JONES                 30100050  000000
00101       C                 V01-00     28 DEC 76                           30100060  000000
00101       C                 V01-01                                         30110061  000000
00101       C                 V01-02                                         30120062  000000
00101       C                 V01-03     24 AUG 78                           30130063  000000
00101       C                 V01-03A    06 FEB 79     SECOND SPLIT OF 'STOK'  302A0064  000000
00101       C                 V01-03B    11 FEB 80     TIME CHECK                      000000
00101       C                 V01-03C    15 SEP 80     FUNCTION TYPE STATEMENTS        000000
00101       C                 V01-03D    15 APR 83     FUNCTION TYPES, EFFICIENCY V01-03D  000000
00101       C                                                                30100080  000000
00101       C      ARGUMENTS IN CASPER 'CACHE' MEMORY ******                 30130084  000000
00101       C                                                                30130086  000000
00101       C      ARGUMENT    TYPE        DIMENSION      DESCRIPTION        30100090  000000
00101       C      ---------   ---------   ---------   ------------------    30130100  000000
```

Listing 6.—Subroutine STOK2.

```
00101   C     NEI        INTEGER     1 TO ISIZE       NEAREST NEIGHBOR LIST   30100140   000000
00101   C                            1 TO 5                                   30100150   000000
00101   C                                                                     30100160   000000
00101   C     INTP       REAL        1 TO ISIZE     INTERPOLATOR MATRICIES    30130170   000000
00101   C                            1 TO 3                                   30100180   000000
00101   C                            1 TO 5                                   30100190   000000
00101   C                                                                     30100200   000000
00101   C     STAT       INTEGER     1 TO ISIZE       STATUS LIST             30100210   000000
00101   C                                                                     30100220   000000
00101   C     P          REAL        1 TO ISIZE       AEROELEMENT PRESSURES   30100230   000000
00101   C                                                                     30100240   000000
00101   C     RHO        REAL        1 TO ISIZE       AEROELEMENT DENSITIES   30100250   000000
00101   C                                                                     30100260   000000
00101   C     MU         REAL        1 TO ISIZE       AEROELEMENT VISCOUSITIES30100270   000000
00101   C                            1 TO 2             1=FIRST, 2=SECOND     30100280   000000
00101   C                                                                     30100290   000000
00101   C     A          REAL        1 TO ISIZE       AEROELEMENT ACCELERATION30100300   000000
00101   C                            1 TO 3                                   30100310   000000
00101   C                                                                     30100320   000000
00101   C     EIJ        REAL        1 TO ISIZE       STRAIN RATE MATRICIES   30100330   000000
00101   C                            1 TO 3             EIJ(I,J)=.5*(U(I),X(J)30100340   000000
00101   C                            1 TO 3                 +U(J),X(I))       30100350   000000
00101   C                                                                     30100360   000000
00101   C     MUTHET     REAL        1 TO ISIZE       SEE DESCRIPTION NOTE 3  30100370   000000
00101   C                                                                     302A0380   000000
00101   C     ARGUMENTS PASSED IN SUBROUTINE CALL ******                      302A0382   000000
00101   C                                                                     302A0384   000000
00101   C     ARGUMENT   TYPE        DIMENSION      DESCRIPTION               302A0386   000000
00101   C     ---------  ---------   ---------      ------------------        302A0388   000000
00101   C     IL         INTEGER     SCALAR         AEROELEMENT ID LOW LIMIT  302A0390   000000
00101   C                                                                     302A0392   000000
00101   C     IH         INTEGER     SCALAR         AEROELEMENT ID HIGH LIMIT 302A0394   000000
00101   C                                                                     302A0396   000000
00101   C                                                                     302A0398   000000
00101   C     DESCRIPTION ******                                              302A0400   000000
00101   C                                                                     302A0402   000000
00101   C     THIS SUBROUTINE PERFORMS THE FOLLOWING OPERATIONS FOR EACH      30100410   000000
00101   C     AEROELEMENT IN THE RANGE 'IL' TO 'IH'.                          302A0420   000000
00101   C                                                                     30100430   000000
00101   C     1  THE DENSITY, RHO, IS CHECKED FOR APPROXIMATE EQUALITY WITH ZERO30100570   000000
00101   C        IF SO, STATUS BIT 1 IS SET (ZERO DENSITY ERROR) AND          30100580   000000
00101   C        REMAINING CALCULATIONS ARE SKIPPED.  THE CALCULATIONS ARE ALSO 302A0590 000000
00101   C        SKIPPED IF THE 'INTERPOLATOR MATRIX NOT AVAILABLE' FLAG IS SET.302A0600 000000
00101   C                                                                     30100610   000000
00101   C     2  THE AEROELEMENT ACCELERATIONS, A, ARE CALCULATED.            30100620   000000
00101   C                                                                     30100630   000000
00101   C                                                                     30100640   000000
00101   C     REQUIRED SUBROUTINES ******                                     30130650   000000
00101   C                                                                     30130652   000000
00101   C     401        AVIRI                                                30130654   000000
00101   C                            404         U                            30130656   000000
00101   C     407        STA                                                  30130658   000000
00101   C                            408         P                            30130660   000000
00101   C     417        STSTAT      416         STAT                         30130662   000000
00101   C                            422         NEI                          30130664   000000
00101   C                            424         INTP                         30130666   000000
00101   C     427        STEIJ       426         EIJ                          30130668   000000
00101   C                            428         MU                           30130670   000000
00101   C     431        STMUTH      430         MUTHET                       30130672   000000
00101   C                                                                     30100680   000000
00101   C                                                                     30100690   000000
00101   C                                                                     30100700   000000
00103         INTEGER IL,IH,IS,ID,IE                                          302A0710   000000
00104         INTEGER STAT                                                               000000
00105         REAL MU,MUTHET                                                             000000
00106         REAL INTP              @ ACCESS TO INTERPOLATOR MATRIX          V01-03D    000000
00107         INTEGER LNEI(5)        @ LOCAL NEIGHBOR LIST                    V01-03D    000000
00110         INTEGER LSTAT          @ LOCAL STATUS                           V01-03D    000000
00111         REAL LP                @ LOCAL COPY -- PRESSURE                 V01-03D    000000
00112         REAL LRHO              @ LOCAL COPY -- DENSITY                  V01-03D    000000
00113         REAL LINTP(5,3)        @ LOCAL COPY -- INTERPOLATOR MATRIX      V01-03D    000000
00114         REAL LMU               @ LOCAL COPY -- VISCOUSITY               V01-03D    000000
```

Listing 6.—Continued.

```
      REAL LNMU(5)            @ LOCAL COPY -- NEIGHBOR VISCOUSITIES    V01-03D    000000
      REAL LEIJ(3,3)          @ LOCAL COPY -- STRAIN RATE MATRIX       V01-03D    000000
      REAL LMUTHE             @ LOCAL COPY -- MUTHET                   V01-03D    000000
      DATA ID/302/                                                     302A0712   000000
      DATA IE/1/                                                       302A0714   000000
      REAL DELT(5),DPDXJ(3),DTDXJ(3),DELTM(5,3,3),DMEDXI(3,3)          30100760   000000
      REAL DEJDXI(3)                                                   30100770   000000
      DEFINE DU(I,J)=U(I,J)                                            30100780   000000
      DEFINE DP(I)=P(I)                                                30100790   000000
      DEFINE DMUTHE(I)=MUTHET(I)                                       30100800   000000
      DEFINE DMU(I,J)=MU(I,J)                                          30100810   000000
      DEFINE DEIJ(I,J,K)=EIJ(I,J,K)                                    30100820   000000
      CALL CHKLH (IL,IH,IS,ID,IE)       @ GO CHECK AEROELEMENT RANGE   302A0990   000000
      IF (IS) 100,701,100               @ VALID RANGE ?                302A1010   000006
100   DO 221 I1=IL,IH,IS                @ YES, DO EACH                 V01-03D    000010
      LSTAT=STAT(I1)                    @ LOCALIZE AEROELEMENT STATUS  V01-03D    000020
      IF (AND(LSTAT,1)) 220,103,220     @ INTERPOLATOR AVAILABLE ?     V01-03D    000023
103   LRHO=RHO(I1)                      @ YES, LOCALIZE DENSITY        V01-03D    000026
      LSTAT=AND(LSTAT,COMPL(2))         @ RESET DENSITY ERROR FLAG     V01-03D    000032
      IF (LRHO-1.0E-25) 106,106,107     @ DENSITY TOO SMALL ?          V01-03D    000035
106   LSTAT=OR(LSTAT,2)                 @ YES, SET FLAG                V01-03D    000040
107   CALL STSTAT (I1,LSTAT)            @ REVISE STATUS IN ANY EVENT   V01-03D    000044
      IF (AND(LSTAT,2)) 220,109,220     @ DENSITY OK ?                 V01-03D    000047
109   LMU=MU(I1,1)                      @ YES, LOCALIZE VISCOUSITY     V01-03D    000052
      DO 112 I=1,5                      @ LOCALIZE NEIGHBOR LIST       V01-03D    000062
      LNEI(I)=NEI(I1,I)                 @                              V01-03D    000062
112   CONTINUE                          @                              V01-03D    000100
      DO 116 I=1,3                      @ LOCALIZE TRANSPOSE OF        V01-03D    000100
      DO 116 J=1,3                      @  STRAIN RATE MATRIX          V01-03D    000100
      LEIJ(J,I)=EIJ(I1,I,J)             @                              V01-03D    000100
116   CONTINUE                          @                              V01-03D    000121
      DO 120 I=1,3                      @ LOCALIZE TRANSPOSE OF        V01-03D    000121
      DO 120 J=1,5                      @  INTERPOLATOR MATRIX         V01-03D    000121
      LINTP(J,I)=INTP(I1,I,J)           @                              V01-03D    000121
120   CONTINUE                          @                              V01-03D    000136
      DO 124 I=1,5                      @ LOCALIZE NEIGHBOR'S          V01-03D    000136
      J=LNEI(I)                         @  VISCOUSITIES                V01-03D    000136
      LNMU(I)=MU(J,1)                   @                              V01-03D    000137
124   CONTINUE                          @                              V01-03D    000145
      LP=P(I1)                          @ LOCALIZE PRESSURE            V01-03D    000145
      LMUTHE=MUTHET(I1)                 @ LOCALIZE MUTHET              V01-03D    000151
      DO 201 I=1,5                      @ FORM PRESSURE DIFFERENCES    V01-03D    000160
201   DELT(I)=DP(LNEI(I))-LP            @                              V01-03D    000160
      DO 207 J=1,3                      @CLR DERIVATIVES OF PRESSURE   30101100   000174
      DPDXJ(J)=0.0                      @  AND CALCULATE NEW BY        30101110   000174
      DO 207 K=1,5                      @  INTERPOLATOR                30101120   000177
207   DPDXJ(J)=DPDXJ(J)+(LINTP(K,J)*DELT(K)) @                         V01-03D    000177
      DO 208 I=1,5                      @CALC NOTE 3 DIFFERENCES       30101140   000211
208   DELT(I)=DMUTHE(LNEI(I))-LMUTHE    @                              V01-03D    000211
      DO 209 J=1,3                      @CLR NOTE 3 DERIVATIVES AND    30101160   000225
      DTDXJ(J)=0.0                      @  CLACULATE NEW BY            30101170   000225
      DO 209 K=1,5                      @  INTERPOLATOR                30101180   000230
209   DTDXJ(J)=DTDXJ(J)+(LINTP(K,J)*DELT(K)) @                         V01-03D    000230
      DO 210 K=1,5                      @CALC DIFFERENCE MATRIX FOR    30101200   000251
      DO 210 I=1,3                      @  2*MU*EIJ(I,J),X(I)          30101210   000251
      DO 210 J=1,3                                                     30101220   000251
210   DELTM(K,I,J)=((LNMU(K)*DEIJ(LNEI(K),I,J))-(LMU*LEIJ(J,I)))*2.0 @ V01-03D    000251
      DO 211 I=1,3                      @CLR DERIVATIVES AND CALC NEW  30101250   000311
      DO 211 J=1,3                      @  FOR 2*MU*EIJ(I,J),X(I)      30101260   000311
      DMEDXI(I,J)=0.0                                                  30101270   000311
      DO 211 K=1,5                                                     30101280   000313
211   DMEDXI(I,J)=DMEDXI(I,J)+(LINTP(K,I)*DELTM(K,I,J)) @              V01-03D    000313
      DO 212 J=1,3                      @PERFORM IMPLIED TENSOR SUM    30101300   000342
      DEJDXI(J)=0.0                     @  PER NAVIER-STOKES EQUATION  30101310   000342
      DO 212 I=1,3                                                     30101320   000344
212   DEJDXI(J)=DEJDXI(J)+DMEDXI(I,J)                                  30101330   000344
      DO 213 J=1,3                      @ CALC NAVIER-STOKES EQUATION  30131340   000355
      RRR=(DTDXJ(J)+DEJDXI(J)-DPDXJ(J))/LRHO @                         V01-03D    000355
213   CALL STA (I1,J,RRR)               @                              30131355   000362
220   CALL CHKTIM (IL,IH,I1)            @ KEEP AN EYE ON THE TIME      V01-03D    000372
221   CONTINUE                          @                              V01-03D    000403
701   RETURN                                                           301A1370   000403
      END                                                              30101380   000450
END FOR
>
```

Listing 6.—Concluded.

```
 @FOR,MS  CASPER9.MOVELD
FOR   4R1   E -04/13/84-15:11:33 (12,)
>@EOF


   SUBROUTINE MOVEL      ENTRY POINT 001010


   STORAGE USED: CODE(1) 001544; DATA(0) 002706; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   NZNC   000001
    0004   IDUC   000001


   EXTERNAL REFERENCES (BLOCK, NAME)

    0005   FZ
    0006   STAT
    0007   ZBL
    0010   IPZBL
    0011   IPLZN
    0012   LZN
    0013   CHKLH
    0014   CHKTIM
    0015   STIAES
    0016   X
    0017   U
    0020   A
    0021   STS
    0022   TOGSW
    0023   GRDBD
    0024   SURFVE
    0025   SURFAC
    0026   STSTAT
    0027   ERROR2
    0030   TSTZN
    0031   TSTBDT
    0032   NWDU$
    0033   NIO2$
    0034   SQRT
    0035   NERR3$
    0036   NIO1$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   001333 1002L    0001   000163 101L     0001   001467 1025L    0001   001476 1028L    0001   001501 1050L
    0001   000412 106L     0001   000262 107L     0000   000243 1075F    0000   000251 1076F    0000   000261 1077F
    0000   000267 1078F    0000   000275 1079F    0000   000304 1080F    0001   001110 15L      0001   000007 175G
    0001   000007 200G     0001   000046 217G     0001   000714 219L     0001   000767 221L     0001   000071 227G
    0001   000153 250G     0001   000011 2755L    0001   000040 2796L    0001   001203 29L      0001   000264 313G
    0001   000274 322G     0001   000311 330G     0001   000324 337G     0001   000333 344G     0001   000442 3500L
    0001   000351 355G     0001   000472 3570L    0001   000503 3572L    0000   000134 3573F    0001   000514 3580L
    0001   000521 3600L    0001   000360 363G     0001   000566 3710L    0001   000371 373G     0001   000604 3760L
    0001   000615 3790L    0001   000617 3800L    0001   000633 3812L    0000   000156 3813F    0001   000644 3816L
    0001   000651 3840L    0001   000665 3852L    0000   000202 3853F    0001   000676 3856L    0001   000703 3880L
    0001   000400 400G     0001   000347 4002L    0001   000376 4010L    0001   000404 4012L    0001   001262 41L
    0001   000717 4100L    0001   000724 4101L    0000   000225 4102F    0001   000433 416G     0001   001264 42L
    0001   000216 4303L    0000   000114 4304F    0001   000227 4307L    0001   000234 4325L    0001   001272 45L
    0001   000524 454G     0001   000534 463G     0001   000575 507G     0001   000610 520G     0001   000737 572G
    0001   000735 6000L    0001   001041 610G     0001   001227 663G     0001   001372 735G     0001   001463 760G
    0020 R 000000 A        0000 R 000003 AL       0000 R 000314 ARB      0000 R 000073 B        0000 R 000074 C
    0005 I 000000 FZ       0000 R 000026 GB       0000 I 000054 I        0000 I 000063 IBUMP    0000 I 000111 ID
    0000 I 000070 IDB      0000 I 000102 IDMIN    0004 I 000000 IDU      0000 I 000104 IDUM     0000 I 000101 IE3F
    0000 I 000064 IFZ      0000   002610 INJP$    0000   002627 INJP$    0000   002623 INJP$    0000   002615 INJP$
    0011 I 000000 IPLZN    0010 I 000000 IPZBL    0000 I 000314 IRB      0000 I 000002 IRBPP    0000 I 000001 IRBPT
    0000 I 000000 IS       0000 I 000113 IT       0000 I 000062 IZVCF    0000 I 000061 I1       0000 I 000055 J
    0000 I 000072 K        0012 I 000000 LZN      0000 I 000103 N        0000 I 000105 N        0000 I 000106 NA
    0000 I 000107 NB       0000 I 000110 NC       0000 I 000075 NNN      0003 I 000000 NZN      0006 I 000000 STAT
    0000 R 000112 SV       0000 R 000071 SVMIN    0000 R 000065 T        0000 R 000077 TA       0000 R 000060 TE
```

Listing 7.—Subroutine MOVEL.

```
0000 R 000056 TINC     0000 R 000057 TINCSQ   0000 R 000067 TLEFT    0000 R 000076 TN       0000 R 000100 TQ
0000 R 000066 TS       0017 R 000000 U        0000 R 000045 UA       0000 R 000031 UL       0000 R 000041 UN
0000 R 000035 UQ       0000 R 000051 VBO      0016 R 000000 X        0000 R 000022 XA       0000 R 000006 XL
0000 R 000016 XN       0000 R 000012 XQ       0007 I 000000 ZBL
```

```
00101          SUBROUTINE MOVEL (IL,IH,CURTIM,GDAST,NINC)                              901A0010   000000
00101     C                                                                            90100030   000000
00101     C                                                                            90100040   000000
00101     C    MOVEL         ****** A SUBROUTINE FOR CASPER ******                     90100050   000000
00101     C                  AUTHOR        WILLIAM HENRY JONES                         90100060   000000
00101     C                  V02-00        14 APR 77                                   90100070   000000
00101     C                  V02-01        22 JUN 77                                   90110071   000000
00101     C                  V02-02        26 JUL 77                                   90120072   000000
00101     C                  V02-03        22 SEP 77                                   90130073   000000
00101     C                  V02-04        22 SEP 77                                   90140074   000000
00101     C                  V02-05        26 SEP 77                                   90150075   000000
00101     C                  V02-06        01 JUN 78                                   90160076   000000
00101     C                  V02-07        16 JUN 78                                   90170077   000000
00101     C                  V02-08        29 AUG 78                                   90180078   000000
00101     C                  V02-08A       13 FEB 79                                   901A0079   000000
00101     C                  V02-08B       09 MAR 79                                   901B0080   000000
00101     C                  V02-08C       13 FEB 80     INPUT/OUTPUT SEGREGATION                 000000
00101     C                  V02-08D       15 SEP 80     FUNCTION TYPE STATEMENTS                 000000
00101     C                  V02-08E       28 SEP 81     MOVING BOUNDARIES             V02-08E    000000
00101     C                  V02-08F       06 JAN 83     BAD POSITION INTEGRATION      V02-08F    000000
00101     C                  V02-08G       16 MAY 83     TYPE OF ZBL WRONG             V02-08G    000000
00101     C                  V02-08H       27 JUN 83     DEBUG MESSAGES AND HISTORY    V02-08H    000000
00101     C                  V02-08I       27 JUL 83     TYPOS                         V02-08I    000000
00101     C                  V02-08J       05 AUG 83     ZERO VELOCITY COLLISIONS      V02-08J    000000
00101     C                  V02-08K       24 OCT 83     FLOW ZONE NOT FOUND TRAPS     V02-08K    000000
00101     C                  V02-08L       25 JAN 84     AVOID ENDLESS BUMPS           V02-08L    000000
00101     C                                                                            90100090   000000
00101     C    ARGUMENTS IN CASPER 'CACHE' MEMORY ******                               90180094   000000
00101     C                                                                            90180096   000000
00101     C    ARGUMENT      TYPE          DIMENSION     DESCRIPTION                   90100100   000000
00101     C    ---------     ---------     ---------     -------------------           90180110   000000
00101     C    X             REAL          1 TO ISIZE    AEROELEMENT POSITION          90100120   000000
00101     C                                1 TO 3          COORDINATES                 90100130   000000
00101     C                                                                            90100140   000000
00101     C    U             REAL          1 TO ISIZE    AEROELEMENT VELOCITIES        90100150   000000
00101     C                                1 TO 3                                      90100160   000000
00101     C                                                                            90100170   000000
00101     C    A             REAL          1 TO ISIZE    AEROELEMENT ACCELERATIONS     90100180   000000
00101     C                                1 TO 3                                      90100190   000000
00101     C                                                                            90100200   000000
00101     C    FZ            INTEGER       1 TO ISIZE    AEROELEMENT FLOW ZONE         90100210   000000
00101     C                                                NUMBERS (BY AEROELEMENT)    90100220   000000
00101     C                                                                            90100230   000000
00101     C    STAT          INTEGER       1 TO ISIZE    AEROELEMENT STATUS LIST       90100240   000000
00101     C                                                                            90100360   000000
00101     C    ZBL           INTEGER       1 TO ZBLSZ    BOUNDARY LIST BY FLOW ZONES90100370      000000
00101     C                                                                            90100380   000000
00101     C    IPZBL         INTEGER       1 TO NZN      ZBL CONTROL PARAMETERS LIST90100390      000000
00101     C                                1 TO 2          (X,1) = STARTING POINT      90100400   000000
00101     C                                                (X,2) = STRING LENGTH       90100410   000000
00101     C                                                                            90100420   000000
00101     C                                                                            90100560   000000
00101     C    ARGUMENTS PASSED IN SUBROUTINE CALL ******                              90180562   000000
00101     C                                                                            90180564   000000
00101     C    ARGUMENT      TYPE          DIMENSION     DESCRIPTION                   90180566   000000
00101     C    ---------     ---------     ---------     ------------------            90180568   000000
00101     C    IL            INTEGER       SCALAR        AEROELEMENT ID LOW LIMIT      901A0569   000000
00101     C                                                                            901A0570   000000
00101     C    IH            INTEGER       SCALAR        AEROELEMENT ID HIGH LIMIT     901A0571   000000
00101     C                                                                            901A0572   000000
00101     C    CURTIM        REAL          SCALAR        CURRENT OPENING TIME          901A0573   000000
00101     C                                                                            901A0574   000000
00101     C    GDAST         REAL          SCALAR        BASIC TIME INCREMENT          901A0575   000000
00101     C                                                                            90180576   000000
```

Listing 7.—Continued.

```
00101   C   NINC         INTEGER      SCALAR          NUMBER OF TIME SUB-        90180578   000000
00101   C                                               INCREMENTS               90180580   000000
00101   C                                                                        90180582   000000
00101   C                                                                        90180584   000000
00101   C   RESULT LOCATIONS ******                                                         000000
00101   C                                                                                   000000
00101   C   LOCATION     CONTENTS                                                           000000
00101   C   ---------    -----------------------                                            000000
00101   C   AESCRA       FINAL FLOW ZONE ID OF AEROELEMENT                                  000000
00101   C                                                                                   000000
00101   C   S(1)         X(1)                                                               000000
00101   C   S(2)         X(2)                                                               000000
00101   C   S(3)         X(3)                                                               000000
00101   C   S(4)         U(1)                                                               000000
00101   C   S(5)         U(2)                                                               000000
00101   C   S(6)         U(3)                                                               000000
00101   C                                                                                   000000
00101   C                                                                                   000000
00101   C   DESCRIPTION ******                                                   90180586   000000
00101   C                                                                        90180588   000000
00101   C   MOVEL IS A SUBROUTINE WHICH, GIVEN THE POSITION, VELOCITY, AND       90100590   000000
00101   C   ACCELERATION OF INDIVIDUAL AEROELEMENTS AS WELL AS A DEFINITION      90100600   000000
00101   C   OF THE BOUNDARIES AND FLOW ZONES OF THE AIRFLOW VOLUME, WILL         90100610   000000
00101   C   REPOSITION THOSE AEROELEMENTS THAT ARE NOT RESTRICTED TO OTHER       90100620   000000
00101   C   PRESET LAWS OF MOTION (E.G. - BOUNDARY ELEMENTS FIXED IN SPACE)      90100630   000000
00101   C   ACCORDING TO THE CLASSIC INTEGRATION OF CONSTANTLY ACCELERATING      90100640   000000
00101   C   MOTION.                                                              90100650   000000
00101   C                                                                        90100660   000000
00101   C   DURING SUCH RELOCATION EACH APPROPRIATE BOUNDARY IS CHECKED FOR      90100670   000000
00101   C   POTENTIAL VIOLATIONS BY THE AEROELEMENT.  IF SUCH A VIOLATION IS     90100680   000000
00101   C   DETECTED THE POINT OF VIOLATION IS FOUND AND THE AEROELEMENT IS      90100690   000000
00101   C   ELASTICALLY BOUNCED OFF THE BOUNDARY AT THAT LOCATION.  TO ENHANCE90100700   000000
00101   C   BOUNDARY VIOLATION DETECTION A SUB-INCREMENTAL TIME STEP IS          90100710   000000
00101   C   SPECIFYABLE BY THE INTEGER ARGUMENT NINC.  THIS WILL DIVIDE THE      90100720   000000
00101   C   PARABOLIC MOTION FROM X @ T TO X @ T+GDAST INTO NINC EQUAL STEPS     90100730   000000
00101   C   AND CHECK FOR BOUNDARY VIOLATIONS AT EACH OF THE INTERMEDIATE        90100740   000000
00101   C   POSITIONS, THUS LOWERING THE PROBABILITY OF AEROELEMENTS 'PASSING 90100750   000000
00101   C   THROUGH' THIN BOUNDARIES SUCH AS LEADING AND TRAILING EDGES OF       90100760   000000
00101   C   AIRFOILS.                                                            90100770   000000
00101   C                                                                        V02-08E    000000
00101   C   THE BOUNDARY BOUNCING PROCESS IS A SIMPLE REFLECTION ALGORITHM,      V02-08E    000000
00101   C   I.E., ANGLE OF INCIDENCE EQUALS ANGLE OF REFLECTION.  TO DO THIS, V02-08E    000000
00101   C   THE VELOCITY VECTOR FOR THE AEROELEMENT IS ADJUSTED AT THE TIME      V02-08E    000000
00101   C   OF BOUNCE TO GIVE THE APPROPRIATE INITIAL DIRECTION.  THE            V02-08E    000000
00101   C   ACCELERATION OF THE AEROELEMENT IS NOT ADJUSTED.  TO ACCOUNT         V02-08E    000000
00101   C   FOR SITUATIONS WHERE THE AEROELEMENT IS NOT MOVING AND IS HIT        V02-08E    000000
00101   C   BY A MOVING BOUNDARY, THE AEROELEMENT VELOCITY IS FIRST CONVERTED V02-08E    000000
00101   C   TO A VELOCITY RELATIVE TO THE BOUNDARY, ADJUSTED FOR THE BOUNCE,     V02-08E    000000
00101   C   AND THEN CONVERTED BACK TO VELOCITY RELATIVE TO THE STATIONARY       V02-08E    000000
00101   C   REFERENCE FRAME.                                                     V02-08E    000000
00101   C                                                                        90100780   000000
00101   C   SELECTED VARIABLES IN THE ARGUMENT LIST, NOTABLY LZN, ZBL, BDZN,     90100830   000000
00101   C   AND NEIZN, ARE PASSED WITH CONTROL PARAMETER LISTS IN DYNAMICALLY 90100840   000000
00101   C   VARIABLE ARRAY FORM.  AS NOTED IN THE ARGUMENT DESCRIPTIONS, THE     90100850   000000
00101   C   CONTROL PARAMETERS CONSIST OF A STARTING POINT LIST AND A STRING     90100860   000000
00101   C   LENGTH LIST.  THESE DYNAMICALLY VARIABLE                             90180870   000000
00101   C   ARRAYS ARE ARRANGED SUCH THAT THE SUB-ARRAY RUNS FROM 1 TO THE       90100890   000000
00101   C   STRING LENGTH AND THE FIRST ELEMENT IS AT THE STARTING POINT PLUS 90100900   000000
00101   C   1.  THUS, FOR LZN, THE J TH ELEMENT OF THE I TH FLOW ZONE LIST       90100910   000000
00101   C   WOULD BE LZN(IPLZN(I,1)+J) AND THE LENGTH OF THE I TH FLOW ZONE      90100920   000000
00101   C   LIST WOULD BE IPLZN(I,2).                                            90100930   000000
00101   C                                                                        V02-08H    000000
00101   C                                                                        V02-08H    000000
00101   C   POSITION HISTORY RING BUFFER                                         V02-08H    000000
00101   C                                                                        V02-08H    000000
00101   C   WITH VERSION V02-08H, A POSITION HISTORY RING BUFFER IS PROVIDED. V02-08H    000000
00101   C   EACH TIME A NEW AEROELEMENT POSITION IS ESTABLISHED, AN ENTRY IS     V02-08H    000000
00101   C   MADE IN THE NEXT (REPLACING THE OLDEST) RING BUFFER SLOT.  IF AN     V02-08H    000000
00101   C   ERROR OCCURS AND DEBUG MESSAGES ARE ENABLED, THE CONTENTS OF THE     V02-08H    000000
00101   C   RING BUFFER ARE DUMPED WITH THE ERROR MESSAGE.                       V02-08H    000000
00101   C                                                                        V02-08H    000000
00101   C   THE FOLLOWING INFORMATION IS RECORDED IN EACH SLOT OF THE            V02-08H    000000
```

Listing 7.—Continued.

```
00101   C   POSITION-HISTORY RING BUFFER:                                          V02-08H    000000
00101   C                                                                          V02-08H    000000
00101   C   INDEX        CONTENTS                                                  V02-08H    000000
00101   C   ---------    --------------------------------------------------------V02-08H    000000
00101   C   1            INFORMATION AND STATUS                                    V02-08H    000000
00101   C   2            AEROELEMENT ID                                            V02-08H    000000
00101   C   3            AEROELEMENT X(1)                                          V02-08H    000000
00101   C   4            AEROELEMENT X(2)                                          V02-08H    000000
00101   C   5            AEROELEMENT X(3)                                          V02-08H    000000
00101   C   6            AEROELEMENT X(4)                                          V02-08H    000000
00101   C   7            FLOW ZONE ID                                              V02-08H    000000
00101   C   8            SURFACE ID                                                V02-08H    000000
00101   C   9            SURFACE VALUE                                             V02-08H    000000
00101   C   10           SURFACE GRADIENT (1)                                      V02-08H    000000
00101   C   11           SURFACE GRADIENT (2)                                      V02-08H    000000
00101   C   12           SURFACE GRADIENT (3)                                      V02-08H    000000
00101   C                                                                          V02-08J    000000
00101   C                                                                          V02-08J    000000
00101   C                                                                          V02-08J    000000
00101   C                                                                          V02-08J    000000
00101   C   ZERO VELOCITY IMPACTS                                                  V02-08J    000000
00101   C                                                                          V02-08J    000000
00101   C   ANOTHER ANOMALOUS SITUATION IS TREATED BY VERSION V02-08J.  IT         V02-08J    000000
00101   C   IS POSSIBLE THAT AN AEROELEMENT MAY LIE DIFFERENTIALLY CLOSE TO        V02-08J    000000
00101   C   A BOUNDARY AND HAVE A ZERO VELOCITY NORMAL TO THE BOUNDARY WITH A      V02-08J    000000
00101   C   FINITE NORMAL ACCELERATION INTO THE BOUNDARY.  IN THIS CASE, ANY       V02-08J    000000
00101   C   INCREMENTAL STEP FORWARD IN TIME WILL CARRY THE AEROELEMENT INTO       V02-08J    000000
00101   C   VIOLATION OF THE BOUNDARY.  THE BOUNDARY BOUNCE OPERATION WILL         V02-08J    000000
00101   C   THEN RETURN THE AEROELEMENT TO ITS ORIGINAL POSITION IN SPACE          V02-08J    000000
00101   C   AND TIME AND MODIFY ITS VELOCITY SO THAT THERE IS (STILL) NO           V02-08J    000000
00101   C   VELOCITY COMPONENT NORMAL TO THE BOUNDARY (THE NEGATIVE OF ZERO        V02-08J    000000
00101   C   BEING ZERO).  IN THIS MANNER AN INFINITE LOOP WILL FORM.               V02-08J    000000
00101   C                                                                          V02-08J    000000
00101   C   THIS SITUATION IS DETECTED BY THE SIMPLE EXPEDIENT OF OBSERVING        V02-08J    000000
00101   C   WHETHER OR NOT THE BOUNCE LOCATOR REPLACED THE ORIGINAL NON-           V02-08J    000000
00101   C   VIOLATING POINT WITH A NEW NON-VIOLATING POINT.  IF SUCH A             V02-08J    000000
00101   C   REPLACEMENT WAS MADE, THEN THE ORIGINAL POINT WAS NOT DIFFER-          V02-08J    000000
00101   C   ENTIALLY CLOSE TO THE BOUNDARY SINCE (HA! HA!) A POINT BETWEEN         V02-08J    000000
00101   C   IT AND THE BOUNDARY EXISTED.                                           V02-08J    000000
00101   C                                                                          V02-08J    000000
00101   C   IF THIS CONDITION IS IDENTIFIED AS DESCRIBED IN THE PRECEEDING         V02-08J    000000
00101   C   PARAGRAPH, THE ACCELERATION OF THE AEROELEMENT RELATIVE TO THE         V02-08J    000000
00101   C   BOUNDARY IS EXAMINED.  IF AN ACCELERATION COMPONENT INTO THE           V02-08J    000000
00101   C   BOUNDARY EXISTS, IT IS SET TO ZERO IN THE FRAME OF REFERENCE OF        V02-08J    000000
00101   C   THE BOUNDARY.                                                          V02-08J    000000
00101   C                                                                          V02-08H    000000
00101   C                                                                          90100940   000000
00101   C                                                                          90100950   000000
00101   C                                                                          90100960   000000
00101   C                                                                          90100970   000000
00101   C                                                                          90100980   000000
00101   C                                                                          90100990   000000
00101   C                                                                          90101000   000000
00101   C                                                                          90102500   000000
00101   C   REQUIRED SUBROUTINES ******                                            90182510   000000
00101   C                                                                          90182512   000000
00101   C    401          AVIRI                                                    90182514   000000
00101   C                              402          X                              90182516   000000
00101   C                              404          U                              90182518   000000
00101   C                              406          A                              90182520   000000
00101   C   417          STSTAT        416          STAT                           90182522   000000
00101   C   421          STFZ          420          FZ                             90182524   000000
00101   C                              470          FVIRI                          90182534   000000
00101   C   471          ZBL                                                       90182536   000000
00101   C   475          HVIRI         476          IPZBL                          90182538   000000
00101   C   911          MRRM          912          TSTZN                          90182540   000000
00101   C                                                                          90182542   000000
00101   C                                                                          90182544   000000
00101   C   ERRORS REPORTED ******                                                 90182546   000000
00101   C                                                                          90182548   000000
00101   C   1             NEITHER RESULT OF BOUNDARY SURFACE FINDER WAS            90182550   000000
00101   C                  SAFE (901X3870).                                        90182552   000000
```

Listing 7.—Continued.

```
00101    C    2        BOUNDARY SURFACE FINDER SETUP PUSHED 'Q' BACK              90182554   000000
00101    C              BEYOND ZERO TIME WITHOUT FINDING A SAFE POSITION          90182556   000000
00101    C             (901X3348).                                                90182558   000000
00101    C    3        AN AEROELEMENT FLOW PATH WAS FOUND THAT LEADS TO           901A2560   000000
00101    C              BOUNDARY VIOLATION WITHOUT CROSSING A LEGITIMATE          901A2565   000000
00101    C              ACTIVE BOUNDARY.                                          901A2570   000000
00101    C    4        THE BOUNDARY INTERCEPT LOCATOR FAILED TO LOCK ON           901A2575   000000
00101    C              TO AN EXISTING ACTIVE BOUNDARY.                           901A2580   000000
00101    C                                                                        901A2585   000000
00101    C-                                                                       901A2590   000000
00103         PARAMETER OURID=901      @                                          V02-08H    000000
00104         PARAMETER IBUMPC=100     @ INITIAL BUMP COUNT                       V02-08L    000000
00105         PARAMETER IRBL=100       @ RING BUFFER LENGTH                       V02-08H    000000
00106         PARAMETER IRBIS=1        @ INFORMATION AND STATUS                   V02-08H    000000
00107         PARAMETER IRBAI=2        @ AEROELEMENT ID                           V02-08H    000000
00110         PARAMETER IRBX1=3        @ AEROELEMENT X(1)                         V02-08H    000000
00111         PARAMETER IRBX2=4        @ AEROELEMENT X(2)                         V02-08H    000000
00112         PARAMETER IRBX3=5        @ AEROELEMENT X(3)                         V02-08H    000000
00113         PARAMETER IRBX4=6        @ AEROELEMENT X(4)                         V02-08H    000000
00114         PARAMETER IRBFI=7        @ FLOW ZONE ID                             V02-08H    000000
00115         PARAMETER IRBSI=8        @ SURFACE ID                               V02-08H    000000
00116         PARAMETER IRBSV=9        @ SURFACE VALUE                            V02-08H    000000
00117         PARAMETER IRBG1=10       @ SURFACE GRADIENT (1)                     V02-08H    000000
00120         PARAMETER IRBG2=11       @ SURFACE GRADIENT (2)                     V02-08H    000000
00121         PARAMETER IRBG3=12       @ SURFACE GRADIENT (3)                     V02-08H    000000
00122         PARAMETER IRBM=12        @ SLOT LENGTH                              V02-08H    000000
00123         INCLUDE PGSDEF           @ PAX GLOBAL STATUS DEFINITIONS            V02-08H    000000
00123    C+                                                                       V02-08H    000000
00123    C    POSITION RING BUFFER STATUS BITS                                    V02-08H    000000
00123    C-                                                                       V02-08H    000000
00126         PARAMETER IRBISA=1       @ AEROELEMENT ID/POSITION VALID            V02-08H    000000
00127         PARAMETER IRBISF=2       @ FLOW ZONE ID VALID                       V02-08H    000000
00130         PARAMETER IRBISS=4       @ SURFACE ID VALID                         V02-08H    000000
00131         PARAMETER IRBISV=8       @ SURFACE VALUE VALID                      V02-08H    000000
00132         PARAMETER IRBISG=16      @ GRADIENT VALID                           V02-08H    000000
00133         PARAMETER LUOUP=6        @ LOGICAL UNIT FOR OUTPUT                  V02-08H    000000
00134         INTEGER IL,IH,IS                                                    901A2595   000000
00135         INTEGER FZ,STAT,EIPLZN,ST,EMOD,STP1,ELZN,DLZN                                  000000
00136         COMMON /NZNC/NZN                                                    90102620   000000
00137         COMMON /IDUC/IDU                                                    90182630   000000
00140         REAL CURTIM,GDAST                                                   90182670   000000
00141         INTEGER NINC                                                        90182680   000000
00142         INTEGER ZBL                                                         V02-08G    000000
00143         INTEGER IRB(IRBM,IRBL)  @ POSITION RING BUFFER                      V02-08H    000000
00144         INTEGER IRBPT           @ POSITION RING BUFFER POINTER              V02-08H    000000
00145         INTEGER IRBPP           @                                           V02-08H    000000
00146         REAL AL(3),XL(4),XQ(4),XN(4),XA(4),GB(3)                            V02-08F    000000
00147         REAL UL(4),UQ(4),UN(4),UA(4)                                        V02-08F    000000
00150         REAL VBO(3)                                                         V02-08E    000000
00151         REAL ARB(IRBM,IRBL)     @ POSITION RING BUFFER                      V02-08H    000000
00152         EQUIVALENCE (IRB(1,1),ARB(1,1))        @                            V02-08H    000000
00153         DATA IRBPT/0/                          @                            V02-08H    000000
00155         DATA IRBPP/0/                          @                            V02-08H    000000
00157         DEFINE EZBL(I)=ZBL(I)                                               90112715   000000
00160         DEFINE DZBL(I,J)=EZBL(IPZBL(I,1)+J)                                 90112720   000000
00161         DEFINE EIPLZN(I,J)=IPLZN(I,J)                                       90112725   000000
00162         DEFINE ST(I)=EIPLZN(FZ(I),1)                                        90112730   000000
00163         DEFINE NU(I)=EIPLZN(FZ(I),2)                                        90112740   000000
00164         DEFINE EMOD(I,J)=MOD(I,J)                                           90112745   000000
00165         DEFINE STP1(I)=EIPLZN(EMOD(FZ(I),NZN)+1,1)                          90112750   000000
00166         DEFINE ELZN(I)=LZN(I)                                               90112755   000000
00167         DEFINE DLZN(I)=ELZN(ST(I)+NU(I))                                    90112760   000000
00170         IF (IRBPT) 2751,2751,2755              @ SHALL WE INIT RING BUFFER ? V02-08H   000000
00173    2751 IRBPT=1                                @ YES, RESET POINTER         V02-08H    000002
00174         DO 2754 I=1,IRBL                       @ ZAP BUFFER                 V02-08H    000007
00177         DO 2754 J=1,IRBM                       @                            V02-08H    000007
00202    2754 IRB(J,I)=0                             @                            V02-08H    000007
00205    2755 CONTINUE                               @                            V02-08H    000011
00206         TINC=GDAST/NINC                        @CALC SUB-INCREMENT          90102770   000011
00207         TINCSQ=0.5*TINC*TINC                   @CALC 1/2 SQUARE SUB-INCREMENT90112780  000015
00210         TE=0.001*TINC                          @TOLERANCE OF BOUNCES IN TIME 90132785  000020
00211         CALL CHKLH (IL,IH,IS,901,5)            @ GO CHECK AEROELEMENT RANGE 901A2790   000023
```

Listing 7.—Continued.

```
00212              IF (IS) 2796,2794,2796               @ VALID RANGE ?                901A2792    000032
00215         2794 RETURN                               @ NO                           901A2794    000034
00216         2796 DO 221 I1=IL,IH,IS                   @ IN RANGE AEROELEMENT LOOP    901A2796    000040
00221              CALL CHKTIM (IL,IH,I1)               @ KEEP AN EYE ON THE TIME                  000046
00222              IZVCF=0                              @ SHOW NO ZERO VELOCITY PROB   V02-08J     000053
00223              IBUMP=IBUMPC                         @ INIT BUMP COUNTER            V02-08L     000054
00224              IFZ=FZ(I1)                           @ LOCALIZE                                 000056
00225              CALL STIAES (I1,IFZ)                 @ OUTPUT IN CASE OF A SKIP                 000062
00226              DO 201 I=1,3                         @ LOCALIZE                                 000071
00231              XL(I)=X(I1,I)                        @  ORIGINAL POSITION                       000071
00232              UL(I)=U(I1,I)                        @  ORIGINAL VELOCITY                       000076
00233              AL(I)=A(I1,I)                        @  ACCELERATION                            000103
00234              CALL STS (I1,I,XL(I))                @ OUTPUT POSITION AND VELOCITY             000110
00235              CALL STS (I1,I+3,UL(I))              @  IN CASE OF A SKIP                       000117
00236      201     CONTINUE                             @                                          000133
00240              IF (AND(STAT(I1),2**9)) 221,2820,221 @ SKIP IF SPECIAL                          000133
00243      2820    IF (IFZ-IDU) 2830,221,2830           @ SKIP IF IN THE DOG HOUSE                 000143
00246      2830    XL(4)=CURTIM                         @                                          000146
00247              DO 219 I=1,NINC                      @START TIME SUB-INCREMENT LOOP90102850     000153
00252              T=TINC                               @SET TIME SPAN                 90102860    000153
00253              TS=TINCSQ                            @SET 1/2 SQUARE TIME SPAN      90102870    000155
00254              TLEFT=0.0                            @SET TIME REMAINING SUB-INC    90102880    000157
00255              IDB=-1                               @SET BOUNDARY NO-VIO FLAG      90102890    000160
00256      101     CALL SPC (XL,UL,T)                   @ POSITION AT END OF SPAN      V02-08F     000163
00257              CALL YNM (XL)                        @ GO CHECK ALL BOUNDARIES      901A2930    000167
00260              IF (IFZ-IDU) 103,4100,103            @ LOST FLOW ZONE TRACK ?       V02-08K     000172
00263      103     IF (SVMIN) 106,106,104               @ NO, 106 ON VIOLATION         V02-08K     000175
00266       104    IF (IDB) 219,105,105                 @CHK FOR PENDING BOUNCE        90103070    000200
00271      105     IBUMP=IBUMP-1                        @ ONE LESS BUMP ALLOWED        V02-08L     000203
00272              IF (IBUMP) 4302,4325,4325            @ BUMPED TOO OFTEN ?           V02-08L     000206
00275      4302    CALL TOGSW (PGSBUG,$4303,$4307)      @ YES, SHALL WE PRINT MESG ?   V02-08L     000210
00276      4303    WRITE (LUOUP,4304,END=4307,ERR=4307) @ YES, DO SO NOW               V02-08L     000216
00300      4304    FORMAT (58HOCASPER9.MOVELD (BUG) -- AEROELEMENT BUMPED TOO MANY TIV02-08L   000224
00300             1MES,/,27H  HISTORY BUFFER FOLLOWS...) @                             V02-08L     000224
00301              CALL RBDMP                           @ DUMP THE HISTORY BUFFER      V02-08L     000224
00302      4307    CALL STIAES (I1,IDU)                 @ DISCARD THIS TURKEY          V02-08L     000227
00303              GO TO 221                            @ DON'T WORK ANY MORE ON HIM   V02-08L     000232
00304      4325    CALL GRDBD (XL,IDB,GB)               @ BOUNCE -- GET GRADIENT       V02-08L     000234
00305              ARB(IRBG1,IRBPP)=GB(1)               @ RECORD GRADIENT RESULT       V02-08H     000242
00306              ARB(IRBG2,IRBPP)=GB(2)               @                              V02-08H     000244
00307              ARB(IRBG3,IRBPP)=GB(3)               @                              V02-08H     000246
00310              IRB(IRBIS,IRBPP)=OR(IRB(IRBIS,IRBPP),IRBISG) @                      V02-08H     000250
00311              CALL SURFVE (XL,IDB,VBO,$107)        @ GET SURFACE VELOCITY         V02-08E     000253
00312      107     DO 108 K=1,3                         @ GET VELOCITY OF AEROELEMENT  V01-08E     000264
00315      108     UL(K)=UL(K)-VBO(K)                   @  RELATIVE TO SURFACE         V02-08E     000264
00317              B=0.0                                @CLR ACCUMULATOR               90103090    000267
00320              C=0.0                                @CLR ACCUMULATOR               90103100    000270
00321              DO 204 K=1,3                         @ACCUMULATE LENGTH SQUARED     90103110    000274
00324       204    B=B+GB(K)**2                         @  OF GRADIENT VECTOR          90103120    000274
00326              B=1.0/SQRT(B)                        @FAST DIVIDE LENGTH OF GRAD    90133130    000300
00327              DO 205 K=1,3                         @LOOP TO                       90103140    000311
00332              GB(K)=B*GB(K)                        @  NORMALIZE GRADIENT          90103150    000311
00333       205    C=C+GB(K)*UL(K)                      @  ACCUMULATE DOT PROD VELOCTY90103160     000313
00335              C=2.0*C                              @ADJUST CONSTANT FOR BOUNCE    90103170    000317
00336              DO 206 K=1,3                         @BOUNCE! VELOCITY ANGLE INCID.90103180     000324
00341       206    UL(K)=UL(K)-C*GB(K)                  @  EQUALS ANGLE OF REFLECTION  90103190    000324
00343              DO 208 K=1,3                         @ AEROELEMENT VELOCITY RELA-   V02-08E     000333
00346      208     UL(K)=UL(K)+VBO(K)                   @  TIVE TO STATIONARY FRAME    V02-08E     000333
00350              IF (IZVCF) 4001,4012,4001            @ GOT A ZERO VELOCITY PROB ?   V02-08J     000336
00353      4001    CALL SURFAC (XL,IDB,VBO,$4002)       @ YES, GET SURFACE'S ACCEL     V02-08J     000340
00354      4002    DO 4003 K=1,3                        @ GET ACCELERATION OF ELEMENT  V02-08J     000351
00357      4003    AL(K)=AL(K)-VBO(K)                   @  RELATIVE TO SURFACE         V02-08J     000351
00361              C=0.0                                @ CALC COMPONENT OF ACCEL      V02-08J     000354
00362              DO 4006 K=1,3                        @  NORMAL TO THE BOUNDARY      V02-08J     000360
00365      4006    C=C+(GB(K)*AL(K))                    @  (GRAD HAS BEEN NORMALIZED)  V02-08J     000360
00367              IF (C) 4008,4010,4010                @ ANY ACCEL INTO BOUNDARY ?    V02-08J     000364
00372      4008    DO 4009 K=1,3                        @ YES, REMOVE IT               V02-08J     000371
00375      4009    AL(K)=AL(K)-(C*GB(K))                @                              V02-08J     000371
00377      4010    DO 4011 K=1,3                        @ ADJUST ACCEL BACK TO         V02-08J     000400
00402      4011    AL(K)=AL(K)+VBO(K)                   @  STATIONARY FRAME            V02-08J     000400
00404      4012    CONTINUE                             @                              V02-08J     000404
00405              T=TLEFT                              @SET TIME SPAN FOR REST OF     90103200    000404
```

Listing 7.—Continued.

ORIGINAL PAGE IS OF POOR QUALITY

```
00406            TLEFT=0.0                                 @CLR REMAINING TIME           90103220     000405
00407            IDB=-1                                    @WIPE OUT PENDING BOUNCE      90103230     000406
00410            GO TO 101                                 @JMP BACK - FINISH SUB-INCREMT90103240     000410
00410      C+                                                                            901A3250     000410
00410      C     THE FOLLOWING SECTION IS ENTERED WHEN A BOUNDARY IS VIOLATED.           901A3260     000410
00410      C     IT BACKS THE PARTICLE UP ALONG ITS PATH TO LOCATE THE POINT             901A3270     000410
00410      C     AT WHICH IT FIRST PENETRATES A BOUNDARY.  WHEN THIS POINT IS            901A3280     000410
00410      C     LOCATED TO WITHIN TOLERANCE 'TE' THE ID OF THE BOUNDARY                 901A3290     000410
00410      C     ABOUT TO BE VIOLATED IS LOADED INTO 'IDB' AND THE TIME IS               901A3300     000410
00410      C     SUBDIVIDED TO CAUSE A STEP JUST TO THE BOUNDARY FOLLOWED BY             901A3310     000410
00410      C     A BOUNCE AND A STEP TO THE END OF THE TIME SUB-INCREMENT.               901A3320     000410
00410      C                                                                             901A3330     000410
00410      C     IN LOCATING THE NEAR-VIOLATION POINT ALL BOUNDARIES IN THE              901A3340     000410
00410      C     MANDATORY FLOW ZONE AND IN THE AEROELEMENT'S FLOW ZONE OF               901A3350     000410
00410      C     RESIDENCE ARE CHECKED TO PRODUCE A VIOLATION/NO-VIOLATION               901A3360     000410
00410      C     DECISION.  THIS DECISION SHOULD BE BASED ON AT LEAST ONE                901A3370     000410
00410      C     BOUNDARY EVALUATION THAT DID NOT TRUNCATE.  IF THIS IS NOT              901A3380     000410
00410      C     THE CASE ERROR #3 IS REPORTED.  A PROPER CASPER PROBLEM                 901A3390     000410
00410      C     SETUP MAY NOT HAVE ANY AEROELEMENT FLOW PATH THAT CROSSES               901A3400     000410
00410      C     FROM A NON-VIOLATION AREA TO A VIOLATION AREA WITHOUT                   901A3410     000410
00410      C     CROSSING A DEFINED BOUNDARY SURFACE.                                    901A3420     000410
00410      C-                                                                            901A3430     000410
00411       106  NNN=OR(STAT(I1),2**8)                     @ SET MANDATORY SIFT BIT      901A3440     000412
00412            CALL STSTAT (I1,NNN)                      @                             901A3450     000417
00413            IZVCF=IZVCF+1                             @ BUMP ZERO VELOCITY FLAG     V02-08J      000423
00414            TN=T                                      @ N IMPLIES A POINT IN        901A3460     000426
00415            DO 3480 J=1,4                             @  VIOLATION                  901A3470     000433
00420            UN(J)=UL(J)                               @                             V02-08F      000433
00421       3480 XN(J)=XL(J)                               @                             901A3480     000434
00423            TA=-T                                     @ XL BACK TO ORIGINAL SPOT    901A3490     000437
00424      3500  CALL SPC (XL,UL,TA)                       @                             V02-08F      000442
00425            CALL YNM (XL)                             @ CHECK BOUNDARIES HERE       901A3510     000446
00426            IF (IFZ-IDU) 3529,4100,3529               @ LOST FLOW ZONE TRACK ?      V02-08K      000451
00431      3529  IF (SVMIN) 3530,3530,3600                 @ NO, NON-VIOLATING ?         V02-08K      000454
00434       3530 IF (XL(4)) 3570,3570,3540                 @ NO, CAN WE BACK UP FURTHER ?901A3530     000457
00437       3540 TN=TN-TA                                  @ KEEP TRACK OF TIME SPANS    901A3540     000462
00440            T=T-TA                                    @                             901A3550     000465
00441            GO TO 3500                                @ GO BACK IT UP               901A3560     000470
00442       3570 CALL ERROR2 (901,2)                       @ CAN'T SHAKE BOUNDARY        901A3570     000472
00443            CALL TOGSW (PGSBUG,$3572,$3580)           @ SHALL WE PROVIDE A MESSAGE ?V02-08H      000475
00444      3572  WRITE (LUOUP,3573,END=3580,ERR=3580) @ YES                              V02-08H      000503
00446      3573  FORMAT (67HOCASPER9.MOVELD (BUG) -- UNABLE TO BACK AEROELEMENT OUTV02-08H      000511
00446           1 OF CONFLICT,/,27H  HISTORY BUFFER FOLLOWS...) @                        V02-08H      000511
00447            CALL RBDMP                                @ DUMP THE HISTORY            V02-08H      000511
00450      3580  CALL STIAES (I1,IDU)                      @ THIS TURKEY GOES TO SHEOL   V02-08H      000514
00451            GO TO 221                                 @                             901A3590     000517
00452       3600 TQ=0.0                                    @ Q IMPLIES A POINT NOT IN    901A3600     000521
00453            DO 3620 J=1,4                             @  VIOLATION                  901A3610     000524
00456            UQ(J)=UL(J)                               @                             V02-08F      000524
00457       3620 XQ(J)=XL(J)                               @                             901A3620     000525
00461            IE3F=0                                    @ ERROR 3 ABORT FLAG          901A3630     000530
00462            DO 3790 J=1,15                            @ BISECTION LOOP              901A3640     000534
00465            IF (ABS(TN-TQ)-TE) 3800,3660,3660         @ CLOSE ENOUGH ?              901A3650     000534
00470       3660 TA=0.5*(TN+TQ)                            @ NO, BISECT AGAIN            901A3660     000542
00471            CALL SPC (XA,UA,TA)                       @ FIND THAT POINT IN SPACE    V02-08F      000546
00472            CALL YNM (XA)                             @ TEST THE BOUNDARIES         901A3680     000553
00473            IF (IFZ-IDU) 3664,4100,3664               @ LOST FLOW ZONE TRACK ?      V02-08K      000556
00476      3664  IF (IDMIN) 3710,3710,3700                 @ NO, DID ALL TRUNCATE ?      V02-08K      000561
00501       3700 IE3F=IDMIN                                @ NO, FLAG AN ACTIVE BOUNDARY 901B3700     000564
00502       3710 IF (SVMIN) 3760,3760,3720                 @ A NON-VIOLATING POINT ?     901A3710     000566
00505       3720 TQ=TA                                     @ YES, REPLACE Q POINT        901A3720     000570
00506            DO 3740 K=1,4                             @                             901A3730     000575
00511            UQ(K)=UA(K)                               @                             V01-08F      000575
00512       3740 XQ(K)=XA(K)                               @                             901A3740     000576
00514            IZVCF=0                                   @ SHOW NO ZERO VELOCITY GLITCHV02-08J      000601
00515            GO TO 3790                                @                             901A3750     000602
00516       3760 TN=TA                                     @ NO, REPLACE N POINT         901A3760     000604
00517            DO 3780 K=1,4                             @                             901A3770     000610
00522            UN(K)=UA(K)                               @                             V02-08F      000610
00523       3780 XN(K)=XA(K)                               @                             901A3780     000611
00525       3790 CONTINUE                                  @                             901A3790     000617
00527       3800 IF (IE3F) 3810,3810,3840                  @ CONSTRUCTION PROBLEM ?      901A3800     000617
```

Listing 7.—Continued.

```
00532   3810 CALL ERROR2 (901,3)                  @ YES, REPORT IT                 901A3810   000621
00533        CALL TOGSW (PGSBUG,$3812,$3816)      @ PRINT A MESSAGE ?              V02-08H    000625
00534   3812 WRITE (LUOUP,3813,END=3816,ERR=3816) @ YES                            V02-08H    000633
00536   3813 FORMAT (82HOCASPER9.MOVELD (BUG) -- A BISECTION PROCESS DID NOT ENV02-08H    000641
00536       1COUNTER A GOVERNING SURFACE,/,27H  HISTORY BUFFER FOLLOWS...) @   V02-08H    000641
00537        CALL RBDMP                           @ PRINT HISTORY BUFFER           V02-08H    000641
00540   3816 CALL STIAES (I1,IDU)                 @ FORGET THIS GUY                V02-08H    000644
00541        GO TO 221                            @                                901A3830   000647
00542   3840 IF (IDMIN) 3850,3850,3880            @ ALGORITHM DIDN'T TRACK ?       901A3840   000651
00545   3850 CALL ERROR2 (901,4)                  @ YES, VERY ODD                  901A3850   000653
00546        CALL TOGSW (PGSBUG,$3852,$3856)      @ SHALL WE PRINT A MESSAGE ?     V02-08H    000657
00547   3852 WRITE (LUOUP,3853,END=3856,ERR=3856) @ YES                            V02-08H    000665
00551   3853 FORMAT (71HOCASPER9.MOVELD (BUG) -- BISECTION PROCESS DID NOT FINIV02-08H    000673
00551       1SH ON A BOUNDARY,/,27H  HISTORY BUFFER FOLLOWS...) @              V02-08H    000673
00552        CALL RBDMP                           @ DUMP THIS HISTORY BUFFER       V02-08H    000673
00553   3856 CALL STIAES (I1,IDU)                 @ BLACK HOLE TIME                V02-08H    000676
00554        GO TO 221                            @                                901A3870   000701
00555   3880 IDB=IDMIN                            @ BOUNCE OFF THIS BOUNDARY       901A3880   000703
00556        TLEFT=TLEFT+T-TQ                     @ TIME TO GO AFTER BOUNCE        901A3890   000704
00557        T=TQ                                 @ TIME TO BOUNCE                 901A3920   000710
00560        GO TO 101                            @TRY SHORTER TIME SPAN           90103960   000712
00561    219 CONTINUE                             @END TIME SUB-INCREMENT LOOP     90103970   000715
00563        GO TO 6000                           @ JOIN NORMAL CLEANUP CODE       V02-08K    000715
00564   4100 CALL TOGSW (PGSBUG,$4101,$6000)      @ PRINT ZONE NOT FOUND MSG ?     V02-08K    000717
00565   4101 WRITE (LUOUP,4102,END=6000,ERR=6000) @ YES                            V02-08K    000724
00567   4102 FORMAT (44HOCASPER9.MOVELD (BUG) -- FLOW ZONE NOT FOUND,/,27H  HISV02-08K    000732
00567       1TORY BUFFER FOLLOWS...)              @                                V02-08K    000732
00570        CALL RBDMP                           @ DUMP THIS HISTORY BUFFER       V02-08K    000732
00571   6000 DO 220 I=1,3                         @ RECORD NEW POSITION AND        V02-08K    000737
00574        CALL STS (I1,I,XL(I))                @  VELOCITY RESULTS IN                      000737
00575    220 CALL STS (I1,I+3,UL(I))              @  ASSIGNED SLOTS                           000746
00577        CALL STIAES (I1,IFZ)                 @ RECORD FINAL FLOW ZONE                    000762
00600    221 CONTINUE                             @END ELEMENT BY ELEMENT LOOP     90104010   000770
00602        RETURN                                                                90104140   000770
00603        SUBROUTINE SPC (Y,Z,TI)              @                                V02-08F    001021
00603   C+                                                                         901A4160   001021
00603   C    LOADS VECTOR Y WITH AEROELEMENT POSITION AT TIME 'TI' RELATIVE        901A4170   001021
00603   C    TO 'XL'.  CALCULATES VELOCITY AT Y AND PLACES IT IN Z.                V02-08F    001021
00603   C                                                                          V02-08H    001021
00603   C    BEGINNING WITH VERSION V02-08H, SPC ALSO BEGINS A NEW ENTRY IN        V02-08H    001021
00603   C    THE POSITION-HISTORY RING BUFFER.                                     V02-08H    001021
00603   C-                                                                         901A4190   001021
00606        REAL Y(4),Z(3),TI                    @                                V02-08F    001021
00607        DO 13 N=1,3                                                           901A4210   001021
00612        Y(N)=XL(N)+(TI*UL(N))+(0.5*TI*TI*AL(N))                               V02-08F    001041
00613        IDUM=IRBX1-1+N                       @ FORM INDEX                     V02-08H    001050
00614        ARB(IDUM,IRBPT)=Y(N)                 @ RECORD POSITION                V02-08I    001053
00615   13   Z(N)=UL(N)+(TI*AL(N))                                                 V02-08F    001057
00617        Y(4)=XL(4)+TI                                                         901A4230   001065
00620        ARB(IRBX4,IRBPT)=Y(4)                @ RECORD TIME                    V02-08H    001070
00621        IRB(IRBAI,IRBPT)=I1                  @ RECORD AEROELEMENT ID          V02-08H    001072
00622        IRB(IRBIS,IRBPT)=IRBISA              @ SHOW INITIAL VALIDITY          V02-08H    001074
00623        IRBPP=IRBPT                          @ RECORD PTR FOR FURTHER USE     V02-08I    001076
00624        IRBPT=IRBPT+1                        @ BUMP TO NEXT SLOT SINCE        V02-08H    001100
00625        IF (IRBPT-IRBL) 15,15,14             @  THIS ENTRY IS NOW 'USED'      V02-08H    001102
00630   14   IRBPT=1                              @  ALTHOUGH ADDITIONAL INFO      V02-08H    001105
00631   15   CONTINUE                             @  MAY BE ADDED LATER            V02-08H    001110
00632        RETURN                                                                901A4240   001110
00633        SUBROUTINE YNM (Y)                                                    901A5000   001142
00633   C+                                                                         901A5010   001142
00633   C    1) IDENTIFIES FLOW ZONE OF SPACE-TIME POINT 'Y' AND UPDATES           901A5020   001142
00633   C       FLOW ZONE OF AEROELEMENT 'I1' IF NECESSARY.                        901A5030   001142
00633   C    2) CHECKS ALL APPROPRIATE BOUNDARIES TO PRODUCE 'IDMIN'/'SVMIN'.      901A5040   001142
00633   C       DOES NOT CONSIDER FOR 'IDMIN'/'SVMIN' BOUNDARIES THAT ARE          901A5050   001142
00633   C       SAFE BY TRUNCATION.                                                901A5060   001142
00633   C    3) DISCONTINUES SEARCH IF 'SVMIN' GOES NEGATIVE.                      901A5070   001142
00633   C                                                                          V02-08H    001142
00633   C    BEGINNING WITH VERSION V02-08H, YNM RECORDS IN THE POSITION-          V02-08H    001142
00633   C    HISTORY RING BUFFER THE FLOW ZONE ID AND THE ID AND SURFACE           V02-08H    001142
00633   C    VALUE OF THE GOVERNING SURFACE.                                       V02-08H    001142
00633   C                                                                          V02-08K    001142
00633   C                                                                          V02-08K    001142
```

Listing 7.—Continued.

```
00633    C      BEGINNING WITH VERSION V02-08K, THIS ROUTINE CHECKS FOR THE       V02-08K     001142
00633    C      FLOW-ZONE-NOT-FOUND ERROR AND REPORTS IT TO THE CALLER BY         V02-08K     001142
00633    C                                                                        V02-08K     001142
00633    C      1.  SETTING THE FLOW ZONE TO THE UNUSED ZONE AND                  V02-08K     001142
00633    C                                                                        V02-08K     001142
00633    C      2.  SETTING SVMIN AND IDMIN TO -1.                                V02-08K     001142
00633    C-                                                                       901A5080    001142
00636           REAL Y(4)                                                         901A5090    001142
00637           N=1                                @ SET MANDATORY ZONE FIRST     V02-08K     001142
00640           IDMIN=-1                           @ INIT FOR SURFACE VALUE       V02-08K     001146
00641           SVMIN=1.0E+30                      @  MINIMUM SEARCH              V02-08K     001150
00642           IRB(IRBFI,IRBPP)=IFZ               @ RECORD ENTERING FLOW ZONE    V02-08K     001152
00643           IRB(IRBIS,IRBPP)=OR(IRB(IRBIS,IRBPP),IRBISF) @ MARK IT VALID      V02-08K     001154
00644           CALL TSTZN (Y,IFZ,IFZ)             @ CHECK FLOW ZONE ID           V02-08K     001157
00645           IF (IFZ-1) 29,26,29                @ DID WE FIND A ZONE ?         V02-08K     001164
00650    26     IFZ=IDU                            @ NO, SET HIM UNUSED           V02-08K     001167
00651           CALL STIAES (I1,IDU)               @  HERE, TOO                   V02-08K     001171
00652           SVMIN=-1.0                         @ SET IN VIOLATION             V02-08K     001175
00653           RETURN                             @ BACK TO CALLER               V02-08K     001177
00654    29     IRB(IRBFI,IRBPP)=IFZ               @ REVISE ZONE ID IN RING       V02-08K     001203
00655           NA=IPZBL(N,2)                      @ NUMBER OF BOUNDARIES IN ZONEV02-08K      001206
00656           IF (NA) 42,42,31                   @ SKIP NULL TESTS              901A5180    001213
00661     31    NB=IPZBL(N,1)                      @ POINT TOWARD BOUNDARY ID     901B5190    001215
00662           DO 41 NC=1,NA                      @ DO EACH THIS ZONE            901A5200    001227
00665           NB=NB+1                            @ POINT TO A BOUNDARY ID       901A5210    001227
00666           ID=ZBL(NB)                         @ PICKUP THE ID                901A5220    001232
00667           CALL TSTBDT (Y,ID,SV,IT)           @ GO CHECK THIS BOUNDARY       901A5230    001236
00670           IF (IT) 37,37,41                   @ SKIP MIN IF WE TRUNCATED     901A5240    001244
00673     37    IF (SV-SVMIN) 38,38,41             @ FIND A NEW MIN ?             901A5250    001247
00676     38    SVMIN=SV                           @ YES, RECORD IT               901A5260    001253
00677           IDMIN=ID                           @                              901A5270    001255
00700           IF (SVMIN) 45,45,41                @ GO INTO VIOLATION ?          901A5280    001257
00703     41    CONTINUE                           @ NO, KEEP CHECKING            901A5290    001264
00705     42    IF (N-IFZ) 43,45,43                @ JUST DO EL'S FLOW ZONE ?     901A5300    001264
00710     43    N=IFZ                              @ NO, SWITCH TO IT             901A5310    001266
00711           GO TO 29                           @ GO BACK AND DO IT            901A5320    001270
00712    45     IRB(IRBSI,IRBPP)=IDMIN             @ RECORD SURFACE ID            V02-08H     001272
00713           ARB(IRBSV,IRBPP)=SVMIN             @ RECORD SURFACE VALUE         V02-08H     001275
00714           IRB(IRBIS,IRBPP)=OR(IRB(IRBIS,IRBPP),IRBISS+IRBISV) @ MARK IT     V02-08H     001277
00715           RETURN                             @                              V02-08H     001302
00716           SUBROUTINE RBDMP                                                  V02-08H     001326
00716    C+                                                                       V02-08H     001326
00716    C      THIS SUBROUTINE DUMPS A SYMBOLIC LISTING OF THE POSITION-HISTORY  V02-08H     001326
00716    C      RING BUFFER TO UNIT LUOUP.                                        V02-08H     001326
00716    C-                                                                       V02-08H     001326
00721           I=IRBPP                            @ POINT TO MOST RECENT ENTRY   V02-08H     001326
00722           IF (I) 1050,1050,1002              @ SKIP IF NO VALID ENTRIES     V02-08H     001330
00725    1002   IF (AND(IRB(IRBIS,I),IRBISA)) 1003,1025,1003 @ THIS ENTRY VALID ? V02-08H     001333
00730    1003   WRITE (LUOUP,1075,END=1050,ERR=1050) IRB(IRBAI,I) @ YES, START    V02-08H     001340
00733           WRITE (LUOUP,1076,END=1050,ERR=1050) (ARB(J,I),J=IRBX1,IRBX4) @   V02-08H     001357
00741           IF (AND(IRB(IRBIS,I),IRBISF).NE.0) WRITE (LUOUP,1077,END=1050,ERR=V02-08H     001375
00741          11050) IRB(IRBFI,I)                 @ FLOW ZONE ID                 V02-08H     001375
00745           IF (AND(IRB(IRBIS,I),IRBISS).NE.0) WRITE (LUOUP,1078,END=1050,ERR=V02-08J     001412
00745          11050) IRB(IRBSI,I)                 @ SURFACE ID                   V02-08H     001412
00751           IF (AND(IRB(IRBIS,I),IRBISV).NE.0) WRITE (LUOUP,1079,END=1050,ERR=V02-08H     001427
00751          11050) ARB(IRBSV,I)                 @ SURFACE VALUE                V02-08H     001427
00755           IF (AND(IRB(IRBIS,I),IRBISG).NE.0) WRITE (LUOUP,1080,END=1050,ERR=V02-08H     001444
00755          11050) (ARB(J,I),J=IRBG1,IRBG3)     @ GRADIENT                     V02-08H     001444
00764    1025   I=I-1                              @ NEXT OLDEST ENTRY            V02-08H     001467
00765           IF (I) 1027,1027,1028              @ TIME TO WRAP AROUND ?        V02-08H     001471
00770    1027   I=IRBL                             @ YES                          V02-08H     001473
00771    1028   IF (I-IRBPP) 1002,1050,1002        @ DONE IF BACK AT START        V02-08H     001476
00774    1050   RETURN                             @                              V02-08H     001501
00775    1075   FORMAT (1H0,6X,18HAEROELEMENT ID  = ,I13) @                                   001523
00776    1076   FORMAT (1H ,6X,18HPOSITION        = ,1P,4(E13.7,2X)) @                        001523
00777    1077   FORMAT (1H ,6X,18HFLOW ZONE ID    = ,I13) @                                   001523
01000    1078   FORMAT (1H ,6X,18HSURFACE ID      = ,I13) @                                   001523
01001    1079   FORMAT (1H ,6X,18HSURFACE VALUE   = ,1PE13.7) @                               001523
01002    1080   FORMAT (1H ,6X,18HGRADIENT        = ,1P,3(E13.7,2X)) @                        001523
01003           END                                                               901A9990    001523
END FOR
>
```

Listing 7.—Concluded.

```
 @FOR,MS  CASPER9.SURFD
FOR   4R1   E -04/13/84-15:22:44 (5,)
>@EOF.

   SUBROUTINE SURF       ENTRY POINT 000731
              SURFTR     ENTRY POINT 000747
              GRAD       ENTRY POINT 000765
              SURFTI     ENTRY POINT 001002
              SURFVE     ENTRY POINT 001020
              SURFAC     ENTRY POINT 001035


   STORAGE USED: CODE(1) 001052; DATA(0) 000124; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   IVSIZC 000001


   EXTERNAL REFERENCES (BLOCK, NAME)

    0004   IBDZN
    0005   BDZN
    0006   IPBDZN
    0007   SPOPI
    0010   SPSHI
    0011   SPSHR
    0012   ADDR
    0013   SPOPR
    0014   ADR
    0015   CALSER
    0016   NERR2$
    0017   NERR4$
    0020   NERR3$
   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001     000004 1002L     0001     000015 1010L     0001     000036 1035L     0001     000101 1101L     0001     000161 1111L
    0001     000235 1121L     0001     000261 1127L     0001     000271 1128L     0001     000275 1130L     0001     000311 1132L
    0001     000331 1135L     0001     000335 1140L     0001     000350 1144L     0001     000360 1145L     0001     000370 1146L
    0001     000412 1150L     0001     000416 1153L     0001     000427 1160L     0001     000450 1166L     0001     000517 1190L
    0001     000525 1201L     0001     000573 1218L     0001     000613 1220L     0001     000626 1222L     0001     000636 1223L
    0001     000042 151G      0001     000141 214G      0001     000463 327G      0001     000620 356G      0001     000651 366G
    0005 R 000000 BDZN        0000 R 000001 GR          0004 I 000000 IBDZN       0000 I 000101 IDUM11      0000 I 000103 IDUM12
    0000 I 000073 INHIB       0000   000112 INJP$       0000 I 000070 IP          0006 I 000000 IPBDZN      0000 I 000076 IPN
    0000 I 000104 IPOS        0000 I 000077 IPT         0000 I 000071 IPTH        0000 I 000074 ISW         0000 I 000004 IV
    0003 I 000000 IVSIZE      0000 I 000100 MDE         0000 I 000075 NRTN        0000 I 000102 NV          0000 R 000072 RIDCM
    0000 R 000000 SV          0000 I 000067 THMDE       0000 I 000066 TRCNT


00101      1*         SUBROUTINE SURF (XL,ID,SF,$)                                        91880010    000000
00101      2*     C                                                                       91800020    000000
00101      3*     C                                                                       91800030    000000
00101      4*     C   SURF        ****** A SUBROUTINE FOR CASPER ******                   91800040    000000
00101      5*     C               AUTHOR      WILLIAM HENRY JONES                         91800050    000000
00101      6*     C               V03-00      19 APR 78                                   91800060    000000
00101      7*     C               V03-01      15 JUN 78                                   91810061    000000
00101      8*     C               V03-02      16 JUN 78                                   91820062    000000
00101      9*     C               V03-03      22 JUN 78                                   91830063    000000
00101     10*     C               V03-04      26 JUN 78                                   91840064    000000
00101     11*     C               V03-05      28 JUN 78                                   91850065    000000
00101     12*     C               V03-06      29 JUN 78                                   91860066    000000
00101     13*     C               V03-07      19 JUL 78                                   91870067    000000
00101     14*     C               V03-08      30 AUG 78                                   91880068    000000
00101     15*     C               V03-08A     27 OCT 80     NEW STACK ENTRY POINTS                    000000
00101     16*     C               V03-08B     23 FEB 81     APPLY DC TO CALC QTY ONLY                 000000
00101     17*     C               V03-08C     28 SEP 81     ADD SURFACE VELOCITIES       V03-08C      000000
00101     18*     C               V03-08D     23 APR 82     TRUNCATIONS OF TRUNCATIONS V03-08D        000000
```

Listing 8.—Subroutine SURF.

```
00101   C            V03-08E     05 AUG 83     ADD SURFACE ACCELERATIONS  V03-08E     000000
00101   C                                                                 91800070    000000
00101   C                                                                 91800080    000000
00101   C                                                                 91800090    000000
00101   C   ARGUMENTS PASSED IN SUBROUTINE CALL ******                    91880094    000000
00101   C                                                                 91880096    000000
00101   C   ARGUMENT    TYPE        DIMENSION     DESCRIPTION             91800100    000000
00101   C   ---------   ---------   ---------     ------------------      91800110    000000
00101   C   XL          REAL        1 TO 4        3 POSITION COORDS AND TIME 91800120 000000
00101   C                                                                 91800130    000000
00101   C   ID          INTEGER     SCALAR        BOUNDARY OR FLOW ZONE   91800140    000000
00101   C                                          IDENTIFICATION NUMBER  91800150    000000
00101   C                                                                 91800260    000000
00101   C   SF          REAL        SCALAR        CALCULATED SURFACE VALUE 91800270   000000
00101   C                                                                 91800280    000000
00101   C   *4          N/A         RETURN LINK   TAKEN IF TRUNCATION OCCURED91880290 000000
00101   C                                                                 91800300    000000
00101   C   GF          REAL        1 TO 3        GRADIENT, REPLACES SF IN 91800310   000000
00101   C                                          SECOND ENTRY POINT     91800320    000000
00101   C                                                                 91880330    000000
00101   C   ARGUMENTS IN CASPER 'CACHE' MEMORY ******                     91880332    000000
00101   C                                                                 91880334    000000
00101   C   ARGUMENT    TYPE        DIMENSION     DESCRIPTION             91880336    000000
00101   C   ---------   ---------   ---------     ------------------      91880338    000000
00101   C   BDZN        REAL        1 TO BDZNSZ   SURFACE EVALUATION ROUTINE 91880340 000000
00101   C                                          PARAMETERS LIST        91880342    000000
00101   C                                                                 91880344    000000
00101   C   IBDZN       INTEGER     1 TO BDZNSZ   SURFACE EVALUATION ROUTINE 91880346 000000
00101   C                                          PARAMETERS (CONTROL) LIST 91880348 000000
00101   C                                                                 91880350    000000
00101   C   IPBDZN      INTEGER     1 TO NBDZN    BDZN CONTROL POINTERS LIST 91880352 000000
00101   C                                          (ID) = SDH POINTER     91880354    000000
00101   C                                                                 91880356    000000
00101   C                                                                 91880358    000000
00101   C                                                                 91880360    000000
00101   C   DESCRIPTION ******                                            91880362    000000
00101   C                                                                 91800370    000000
00101   C   SURF HANDLES THE EVALUATION OF ALL BOUNDARIES AND TRUNCATION  91800380    000000
00101   C   SURFACES.  IT PROVIDES ALL NECESSARY LOGIC TO HANDLE TRUNCATION 91800390  000000
00101   C   SURFACES, FIND SURFACE EVALUATION ROUTINE PARAMETER LISTS,    91800400    000000
00101   C   AND VECTOR CONTROL TO PROPER SURFACE EVALUATION ROUTINES.     91800410    000000
00101   C                                                                 91800420    000000
00101   C   ****** BDZN DATA BASE ******                                  91800430    000000
00101   C                                                                 91800440    000000
00101   C   SURFACE DEFINITION HEADER ******                              91800450    000000
00101   C                                                                 91800460    000000
00101   C   EACH BOUNDARY AND FLOW ZONE IN CASPER IS DEFINED BY MEANS OF A 91800470   000000
00101   C   SERIES OF NUMBERS IN THE CONTROL LIST 'BDZN'.  A POINTER      91800480    000000
00101   C   TO THESE NUMBERS CAN BE OBTAINED FROM 'IPBDZN(N)' WHERE 'N' IS 91800490   000000
00101   C   THE ID NUMBER OF THE DESIRED BOUNDARY OR FLOW ZONE.  WHAT IS  91800500    000000
00101   C   OBTAINED IS A POINTER TO A SURFACE DEFINITION HEADER, 'SDH'.  91800510    000000
00101   C   THE POINTER IS REFERRED TO AS 'IP'.  THE HEADER APPEARS AS    91800520    000000
00101   C   FOLLOWS.                                                      91800530    000000
00101   C                                                                 91800540    000000
00101   C   IP-5        SURFACE TYPE NUMBER                               91800550    000000
00101   C   IP-4        SURFACE PARAMETER LOCATION CONTROL (SPLC)         91800560    000000
00101   C   IP-3        TRUNCATION HEADER LOCATION CONTROL (THLC)         91800570    000000
00101   C   IP-2        DEFEND/CONTAIN SURFACE VALUE                      91800580    000000
00101   C   IP-1        NUMBER OF TRUNCATION HEADERS                      91800590    000000
00101   C   IP          NUMBER OF PARAMETERS FOLLOWING THIS HEADER        91800600    000000
00101   C                                                                 91800610    000000
00101   C                                                                 91800620    000000
00101   C   SURFACE PARAMETER LOCATION CONTROL (SPLC) ******              91800630    000000
00101   C                                                                 91800640    000000
00101   C   THIS NUMBER PROVIDES THE METHOD FOR CALCULATING THE POINTER(S) 91800650   000000
00101   C   TO THE ACTUAL NUMBERS WHICH ARE USED FOR CALCULATING THE SURFACE 91800660 000000
00101   C   VALUE.  THE METHODS ARE AS FOLLOWS.                           91800670    000000
00101   C                                                                 91800680    000000
00101   C   MODE        METHOD                                            91800690    000000
00101   C   ------      ------------                                      91800700    000000
00101   C   1           IV = IP+1                                         91800710    000000
00101   C   2           IV = IBDZN(IP+1)                                  91800720    000000
```

Listing 8.—Continued.

```
00101   C   3           IV = IBDZN(IBDZN(IP+1))                                 91800730   000000
00101   C   4           IV = IP+IBDZN(IP+1)                                     91800740   000000
00101   C   5           IV = IBDZN(IP+IBDZN(IP+1))                              91800750   000000
00101   C   6           IV = IBDZN(IBDZN(IP+1)+IBDZN(IP+2))                     91800760   000000
00101   C   7           IV = IBDZN(IBDZN(IP+1)+IBDZN(IBDZN(IP+2)+IBDZN(IP+3)))91800770   000000
00101   C   8           IV = IBDZN(IBDZN(IP+1)+IBDZN(IBDZN(IP+2)+IBDZN(2+       91800780   000000
00101   C                IOFSET)))                                              91800790   000000
00101   C   9           IV = IBDZN(IBDZN(IP+1)+IBDZN(IBDZN(IP+2)+IBDZN(IBDZN    91800800   000000
00101   C                (IP+3)+IBDZN(3+IOFSET))))                              91800810   000000
00101   C   10          MULTIPLE MODES - FORMAT AS FOLLOWS                      91800820   000000
00101   C                                                                       91800830   000000
00101   C               IP ->        TOTAL NUMBER OF PARAMETERS FOLLOWING THIS 91800840   000000
00101   C                             SURFACE DEFINITION HEADER                 91800850   000000
00101   C                            NUMBER OF ADDRESSING MODES TO BE EVALUATED91800860   000000
00101   C                            FIRST SPLC                                 91800870   000000
00101   C                (IPE 1 ->) NUMBER OF PARAMETERS FOLLOWING THIS ENTRY 91800880   000000
00101   C                            PARAMETER                                  91800890   000000
00101   C                            PARAMETER                                  91800900   000000
00101   C                            PARAMETER                                  91800910   000000
00101   C                            ...                                        91800920   000000
00101   C                            SECOND SPLC                                91800930   000000
00101   C                (IPE 2 ->) NUMBER OF PARAMETERS FOLLOWING THIS ENTRY 91800940   000000
00101   C                            PARAMETER                                  91800950   000000
00101   C                            PARAMETER                                  91800960   000000
00101   C                            PARAMETER                                  91800970   000000
00101   C                            ...                                        91800980   000000
00101   C                            ...                                        91800990   000000
00101   C                            N TH SPLC                                  91801000   000000
00101   C                (IPE N ->) NUMBER OF PARAMETERS FOLLOWING THIS ENTRY 91801010   000000
00101   C                            PARAMETER                                  91801020   000000
00101   C                            PARAMETER                                  91801030   000000
00101   C                            PARAMETER                                  91801040   000000
00101   C                            ....                                       91801050   000000
00101   C                                                                       91801060   000000
00101   C   NOTE THAT THE POINTERS GENERATED POINT TO THE ACTUAL LOCATION       91801070   000000
00101   C   IN BDZN OF THE FIRST PARAMETER OF ANY LIST (OR SET OF LISTS), NOT 91801080   000000
00101   C   THE LOCATION BEFORE IT AS IN PREVIOUS INCARNATIONS.  THUS, THE      91801090   000000
00101   C   FIRST PARAMETER IS #0, NOT #1 .                                     91801100   000000
00101   C                                                                       91801110   000000
00101   C   SINCE ALL SURFACE EVALUATION ROUTINES MAY DIVIDE THE COMPLETE       91801120   000000
00101   C   PARAMETER SET INTO DIFFERENT LISTS IN DIFFERENT WAYS, THE           91801130   000000
00101   C   RESPONSIBILITY FOR CORRECTING POINTERS LIES WITH THESE ROUTINES.  91801140   000000
00101   C   AFTER ONE OF THESE ROUTINES IS CALLED IT MUST BE ASSUMED THAT       91801150   000000
00101   C   THE SUPPLIED POINTERS HAVE BEEN SO ALTERED.                         91801160   000000
00101   C                                                                       91801170   000000
00101   C   WHEN FEWER INDEPENDENT ADDRESSING POINTERS THAN PARAMETER LISTS     91801180   000000
00101   C   TO BE LOCATED ARE SUPPLIED THE UN-LOCATED LISTS ARE ASSUMED         91801190   000000
00101   C   TO FOLLOW THE LAST LOCATED LIST CONTIGUOUSLY IN BDZN AND THE        91801200   000000
00101   C   ASSOCIATED DEPENDENT POINTERS ARE CORRECTED ACCORDINGLY.            91801210   000000
00101   C                                                                       91801220   000000
00101   C                                                                       91801230   000000
00101   C   TRUNCATION HEADER LOCATION CONTROL (THLC) ******                    91801240   000000
00101   C                                                                       91801250   000000
00101   C   THIS PARAMETER SPECIFIES THE METHOD OF LOCATING THE APPROPRIATE     91801260   000000
00101   C   TRUNCATION HEADERS.  A TRUNCATION POINTER, 'IPT', SIMILAR TO THE   91801270   000000
00101   C   SURFACE PARAMETER POINTER, 'IP', IS DEFINED AS                      91801280   000000
00101   C                                                                       91801290   000000
00101   C   IPT = IP + 1 + IBDZN(IP) ,                                          91801300   000000
00101   C                                                                       91801310   000000
00101   C   IT CAN BE SEEN THAT IBDZN(IP) CONTAINS THE NUMBER OF PARAMETERS     91801320   000000
00101   C   FOLLOWING THE SURFACE DEFINITION HEADER THAT ARE ASSOCIATED WITH   91801330   000000
00101   C   THE DESCRIPTION OF THE SURFACE - NOT WITH ITS TRUNCATION.           91801340   000000
00101   C                                                                       91801350   000000
00101   C   THE ADDRESSING MODES USED TO LOCATE TRUNCATION HEADERS ARE          91801360   000000
00101   C   IDENTICAL TO THOSE USED FOR SURFACE PARAMETER LOCATION              91801370   000000
00101   C   EXCEPT THAT ABSOLUTE LOCATIONS 2 AND 3 ARE REPLACED BY              91801380   000000
00101   C   ABSOLUTE LOCATIONS 4 AND 5, RESPECTIVELY.                           91801390   000000
00101   C                                                                       91801400   000000
00101   C   IBDZN(IPT) CONTAINS THE NUMBER OF PARAMETERS FOLLOWING THE 'IPT'   91801410   000000
00101   C   LOCATION THAT ARE ASSOCIATED WITH THE SURFACE DEFINITION            91801420   000000
00101   C   TRUNCATION.  ASSUMING THAT TWO COMPLETE SURFACE DEFINITION          91801430   000000
00101   C   HEADERS OCCUPY CONTIGUOUS AREAS IN BDZN/IBDZN, THE POINTER TO       91801440   000000
```

Listing 8.—Continued.

```
00101   C   THE SECOND, 'IP2', CAN BE FOUND AS FOLLOWS BASED ON THE POINTER   91801450   000000
00101   C   TO THE FIRST, 'IP1'.                                              91801460   000000
00101   C                                                                     91801470   000000
00101   C   IPT1 = IP1 + 1 + IBDZN(IP1)                                       91801480   000000
00101   C   IP2 = IPT1 + 6 + IBDZN(IPT1)                                      91801490   000000
00101   C                                                                     V03-08D    000000
00101   C                                                                     V03-08D    000000
00101   C   TRUNCATION RECURSION ******                                       V03-08D    000000
00101   C                                                                     V03-08D    000000
00101   C   THE TRUNCATION PROCESS IS RECURSIVE.  IF A TRUNCATION IS, ITSELF, V03-08D    000000
00101   C   TRUNCATED, IT NO LONGER HAS ANY EFFECT UPON ITS PARENT SURFACE.   V03-08D    000000
00101   C   IF THE PARENT SURFACE IS A TRUNCATION SURFACE, IT MUST STILL BE   V03-08D    000000
00101   C   EVALUATED AND MAY TRUNCATE ITS PARENT.                            V03-08D    000000
00101   C                                                                     V03-08D    000000
00101   C   USING THIS MECHANISM, TRUNCATION SURFACES MAY BE BUILT BY THE     V03-08D    000000
00101   C   CONCATENATION OF INDIVIDUAL SURFACES.                             V03-08D    000000
00101   C                                                                     91801500   000000
00101   C                                                                     91801510   000000
00101   C                                                                     91801520   000000
00101   C   TRUNCATION HEADER ******                                          91801530   000000
00101   C                                                                     91801540   000000
00101   C   TRUNCATION HEADERS PROVIDE THE INFORMATION NECESSARY TO LOCATE    91801550   000000
00101   C   AND MODIFY (AS APPROPRIATE) THE SURFACE DEFINITION HEADER FOR AN  91801560   000000
00101   C   INDIVIDUAL TRUNCATION SURFACE.  ONE TRUNCATION HEADER MUST BE     91801570   000000
00101   C   SUPPLIED FOR EACH TRUNCATION LISTED BY THE PARENT SURFACE         91801580   000000
00101   C   DEFINITION HEADER.  IF THE SUPPLIED ADDRESSING INFORMATION        91801590   000000
00101   C   DOES NOT GIVE ENOUGH INDEPENDENT POINTERS TO TRUNCATION HEADERS   91801600   000000
00101   C   THE AVAILABLE N POINTERS ARE ASSUMED TO POINT TO THE FIRST        91801610   000000
00101   C   N TRUNCATION HEADERS AND THE REMAINING TRUNCATION HEADERS         91801620   000000
00101   C   ARE ASSUMED TO FOLLOW CONSECUTIVELY AFTER THE N TH HEADER.        91801630   000000
00101   C                                                                     91801640   000000
00101   C   MODIFICATION OF THE TRUNCATION SURFACE'S SURFACE DEFINITION       91801650   000000
00101   C   HEADER CONSISTS SOLELY OF AN ADDITIONAL DEFEND/CONTAIN            91801660   000000
00101   C   MODIFIER TO FLOP POSITIVE AND NEGATIVE SIDES OF THE SURFACE       91801670   000000
00101   C   USED.  THUS, ANY DEFINED SURFACE CAN BE REUSED ANY NUMBER         91801680   000000
00101   C   OF TIMES TO TRUNCATE OTHER SURFACES.                              91801690   000000
00101   C                                                                     91801700   000000
00101   C   A POINTER, 'IPTH', TO THE FIRST WORD OF THE TRUNCATION HEADER     91801710   000000
00101   C   IS ASSUMED.  THE TRUNCATION HEADER FORMAT IS AS FOLLOWS           91801720   000000
00101   C                                                                     91801730   000000
00101   C   POSITION    REMARKS                                               91801740   000000
00101   C   ---------   ------------------                                    91801750   000000
00101   C   IPTH        MODE NUMBER                                           91801760   000000
00101   C   IPTH+1      DEFEND/CONTAIN MODIFIER                               91801770   000000
00101   C   IPTH+2      ADDITIONAL PARAMETER(S)                               91801780   000000
00101   C                                                                     91801790   000000
00101   C   THREE MODES ARE DEFINED                                           91801800   000000
00101   C                                                                     91801810   000000
00101   C   MODE        SDH LOCATION                                          91801820   000000
00101   C   ------      ------------------                                    91801830   000000
00101   C   1          IP = IPTH + 8 -- COMPLETE SURFACE DEFINITION HEADER    91801840   000000
00101   C               WITH ALL APPROPRIATE PARAMETERS AND TRUNCATION        91801850   000000
00101   C               INFORMATION FOLLOWS IMMEDIATELY.                      91801860   000000
00101   C                                                                     91801870   000000
00101   C   2          IP = IPBDZN(IBDZN(IPTH+2)) -- THE IDENTIFICATION       91801880   000000
00101   C               NUMBER OF THE SURFACE IS IN IPTH+2.  THIS IS          91801890   000000
00101   C               A SURFACE EXTERNALLY DEFINED IN THE BOUNDARY          91801900   000000
00101   C               AND FLOW ZONE POINTERS LIST, 'IPBDZN'.                91801910   000000
00101   C                                                                     91801920   000000
00101   C   3          IP = IBDZN(IBDZN(1)+IBDZN(IPTH+2)) -- THE              91801930   000000
00101   C               IDENTIFICATION NUMBER OF THE SURFACE IS IN            91801940   000000
00101   C               IPTH+2.  THIS IS A SURFACE INTERNALLY DEFINED         91801950   000000
00101   C               IN BDZN/IBDZN.  A TABLE OF POINTERS TO THE            91801960   000000
00101   C               SURFACE DEFINITION HEADERS FOR THESE SURFACES         91801970   000000
00101   C               IS POINTED TO BY IBDZN(1).  EXCEPT FOR THE            91801980   000000
00101   C               INTERNAL/EXTERNAL DEFINITION DIFFERENCE, ALL          91801990   000000
00101   C               OTHER FEATURES OF THE EXTERNALLY DEFINED SURFACE      91802000   000000
00101   C               ARE AVAILABLE TO THE INTERNALLY DEFINED SURFACE.      91802010   000000
00101   C                                                                     91802020   000000
00101   C   NOTE -- IF THE MODE NUMBER IS NEGATIVE SUBTRUNCATIONS OF THE      91832022   000000
00101   C          CITED TRUNCATION SURFACE ARE INHIBITED.                    91832023   000000
00101   C                                                                     91802030   000000
```

Listing 8.—Continued.

```
00101  C                                                                    91802040  000000
00101  C   BDZN/IBDZN RESERVED SPACE ******                                 91802050  000000
00101  C                                                                    91802060  000000
00101  C   AS INFERRED PREVIOUSLY A NUMBER OF WORDS IN BDZN ARE RESERVED    91802070  000000
00101  C   FOR USE BY THE ADDRESSING MODE INTERPRETER.  THEY ARE AS         91802080  000000
00101  C   FOLLOWS --                                                       91802090  000000
00101  C                                                                    91802100  000000
00101  C   WORD        USE                                                  91802110  000000
00101  C   ---------   ------------------                                   91802120  000000
00101  C   1           POINTER TO BASE OF INTERNALLY IDENTIFIED SURFACE     91802130  000000
00101  C                SDH POINTERS TABLE.                                 91802140  000000
00101  C   2           POINTER TO THE TABLE OF BOUNDARY AND FLOW ZONE       91802150  000000
00101  C                SURFACE TABLES.                                     91802160  000000
00101  C   3           BELIEVE IT OR NOT, THIS IS A POINTER TO A TABLE      91802170  000000
00101  C                TABLE TABLE.  THIS PROVIDES A CROSS INDEXING        91802180  000000
00101  C                FEATURE BETWEEN TYPES OF SURFACE.  FOR INSTANCE,    91802190  000000
00101  C                A SETUP USING THIS FEATURE COULD SWITCH A PROBLEM   91802200  000000
00101  C               BETWEEN STATIC AND DYNAMIC BOUNDARIES OF IDENTICAL   91882210  000000
00101  C                TYPES BY SIMPLY REWRITING THE TABLE TABLE TABLE.    91802220  000000
00101  C                OF COURSE, YOU WOULD GO CRAZY....                   91802230  000000
00101  C   4 AND 5     AS 2 AND 3, RESPECTIVELY, BUT USED WHEN SEEKING      91802240  000000
00101  C                TRUNCATION HEADERS.                                 91802250  000000
00101  C                                                                    91802260  000000
00101  C                                                                    91802270  000000
00101  C                                                                    91802280  000000
00101  C   EXAMPLE ******                                                   91802290  000000
00101  C                                                                    91802300  000000
00101  C   THE FOLLOWING EXAMPLE SHOWS HOW THE FOREGOING CONVENTIONS MAY    91802310  000000
00101  C   BE APPLED TO EFFECTIVELY DESCRIBE THE ROTATING BLADES OF A       91802320  000000
00101  C   TURBOJET FAN MODULE.  IT IS ASSUMED THAT ALL THESE BLADES CAN    91802330  000000
00101  C   BE DESCRIBED BY A SINGLE SURFACE EVALUATION ROUTINE.  THIS       91802340  000000
00101  C   EXAMPLE ALSO ASSUMES THAT OTHER ELEMENTS OF SUCH A FAN MODULE    91802350  000000
00101  C   ARE ALSO BEING DESCRIBED AT THAT SAME TIME.                      91802360  000000
00101  C                                                                    91802370  000000
00101  C   SURFACE EVALUATION ROUTINE ******                                91802380  000000
00101  C                                                                    91802390  000000
00101  C   THE GENERALIZED ROUTINE WHICH SPECIFIES THE BLADES IS ASSUMED    91802400  000000
00101  C   TO REQUIRE A PARAMETER LIST AS FOLLOWS --                        91802410  000000
00101  C                                                                    91802420  000000
00101  C   REF LOC     SEG NO      REMARKS                                  91802430  000000
00101  C   ---------   ---------   ------------------                       91802440  000000
00101  C   0           1           THESE 13 PARAMETERS SPECIFY A COORDINATE 91802450  000000
00101  C   1           1            TRANSFORMATION AS A FUNCTION OF TIME.   91802460  000000
00101  C   2           1            THEY ARE USED TO DETERMINE DISK/BLACE   91802470  000000
00101  C   3           1            LOCATION AT ANY TIME.                   91802480  000000
00101  C   4           1                                                    91802490  000000
00101  C   5           1                                                    91802500  000000
00101  C   6           1                                                    91802510  000000
00101  C   7           1                                                    91802520  000000
00101  C   8           1                                                    91802530  000000
00101  C   9           1                                                    91802540  000000
00101  C   10          1                                                    91802550  000000
00101  C   11          1                                                    91802560  000000
00101  C   12          1                                                    91802570  000000
00101  C   13          2           ENGINE 'SPEED' PARAMETER                 91802580  000000
00101  C   14          3           THESE 12 PARAMETERS SPECIFY A COORDINATE 91802590  000000
00101  C   15          3            TRANSFORMATION TO LOCATE THE BLADE      91802600  000000
00101  C   16          3            ON THE DISK GIVEN THE RADIUS OF THE     91802610  000000
00101  C   17          3            DISK AND THE BLADE NUMBER               91802620  000000
00101  C   18          3                                                    91802630  000000
00101  C   19          3                                                    91802640  000000
00101  C   20          3                                                    91802650  000000
00101  C   21          3                                                    91802660  000000
00101  C   22          3                                                    91802670  000000
00101  C   23          3                                                    91802680  000000
00101  C   24          3                                                    91802690  000000
00101  C   25          3                                                    91802700  000000
00101  C   26          4           RADIUS OF DISK                           91802710  000000
00101  C   27          5           BLADE NUMBER ON DISK                     91802720  000000
00101  C   28          6           THE PARAMETERS OF GROUP 6 SPECIFY A      91802730  000000
00101  C   29          6            BASIC AIRFOIL CHORDWISE THICKNESS       91802740  000000
00101  C   30          6            DISTRIBUTION                            91802750  000000
```

Listing 8.—Continued.

```
00101  C  31         6                                                91802760  000000
00101  C  32         6                                                91802770  000000
00101  C  33         6                                                91802780  000000
00101  C  34         6                                                91802790  000000
00101  C  35         6                                                91802800  000000
00101  C  36         6                                                91802810  000000
00101  C  37         6                                                91802820  000000
00101  C  38         6                                                91802830  000000
00101  C  39         7          THE PARAMETERS OF GROUP 7 SPECIFY A   91802840  000000
00101  C  40         7           RADIAL MODIFICATION OF CHORDWISE     91802850  000000
00101  C  41         7           THICKNESS DISTRIBUTION               91802860  000000
00101  C  42         7                                                91802870  000000
00101  C  43         7                                                91802880  000000
00101  C  44         7                                                91802890  000000
00101  C  45         7                                                91802900  000000
00101  C  46         7                                                91802910  000000
00101  C  47         8          THE PARAMETERS OF GROUP 8 SPECIFY A   91802920  000000
00101  C  48         8           RADIAL DISTRIBUTION OF THICKNESS     91802930  000000
00101  C  49         8                                                91802940  000000
00101  C  50         8                                                91802950  000000
00101  C  51         8                                                91802960  000000
00101  C  52         8                                                91802970  000000
00101  C  53         9          THE PARAMETERS OF GROUP 9 SPECIFY A   91802980  000000
00101  C  54         9           CAMBER FOR THE BLADE                 91802990  000000
00101  C  55         9                                                91803000  000000
00101  C  56         9                                                91803010  000000
00101  C  57         9                                                91803020  000000
00101  C  58         9                                                91803030  000000
00101  C  59         10         THE PARAMETERS OF GROUP 10 SPECIFY    91803040  000000
00101  C  60         10          THE RADIAL STACKING OF THE BLADE     91803050  000000
00101  C  61         10                                               91803060  000000
00101  C  62         10                                               91803070  000000
00101  C  63         10                                               91803080  000000
00101  C  64         10                                               91803090  000000
00101  C  65         10                                               91803100  000000
00101  C  66         10                                               91803110  000000
00101  C  67         10                                               91803120  000000
00101  C  68         10                                               91803130  000000
00101  C  69         10                                               91803140  000000
00101  C  70         10                                               91803150  000000
00101  C                                                              91803160  000000
00101  C  BDZN/IBDZN LIST ******                                      91803170  000000
00101  C                                                              91803180  000000
00101  C  LOCATION   VALUE      REMARKS                               91803190  000000
00101  C  --------   ---------  ------------------                    91803200  000000
00101  C  1                                                           91803210  000000
00101  C  2          TT         POINTER TO TABLE TABLE                91803220  000000
00101  C  3                                                           91803230  000000
00101  C  4                                                           91803240  000000
00101  C  5                                                           91803250  000000
00101  C  ...                                                         91803260  000000
00101  C  TT         T0        ROTOR ORIENTATIONS TABLE POINTER       91803270  000000
00101  C  TT+1       T1        ROTOR SPEEDS TABLE POINTER             91803280  000000
00101  C  TT+2       T2        BLADE ORIENTATIONS TABLE POINTER       91803290  000000
00101  C  TT+3       T3        BLADE NUMBER TABLE POINTER             91803300  000000
00101  C  TT+4       T4        BLADE MECHANICAL DESCRIPTION TABLE POINTER91803310  000000
00101  C  TT+5       T5        OTHER TABLE POINTERS                   91803320  000000
00101  C  TT+6       T6         .                                     91803330  000000
00101  C  TT+7       T7         .                                     91803340  000000
00101  C  ...                                                         91803350  000000
00101  C  T0         LOWOR      POINTER TO LOW ROTOR ORIENTATION      91803360  000000
00101  C  T0+1       HIGHOR     POINTER TO HIGH ROTOR ORIENTATION     91803370  000000
00101  C  ...                                                         91803380  000000
00101  C  T1         N1         POINTER TO LOW ROTOR SPEED            91803390  000000
00101  C  T1+1       N2         POINTER TO HIGH ROTOR SPEED           91803400  000000
00101  C  ...                                                         91803410  000000
00101  C  T2         BD11       POINTER TO ORIENTER OF BLADES - FAN DISK 191803420  000000
00101  C  T2+1       BD21        FAN DISK 2                           91803430  000000
00101  C  T2+2       BD31        FAN DISK 3                           91803440  000000
00101  C  T2+3       BD12        CORE DISK 1                          91803450  000000
00101  C  T2+4       BD22        CORE DISK 2                          91803460  000000
00101  C  T2+5       BD32        CORE DISK 3                          91803470  000000
```

Listing 8.—Continued.

```
00101    379*    C    T2+6        BD42      CORE DISK 4                                91803480    000000
00101    380*    C    T2+7        BD52      CORE DISK 5                                91803490    000000
00101    381*    C    T2+8        BD62      CORE DISK 6                                91803500    000000
00101    382*    C    ...                                                              91803510    000000
00101    383*    C    T3          B1       POINTERS TO BLADE NUMBERS (LARGEST          91803520    000000
00101    384*    C    T3+1        B2        NUMBER OF BLADES ON ANY DISK THIS          91803530    000000
00101    385*    C    T3+2        B3        EXAMPLE IS 120).                           91803540    000000
00101    386*    C    T3+3        B4                                                   91803550    000000
00101    387*    C    T3+4        B5                                                   91803560    000000
00101    388*    C    ...         ...                                                  91803570    000000
00101    389*    C    T3+119      B120                                                 91803580    000000
00101    390*    C    ...                                                              91803590    000000
00101    391*    C    T4          FB1      POINTERS TO FAN BLADE (FB) AND CORE         91803600    000000
00101    392*    C    T4+1        FB2       BLADE (CB) MECHANICAL DESCRIPTIONS         91803610    000000
00101    393*    C    T4+2        FB3       FOR EACH ROTOR.  (THESE POINT TO           91803620    000000
00101    394*    C    T4+3        CB1       PARAMETERS 28 TO 70 OF THE BLADE           91803630    000000
00101    395*    C    T4+4        CB2       SURFACE EVALUATION ROUTINE).               91803640    000000
00101    396*    C    T4+5        CB3                                                  91803650    000000
00101    397*    C    T4+6        CB4                                                  91803660    000000
00101    398*    C    T4+7        CB5                                                  91803670    000000
00101    399*    C    T4+8        CB6                                                  91803680    000000
00101    400*    C    ...                                                              91803690    000000
00101    401*    C    LOWOR       XXX      GROUP 1 PARAMETERS FOR BLADES OF            91803700    000000
00101    402*    C    LOWOR+1     XXX       THE LOW ROTOR                              91803710    000000
00101    403*    C    LOWOR+2     XXX                                                  91803720    000000
00101    404*    C    LOWOR+3     XXX                                                  91803730    000000
00101    405*    C    ...         ...                                                  91803740    000000
00101    406*    C    LOWOR+12    XXX                                                  91803750    000000
00101    407*    C    ...                                                              91803760    000000
00101    408*    C    HIGHOR      XXX      GROUP 1 PARAMETERS FOR BLADES OF THE HIGH   91803770    000000
00101    409*    C    HIGHOR+1    XXX       ROTOR                                      91803780    000000
00101    410*    C    HIGHOR+2    XXX                                                  91803790    000000
00101    411*    C    HIGHOR+3    XXX                                                  91803800    000000
00101    412*    C    ...         ...                                                  91803810    000000
00101    413*    C    HIGHOR+12   XXX                                                  91803820    000000
00101    414*    C    ...                                                              91803830    000000
00101    415*    C    N1          9000     LOW ROTOR SPEED                             91803840    000000
00101    416*    C    N2          12000    HIGH ROTOR SPEED                            91803850    000000
00101    417*    C    ...                                                              91803860    000000
00101    418*    C    BD11        XXX      GROUP 3 PARAMETERS FOR THE BLADES OF        91803870    000000
00101    419*    C    BD11+1      XXX       FAN ROTOR 1                                91803880    000000
00101    420*    C    BD11+2      XXX                                                  91803890    000000
00101    421*    C    BD11+3      XXX                                                  91803900    000000
00101    422*    C    ...                                                              91803910    000000
00101    423*    C    BD21        XXX      GROUP 3 PARAMETERS FOR THE BLADES OF        91803920    000000
00101    424*    C    BD21+1      XXX       FAN ROTOR 2                                91803930    000000
00101    425*    C    BD21+2      XXX                                                  91803940    000000
00101    426*    C    BD21+3      XXX                                                  91803950    000000
00101    427*    C    ...                                                              91803960    000000
00101    428*    C    BD32        XXX      GROUP 3 PARAMETERS FOR THE BLADES OF        91803970    000000
00101    429*    C    BD32+1      XXX       CORE ROTOR 3                               91803980    000000
00101    430*    C    BD32+2      XXX                                                  91803990    000000
00101    431*    C    BD32+3      XXX                                                  91804000    000000
00101    432*    C    ...                                                              91804010    000000
00101    433*    C    B1          1        BLADES IN THIS EXAMPLE ARE NUMBERED IN      91804020    000000
00101    434*    C    B2          2         CIRCUMFERENTIAL SEQUENCE; HOWEVER, THIS    91804030    000000
00101    435*    C    B3          3         NEED NOT BE SO WHEN USING THIS METHOD      91804040    000000
00101    436*    C    B4          4                                                    91804050    000000
00101    437*    C    ...                                                              91804060    000000
00101    438*    C    FB1         XXX      GROUP 6 THROUGH GROUP 10 PARAMETERS         91804070    000000
00101    439*    C    FB1+1       XXX       FOR BLADES ON THE FIRST FAN DISK           91804080    000000
00101    440*    C    FB1+2       XXX                                                  91804090    000000
00101    441*    C    FB1+3       XXX                                                  91804100    000000
00101    442*    C    ...                                                              91804110    000000
00101    443*    C    CB2         XXX      GROUP 3 THROUGH GROUP 10 PARAMETERS         91804120    000000
00101    444*    C    CB2+1       XXX       FOR BLADES ON THE SECOND CORE DISK         91804130    000000
00101    445*    C    CB2+2       XXX                                                  91804140    000000
00101    446*    C    CB2+3       XXX                                                  91804150    000000
00101    447*    C    ...                                                              91804160    000000
00101    448*    C                                                                     91804170    000000
00101    449*    C                         THE FOLLOWING IS THE SURFACE DEFINITION     91804180    000000
00101    450*    C                          HEADER FOR BLADE 17 OF FAN ROTOR 1.        91804190    000000
```

Listing 8.—Continued.

```
00101   451*   C                           THIS BLADE'S ID NUMBER IS 61.              91804200   000000
00101   452*   C   IP-5      711          BLADE SURFACE TYPE NUMBER                  91804210   000000
00101   453*   C   IP-4      10           LOCATE PARAMETERS BY MULTIMODES            91804220   000000
00101   454*   C   IP-3      1            TRUNCATION HEADERS ARE IMMEDIATE           91804230   000000
00101   455*   C   IP-2      1.0          THIS IS A DEFEND SURFACE                   91804240   000000
00101   456*   C   IP-1      2            THERE ARE TWO TRUNCATION HEADERS           91804250   000000
00101   457*   C   IP        24           NUMBER OF FOLLOWING PARAMETERS             91804260   000000
00101   458*   C   IP+1      6            NUMBER OF MULTIMODES TO EVALUATE           91804270   000000
00101   459*   C   IP+2      8            FIRST MULTIMODE - MODE 8                   91804280   000000
00101   460*   C   IP+3      2             NUMBER OF PARAMETERS THIS MODE            91804290   000000
00101   461*   C   IP+4      0             LOW ROTOR ENTRY                           91804300   000000
00101   462*   C   IP+5      0             ROTOR ORIENTATIONS TABLE                  91804310   000000
00101   463*   C   IP+6      8            SECOND MULTIMODE - MODE 8                  91804320   000000
00101   464*   C   IP+7      2             NUMBER OF PARAMETERS THIS MODE            91804330   000000
00101   465*   C   IP+8      0             LOW ROTOR ENTRY                           91804340   000000
00101   466*   C   IP+9      1             ENGINE SPEEDS TABLE                       91804350   000000
00101   467*   C   IP+10     8            THIRD MULTIMODE - MODE 8                   91804360   000000
00101   468*   C   IP+11     2             NUMBER OF PARAMETERS THIS ENTRY           91804370   000000
00101   469*   C   IP+12     0             FAN DISK 1 ENTRY                          91804380   000000
00101   470*   C   IP+13     2             BLADE ON DISK ORIENTATIONS TABLE          91804390   000000
00101   471*   C   IP+14     1            FOURTH MULTIMODE - MODE 1                  91804400   000000
00101   472*   C   IP+15     1             NUMBER OF PARAMETERS THIS ENTRY           91804410   000000
00101   473*   C   IP+16     7.53          DISK RADIUS                               91804420   000000
00101   474*   C   IP+17     8            FIFTH MULTIMODE - MODE 8                   91804430   000000
00101   475*   C   IP+18     2             NUMBER OF PARAMETERS THIS ENTRY           91804440   000000
00101   476*   C   IP+19     17            BLADE 17 ENTRY                            91804450   000000
00101   477*   C   IP+20     3             BLADE NUMBER TABLE                        91804460   000000
00101   478*   C   IP+21     8            SIXTH MULTIMODE - MODE 8                   91804470   000000
00101   479*   C   IP+22     2             NUMBER OF PARAMETERS THIS ENTRY           91804480   000000
00101   480*   C   IP+23     0             FAN DISK 1 BLADES ENTRY                   91804490   000000
00101   481*   C   IP+24     4             MECHANICAL BLADE DESCRIPTIONS TABLE       91804500   000000
00101   482*   C                          NOTE -  THIS EXAMPLE SURFACE EVALUATION    91804510   000000
00101   483*   C                           ROUTINE EXPECTS 10 GROUPS OF PARAMETERS,  91804520   000000
00101   484*   C                           AND THUS 10 MULTIMODE POINTERS.  SINCE    91804530   000000
00101   485*   C                           ONLY 6 POINTERS ARE SUPPLIED THE STANDARD91804540   000000
00101   486*   C                           LOGIC WILL ASSUME THAT PARAMETER GROUPS   91804550   000000
00101   487*   C                           7, 8, 9, AND 10 FOLLOW CONSECUTIVELY      91804560   000000
00101   488*   C                           BEHIND GROUP 6, WHICH IS THE LAST         91804570   000000
00101   489*   C                           GROUP POINTED TO.                         91804580   000000
00101   490*   C   IP+25     6            TRUNCATION REFERENCE POSITION - # PARAM    91804590   000000
00101   491*   C   IP+26     2            FIRST TRUNCATION HEADER - OUTER CASE       91804600   000000
00101   492*   C   IP+27     1.0           DO NOT FLOP SURFACE SIDE                  91804610   000000
00101   493*   C   IP+28     29            EXTERNAL ID NUMBER                        91804620   000000
00101   494*   C   IP+29     2            SECOND TRUNCATION HEADER - DISK RIM        91804630   000000
00101   495*   C   IP+30     1.0           DO NOT FLOP SURFACE SIDE                  91804640   000000
00101   496*   C   IP+31     32            EXTERNAL ID NUMBER                        91804650   000000
00101   497*   C                                                                     91804660   000000
00101   498*   C                                                                     91804670   000000
00101   499*   C   THIS CONCLUDES THE EXAMPLE                                        91804680   000000
00101   500*   C                                                                     91804690   000000
00101   501*   C                                                                     91804700   000000
00101   502*   C                                                                     91804710   000000
00101   503*   C   REQUIRED SUBROUTINES ******                                       91804720   000000
00101   504*   C                                                                     91804730   000000
00101   505*   C   003       STACK                                                   91804740   000000
00101   506*   C                          450        BVIRI                           91884750   000000
00101   507*   C   451       BDZN                                                    91884752   000000
00101   508*   C   453       IBDZN                                                   91884754   000000
00101   509*   C   455       CVIRI                                                   91884756   000000
00101   510*   C                          456        IPBDZN                          91884758   000000
00101   511*   C   917       POINTC                                                  91884760   000000
00101   512*   C   919       ADDR                                                    91884762   000000
00101   513*   C                          920        CALSER                          91884764   000000
00101   514*   C             A COMPLETE SURFACE EVALUATION ROUTINES PACKAGE          91804770   000000
00101   515*   C                                                                     91804780   000000
00101   516*   C                                                                     91804790   000000
00101   517*   C                                                                     91804800   000000
00101   518*   C   ERRORS REPORTED ******                                            91804810   000000
00101   519*   C                                                                     91804820   000000
00101   520*   C   NONE                                                              91874830   000000
00101   521*   C                                                                     91804840   000000
00101   522*   C                                                                     91804850   000000
```

Listing 8.—Continued.

```
00101       C                                                                 91804860    000000
00103             PARAMETER IVSIZ=50                                          91804870    000000
00104             COMMON /IVSIZC/IVSIZE                                       91804890    000000
00105             REAL XL(4),SV,GR(3),SF,GF(3)                                91884910    000000
00106             INTEGER ID                                                  91884920    000000
00107             INTEGER IV(IVSIZ),TRCNT                                     91804930    000000
00110             INTEGER THMDE                                               91874935    000000
00110       C                                                                 91884940    000000
00110       C     CARDS 918X4940 THROUGH 918X5030 DELETED BY V03-08.  THE FORTRAN 91884950    000000
00110       C     ADDRESS MAP IS NO LONGER NEEDED SINCE CALSED HAS TAKEN OVER     91884960    000000
00110       C     THAT FUNCTION.                                              91884970    000000
00110       C                                                                 91884980    000000
00111             DEFINE ITYPE=IBDZN(IP-5)                                    91805040    000000
00112             DEFINE ISPLC=IBDZN(IP-4)                                    91805050    000000
00113             DEFINE ITHLC=IBDZN(IP-3)                                    91805060    000000
00114             DEFINE DC=BDZN(IP-2)                                        91805070    000000
00115             DEFINE NTRUNC=IBDZN(IP-1)                                   91805080    000000
00116             DEFINE NPARAM=IBDZN(IP)                                     91805090    000000
00117             DEFINE THMODE=IBDZN(IPTH)                                   91805100    000000
00120             DEFINE THDCMD=BDZN(IPTH+1)                                  91805110    000000
00121             DEFINE THIDNM=IBDZN(IPTH+2)                                 91805120    000000
00122             DEFINE DIPBDZ(I)=IPBDZN(I)                                  91805130    000000
00123             IVSIZE=IVSIZ                                                91805140    000000
00124             DATA IVSIZE/IVSIZ/                                                      000001
00126        1001 RIDCM=1.0                       @ INTERNAL D/C MODIFIER     91805150    000001
00127        1002 INHIB=0                         @ ALLOW TRUNCATIONS         91805160    000004
00130        1003 IP=IPBDZN(ID)                   @ SDH POINTER               91805170    000004
00131             ISW=1                           @ CALC SURFACE VALUES       91805180    000010
00132             NRTN=1                          @ SWITCH FOR SURFACES       91805190    000012
00133             GO TO 1101                      @ START WITH TRUNCATIONS    91805200    000013
00134        1010 SF=SV                           @ PASS RESULT OUT           91805210    000015
00135             RETURN                          @ BACK TO CALLER            91805220    000016
00135       C                                                                 91805230    000016
00135       C     SECOND ENTRY POINT - REVERSED FIRST NEST TRUNCATIONS        91805240    000016
00135       C                                                                 91805250    000016
00135       C                                                                 91805260    000016
00136             ENTRY SURFTR (XL,ID,SF,$)                                   91885270    000021
00140        1020 RIDCM=-1.0                      @ INTERNAL D/C MODIFIER     91805280    000021
00141             GO TO 1002                      @ REST OF ENTRY IS COMMON   91805290    000023
00141       C                                                                 91805300    000023
00141       C     THIRD ENTRY - GRADIENTS                                     91805310    000023
00141       C                                                                 91805320    000023
00141       C                                                                 91805330    000023
00142             ENTRY GRAD (XL,ID,GF,$)                                     91885340    000024
00144        1030 IP=IPBDZN(ID)                   @ SDH POINTER               91805350    000024
00145             ISW=2                           @ CALC GRADIENTS            91805360    000030
00146             NRTN=3                          @ SWITCH ON GRADIENTS       91805370    000032
00147             GO TO 1201                      @ GO STRAIGHT TO CALCS      91805380    000034
00150        1035 DO 1036 IP=1,3                  @ PASS GRADIENT OUT         91805390    000042
00153        1036 GF(IP)=GR(IP)                   @                           91805400    000042
00155             RETURN                          @ BACK TO CALLER            91805410    000044
00155       C                                                                 91805420    000044
00155       C     FOURTH ENTRY - INHIBIT ALL TRUNCATIONS                      91805430    000044
00155       C                                                                 91805440    000044
00155       C                                                                 91805450    000044
00156             ENTRY SURFTI (XL,ID,SF,$)                                   91885460    000047
00160             IP=IPBDZN(ID)                   @ SDH POINTER               91805470    000047
00161             ISW=1                           @ CALC SURFACE VALUES       91805480    000053
00162             NRTN=1                          @ SWITCH FOR SURFACES       91805490    000055
00163             GO TO 1201                      @ GO DIRECT TO SURFACE      91805500    000056
00163       C+                                                                V03-08C     000056
00163       C     FIFTH ENTRY -- SURFACE VELOCITIES                           V03-08C     000056
00163       C                                                                 V03-08C     000056
00163       C     THIS ENTRY POINT INVOKES THE APPROPRIATE SURFACE EVALUTION  V03-08C     000056
00163       C     ROUTINE TO OBTAIN THE VELOCITY VECTOR FOR THE DESIGNATED SURFACE V03-08C 000056
00163       C     AT THE POINT XL.  THE VELOCITY VECTOR RESULT IS REPORTED IN GF. V03-08C 000056
00163       C                                                                 V03-08C     000056
00163       C     NOTE:                                                       V03-08C     000056
00163       C                                                                 V03-08C     000056
00163       C     THE POSITION OF A BOUNDARY IN SPACE-TIME IS INDEPENDENT OF ITS V03-08C  000056
00163       C     DEFEND/CONTAIN VALUE.  THUS, ITS VELOCITY VECTOR IS ALSO    V03-08C     000056
00163       C     INDEPENDENT OF ITS DEFEND/CONTAIN VALUE.                    V03-08C     000056
00163       C-                                                                V03-08C     000056
00164             ENTRY SURFVE (XL,ID,GF,$)                                   V03-08C     000057
00166             IP=IPBDZN(ID)                   @ POINT TO SDH              V03-08C     000057
00167             ISW=3                           @ SET SER SWITCH            V03-08C     000063
00170             NRTN=3                          @ SET INTERNAL SWITCH       V03-08C     000065
00171             GO TO 1201                      @ INTERNAL CODE SERVICE     V03-08C     000066
```

Listing 8.—Continued.

```
00171  C+                                                                   V03-08E   000066
00171  C     SIXTH ENTRY -- SURFACE ACCELERATIONS                           V03-08E   000066
00171  C                                                                    V03-08E   000066
00171  C     THIS ENTRY POINT INVOKES THE APPROPRIATE SURFACE EVALUATION    V03-08E   000066
00171  C     ROUTINE TO OBTAIN THE ACCELERATION VECTOR FOR THE DESIGNATED   V03-08E   000066
00171  C     SURFACE AT THE POINT XL.  THE ACCELERATION VECTOR RESULT IS    V03-08E   000066
00171  C     REPORTED IN GF.  SINCE ACCELERATION IS DEPENDENT ONLY ON       V03-08E   000066
00171  C     SURFACE POSITION, THE DEFEND/CONTAIN VALUE IS NOT APPLIED TO   V03-08E   000066
00171  C     THE RESULT.                                                    V03-08E   000066
00171  C-                                                                   V03-08E   000066
00172        ENTRY SURFAC (XL,ID,GF,$)          @                           V03-08E   000067
00174        IP=IPBDZN(ID)                      @ POINT TO SDH              V03-08E   000067
00175        ISW=4                              @ SET SER SWITCH            V03-08E   000073
00176        NRTN=3                             @ SET INTERNAL RETURN       V03-08E   000075
00177        GO TO 1201                         @ JOIN COMMON SERVICE       V03-08E   000077
00177  C                                                                    91805510  000077
00177  C     EVALUATION OF A SURFACE BASED ON AN SDH POINTED TO BY IP STARTS 91805520 000077
00177  C     HERE.  IT IS RE-ENTRANT AND SHOULD FOLLOW TRUNCATIONS OUT TO   91805530  000077
00177  C     A CONSIDERABLE DEPTH IF THEY ARE NOT INHIBITED BY 'INHIB'.     91805540  000077
00177  C                                                                    91805550  000077
00177  C                                                                    91805560  000077
00200   1101 IF (NRTN-2) 1111,    ,             @ SKIP SAVE ON FIRST NEST   91805570  000101
00203        CALL SPOPI (IPN)                   @ GET NEW SDH POINTER                 000104
00204        CALL SPSHI (IP)                    @ SAVE CURRENT SDH POINTER  91805590  000107
00205        CALL SPSHI (IPT)                   @  TRUNCATION POINTER       91805600  000112
00206        CALL SPSHI (IPTH)                  @  TRUNCATION HEADER POINTER 91805610 000115
00207        CALL SPSHI (MDE)                   @  ADDRESSING MODE          91805620  000120
00210        CALL SPSHI (THMDE)                 @  TRUNCATION MODE          91805630  000123
00211        CALL SPSHI (IDUM11)                @  TRUNCATION LOOP COUNTER  91805640  000126
00212        CALL SPSHI (NV)                    @  NUMBER OF MULTIMODES     91805650  000131
00213        DO 1103 IDUM11=1,TRCNT             @  APPROPRIATE ADDRESS VECTORS91805660 000141
00216        IP=IV(IDUM11)                      @                           91805670  000141
00217   1103 CALL SPSHI (IP)                    @                           91805680  000142
00221        CALL SPSHI (TRCNT)                 @  NUMBER OF TRUNCATIONS    91805690  000146
00222        CALL SPSHR (RIDCM)                 @  INTERNAL MODIFIER        91805700  000151
00223        RIDCM=1.0                          @ LOAD NEW MODIFIER         91805710  000154
00224        IP=IPN                             @ LOAD NEW SDH POINTER      91805720  000156
00225   1111 IF (INHIB)     ,     ,1201         @ BR IF TRUNCATIONS INHIBITED 91805730 000161
00230        TRCNT=NTRUNC                       @ GET NUMBER OF TRUNCATIONS 91805740  000163
00231        IF (TRCNT) 1201,1201,              @ BR IF NO TRUNCATIONS ANYWAY 91805750 000172
00234        IPT=IP+1+NPARAM                    @ POINT TO REFERENCE SPOT   91805760  000174
00235        MDE=ITHLC                          @ TRUNCATION ADDRESSING MODE 91805770 000203
00236        CALL ADDR (IPT,MDE,IV,NV,TRCNT)    @ FILL IN IV BASED ON ADDRESS 91885780 000212
00237                                           @  MODE EVALUATIONS         91885790  000221
00237        CALL SPSHI (INHIB)                 @ SAVE INHIBIT AND          91805810  000221
00240        CALL SPSHI (NRTN)                  @  RETURN SWITCH INFORMATION 91805820 000224
00241        NRTN=2                             @ SURFACES CALLS FROM TRUNC 91805830  000227
00242                                           @  HANDLER                  91805840  000231
00242        IPTH=IV(1)                         @ POINT TO FIRST TRUNC HEADER 91805850 000231
00243        IDUM11=0                           @ CLEAR LOOP COUNTER        91805860  000233
00244   1121 IDUM11=IDUM11+1                    @ DO LOOP THROUGH TRUNCATIONS 91805870 000235
00245        IF (IDUM11-TRCNT)      ,     ,1190 @ BR IF LOOP FINISHED       91805880  000237
00250        THMDE=THMODE                       @ GET BASIC TRUNCATION HEADER 91805890 000242
00251        THMDE=ABS(THMDE)                   @  MODE                     91805900  000246
00252        INHIB=0                            @ SETUP NEXT NEST LEVEL     91805910  000250
00253        IF (THMODE)     ,1127,1127         @  INHIBIT TRUNC CONTROL    91805920  000251
00256        INHIB=1                            @                           91805930  000256
00257   1127 GO TO (1128,1130,1132),THMDE       @ BR BY TRUNC HEADER MODE   91805940  000261
00260   1128 IDUM12=IPTH+8                      @ MODE 1 - SDH RIGHT HERE   91805950  000271
00261        GO TO 1135                         @                           91805960  000273
00262   1130 IDUM12=DIPBDZ(THIDNM)              @ MODE 2 - SDH DEFINED IN   91815970  000275
00263        GO TO 1135                         @  IPBDZN (EXTERNAL DEF)    91805980  000307
00264   1132 IDUM12=THIDNM+IBDZN(1)             @ MODE 3 - SDH DEFINED IN   91805990  000311
00265        IDUM12=IBDZN(IDUM12)               @  INTERNAL TABLE           91806000  000324
00266   1135 CALL SPSHI (IDUM12)                @ PUT NEW SDH POINTER ON STACK91806010 000331
00267        GO TO 1101                         @ GO EVALUATE THE SURFACE   91806020  000333
00267  C                                                                    91806030  000333
00267  C     ALL INTERNAL TRUNCATION SURFACE EVALUATIONS RETURN TO THE      91806040  000333
00267  C     FOLLOWING SEGMENT OF CODE.                                     91806050  000333
00267  C                                                                    91806060  000333
00267  C                                                                    91806070  000333
00270   1140 SV=SV*THDCMD*RIDCM                 @ DEFEND/CONTAIN MODIFIERS  91806080  000335
```

Listing 8.—Continued.

```
00271          IF (SV) 1160,    ,                 @ BR IF SURFACE TRUNCATED      91806090    000345
00274     1144 IF (ITHLC-10) 1145,1153,1145       @ BR IF MULTIMODES IN EFFECT   V03-08D     000350
00277      1145 GO TO (1146,1150,1150),THMDE      @ REPOINT IPTH BY TH MODE      91806110    000360
00300      1146 IPTH=IPTH+8                       @ MODE 1 - SKIP A STANDARD     91816120    000370
00301          IPTH=IPTH+1+IBDZN(IPTH)            @  SDH                         91806130    000372
00302          IPTH=IPTH+1+IBDZN(IPTH)            @                              91806140    000401
00303          GO TO 1121                         @  DO ANOTHER TRUNCATION       91806150    000410
00304      1150 IPTH=IPTH+3                       @ MODES 2,3 -JUST SKIP A       91806160    000412
00305                                             @  TRUNCATION HEADER           91806170    000414
00305          GO TO 1121                         @  DO ANOTHER TRUNCATION       91806180    000414
00306      1153 NV=NV-1                           @ DOWN COUNT ACTUAL MULTIMODES91806190     000416
00307                                             @  LEFT                        91806200    000421
00307          IF (NV) 1145,1145,                 @ BR IF NONE LEFT              91806210    000421
00312          IPTH=IV(IDUM11+1)                  @ MULTIMODE POINTERS FOR       91806220    000423
00313                                             @  TRUNCATION HEADERS ALWAYS   91806230    000425
00313                                             @  POINT CORRECTLY             91806240    000425
00313          GO TO 1121                         @  GO DO ANOTHER TRUNCATION    91806250    000425
00313     C                                                                      91806260    000425
00313     C    CONTROL ARRIVES HERE IF A TRUNCATION OCCURS                       91806270    000425
00313     C                                                                      91806280    000425
00313     C                                                                      91806290    000425
00314      1160 SV=1.0                            @ REPORT A SAFE VALUE          91806300    000427
00315          CALL SPOPI (NRTN)                  @ GET CALLING NEST'S SWITCHES  91806310    000430
00316          CALL SPOPI (INHIB)                 @                              91806320    000433
00317          IF (NRTN-2) 1164,1166,1164         @ ARE WE NESTED ?              V03-08D     000436
00322     1164 SF=SV                              @ NO, MAKE REPORT TO CALLER    V03-08D     000441
00323          RETURN 4                           @  VIA TRUNCATION ROUTE        V03-08D     000443
00324     1166 CALL SPOPR (RIDCM)                 @ YES, POP BACK TO PARENT      V03-08D     000450
00325          CALL SPOPI (TRCNT)                 @  SURFACE                     V03-08D     000452
00326          DO 1170 IDUM11=TRCNT,1,-1          @                              V03-08D     000455
00331          CALL SPOPI (IP)                    @                              V03-08D     000463
00332     1170 IV(IDUM11)=IP                      @                              V03-08D     000465
00334          CALL SPOPI (NV)                    @                              V03-08D     000470
00335          CALL SPOPI (IDUM11)                @                              V03-08D     000473
00336          CALL SPOPI (THMDE)                 @                              V03-08D     000476
00337          CALL SPOPI (MDE)                   @                              V03-08D     000501
00340          CALL SPOPI (IPTH)                  @                              V03-08D     000504
00341          CALL SPOPI (IPT)                   @                              V03-08D     000507
00342          CALL SPOPI (IP)                    @                              V03-08D     000512
00343          GO TO 1144                         @ FORCE NO PARENT TRUNCATION   V03-08D     000515
00343     C                                                                      91806360    000515
00343     C    CONTROL ARRIVES HERE IF TRUNCATION LOOP COMPLETES WITHOUT ANY     91806370    000515
00343     C    TRUNCATIONS OCCURING.  MUST PRECEDE STATEMENT 1201 OR HAVE A GO TO91806380    000515
00343     C                                                                      91806390    000515
00343     C                                                                      91806400    000515
00344      1190 CALL SPOPI (NRTN)                 @ LEAVING TRUNC LOOP TAKE      91806410    000517
00345          CALL SPOPI (INHIB)                 @  US BACK UP IN NEST          91806420    000521
00345     C                                                                      91806430    000521
00345     C    SURFACE VALUE ROUTINE VECTORING DONE HERE.                        91806440    000521
00345     C                                                                      91806450    000521
00345     C                                                                      91806460    000521
00346      1201 MDE=ISPLC                         @ GET PARAMETER ADDRESSING     91806470    000525
00347          CALL ADR (IP,MDE,IV,NV)            @  MODE AND FILL IV FROM IT    91886480    000533
00350          IPOS=ITYPE                         @ GET SER'S NUMBER             91876490    000541
00351          CALL CALSER (IPOS,XL,IV,NV,SV,GR,ISW)                             91886500    000550
00352                                             @ GO EVALUATE SURFACE          91876510    000561
00352     C    CARDS 918X6520 THROUGH 918X6610 DELETED BY V03-08                 91876520    000561
00352     C                                                                      91806620    000561
00352     C    SURFACE EVALUATION ROUTINES RETURN CONTROL TO THIS POINT.         91806630    000561
00352     C                                                                      91806640    000561
00352     C                                                                      91806650    000561
00352          GO TO (1218,1220,1222,1222),ISW    @ APPLY DC AS APPROPRIATE      V03-08E     000561
00353     1218 SV=SV*DC                           @ APPLY DC TO SURFACE VALUE                000573
00354          GO TO (1010,1223,1035),NRTN        @                                          000602
00355     1220 DO 1221 IDUM11=1,3                 @ APPLY DC TO GRADIENT                     000613
00360      1221 GR(IDUM11)=DC*GR(IDUM11)          @                              91826680    000620
00362     1222 GO TO (1010,1223,1035),NRTN        @ GO SOMEPLACE                 V03-08C     000626
00363      1223 CALL SPOPR (RIDCM)                @ POP THE THINGS WE            91806700    000636
00364          CALL SPOPI (TRCNT)                 @  STORED WHEN WE STARTED      91806710    000640
00365          DO 1227 IDUM11=TRCNT,1,-1          @  TO EVALUATE THIS SURFACE    91806720    000643
00370          CALL SPOPI (IP)                    @                              91806730    000651
00371      1227 IV(IDUM11)=IP                     @                              91806740    000653
00373          CALL SPOPI (NV)                    @                              91806750    000656
00374          CALL SPOPI (IDUM11)                @                              91806760    000661
00375          CALL SPOPI (THMDE)                 @                              91806770    000664
00376          CALL SPOPI (MDE)                   @                              91806780    000667
00377          CALL SPOPI (IPTH)                  @                              91806790    000672
00400          CALL SPOPI (IPT)                   @                              91806800    000675
00401          CALL SPOPI (IP)                    @                              91806810    000700
00402          GO TO 1140                         @ GO BACK TO TRUNC HANDLER     91806820    000703
00402     C    CARDS 918X6830 THROUGH 918X8999 DELETED BY V03-08                 91876830    000703
00403          END                                                               91809990    001051
END FOR
>
```

Listing 8.—Concluded.

```
 @FOR,MS  CASPER9.SA100D
FOR   4R1   E -04/13/84-15:37:21 (2,)
>@EOF

   SUBROUTINE SA100     ENTRY POINT 000151

   STORAGE USED: CODE(1) 000174; DATA(0) 000026; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   IVSIZC 000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004   BDZN
    0005   POINTC
    0006   NERR2$
    0007   SQRT
    0010   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000012 100L      0001   000056 105L      0001   000071 107L      0001   000130 114L      0001   000025 122G
    0001   000110 135G      0001   000133 143G      0004 R 000000 BDZN      0000 I 000004 I         0000   000010 INJP$
    000C I 000000 IVC       0003 I 000000 IVSIZE    0000 I 000002 NC        0000 R 000003 R


00101   SUBROUTINE SA100 (XL,IV,NV,SV,GR,ISW)                                             92110010   000000
00101  C                                                                                  92100020   000000
00101  C                                                                                  92100030   000000
00101  C      SA100      ****** A SUBROUTINE FOR CASPER ******                            92100040   000000
00101  C                 AUTHOR      WILLIAM HENRY JONES                                  92100050   000000
00101  C                 V03-00      24 JUL 78                                            92100060   000000
00101  C                 V03-01      06 SEP 78                                            92110061   000000
00101  C                 V03-01A     28 SEP 81      ADD VELOCITY HANDLING                 V03-01A    000000
00101  C                 V03-01B     05 AUG 83      ADD ACCELERATION SUPPORT              V03-01B    000000
00101  C                                                                                  90000020   000000
00101  C                                                                                  90000030   000000
00101  C      ARGUMENTS PASSED IN SUBROUTINE CALL ******                                  90010034   000000
00101  C                                                                                  90010036   000000
00101  C      ARGUMENT     TYPE        DIMENSION      DESCRIPTION                         90000040   000000
00101  C      ---------    ---------   ---------      ------------------                  90000050   000000
00101  C      XL           REAL        1 TO 4         THREE POSITION COORDINATES          90000060   000000
00101  C                                               FOLLOWED BY CURRENT TIME           90000070   000000
00101  C                                                                                  90000110   000000
00101  C      IV           INTEGER     1 TO IVSIZE    POINTER LIST TO PARAMETERS          90000120   000000
00101  C                                               IN 'BDZN'                          90000130   000000
00101  C                                                                                  90000140   000000
00101  C      NV           INTEGER     SCALAR         NUMBER OF INDEPENDENT               90000150   000000
00101  C                                               ENTRIES IN 'IV'                    90000160   000000
00101  C                                                                                  90000170   000000
00101  C      SV           REAL        SCALAR         SURFACE VALUE RETURNED              90000180   000000
00101  C                                                                                  90000190   000000
00101  C      GR           REAL        1 TO 3         GRADIENT RETURNED                   90000200   000000
00101  C                                                                                  90000210   000000
00101  C      ISW          INTEGER     SCALAR         FUNCTION SELECTOR                   90000220   000000
00101  C                                               1 => SURFACE                       90000230   000000
00101  C                                               2 => GRADIENT                      90000240   000000
00101  C                                                                                  90010262   000000
00101  C                                                                                  90010264   000000
00101  C      ARGUMENTS IN CASPER 'CACHE' MEMORY ******                                   90010266   000000
00101  C                                                                                  90010268   000000
00101  C      ARGUMENT     TYPE        DIMENSION      DESCRIPTION                         90010270   000000
00101  C      ---------    ---------   ---------      ------------------                  90010272   000000
00101  C      BDZN         REAL        1 TO BDZNSZ    SURFACE EVALUATION ROUTINE          90010274   000000
00101  C                                               PARAMETERS LIST                    90010276   000000
```

Listing 9.—Subroutine SA100.

```
00101   C                                                                      90010278  000000
00101   C                                                                      90000270  000000
00101   C                                                                      90000280  000000
00101   C                                                                      90000290  000000
00101   C     DESCRIPTION ******                                               90000300  000000
00101   C                                                                      90000310  000000
00101   C     SA100 DEFINES A TIME-INVARIANT SPHERE IN SPACE.                  92100300  000000
00101   C                                                                      90000340  000000
00101   C                                                                      90000350  000000
00101   C                                                                      90000360  000000
00101   C     PARAMETER LIST ******                                            90000370  000000
00101   C                                                                      90000380  000000
00101   C     REL POSIT   'IV' SLOT   DESCRIPTION                              90000390  000000
00101   C     ---------   ---------   ------------------                       90000400  000000
00101   C     IVP         1           X COORD OF SPHERE CENTER                 92100500  000000
00101   C     IVP+1       1           Y COORD OF SPHERE CENTER                 92100510  000000
00101   C     IVP+2       1           Z COORD OF SPHERE CENTER                 92100520  000000
00101   C     IVP+3       2           RADIUS OF SPHERE                         92100530  000000
00101   C                                                                      90000430  000000
00101   C                                                                      90000440  000000
00101   C                                                                      90000450  000000
00101   C     REQUIRED SUBROUTINES ******                                      90000460  000000
00101   C                                                                      90000470  000000
00101   C                             450         BVIRI                        90010474  000000
00101   C     451         BDZN                                                 90010476  000000
00101   C     917         POINTC                                               90000480  000000
00101   C                                                                      90000490  000000
00101   C                                                                      90000500  000000
00101   C                                                                      90000510  000000
00101   C     ERRORS REPORTED ******                                           90000520  000000
00101   C                                                                      90000530  000000
00101   C     NONE                                                             92100700  000000
00101   C                                                                      90000560  000000
00101   C                                                                      90000570  000000
00101   C                                                                      90000580  000000
00103         COMMON /IVSIZC/IVSIZE                                            90000600  000000
00104         INTEGER IV(IVSIZE),NV,ISW                                        90000610  000000
00105         REAL XL(4),SV,GR(3)                                              90010620  000000
00106         INTEGER IVC(2)                                                   92100900  000000
00107         DATA NC/2/                                                       92100910  000000
00111         DATA IVC/0,3/                                                    92100920  000000
00113         DEFINE P(I)=BDZN(I)                                              92100930  000000
00114         DEFINE A(I)=P(IV(1)+I-1)                                         92100940  000000
00115         DEFINE B=P(IV(2)+3)                                              92100950  000000
00116         GO TO (100,100,114,114),ISW      @ TRAP VEL AND ACCEL REQ        V03-01B   000000
00117    100  CALL POINTC (IV,IVC,NV,NC)       @ CORRECT PARAMETER POINTERS    V03-01A   000012
00120         R=0.0                            @ CALCULATE THE RADIUS FROM     92100970  000017
00121         DO 102 I=1,3                     @  THE CENTER OF THE SPHERE     92100980  000025
00124    102  R=R+(A(I)-XL(I))**2              @  TO THE POINT IN QUESTION     92100990  000025
00126         R=SQRT(R)                        @                               92101000  000042
00127         GO TO (105,107),ISW              @ JMP BY TASK                   92101010  000046
00130    105  SV=R-B                           @ SURFACE                       92101020  000056
00131         RETURN                           @                               92101030  000065
00132    107  R=AMAX1 (R,1.0E-19)              @ AVOID ZERO DIVIDE             92101040  000071
00133         R=1.0/R                          @ DIVIDE BY MULTIPLYING         92101050  000076
00134         DO 112 I=1,3                     @ DO THE GRADIENT               92101060  000110
00137    112  GR(I)=R*(XL(I)-A(I))             @                               92101070  000110
00141         RETURN                           @                               92101080  000124
00142    114  DO 115 I=1,3                     @ TIME INVARIANT SURFACES HAVEV03-01A     000133
00145    115  GR(I)=0.0                        @  A ZERO VELOCITY              V03-01A   000133
00147         RETURN                           @                               V03-01A   000134
00150         END                                                              92101090  000173
END FOR
>
```

Listing 9.—Concluded.

```
 @FOR,MS  CASPER1.POWERD
FOR   4R1   E -04/13/84-16:03:49 (4,)
>@EOF

   SUBROUTINE POWERC    ENTRY POINT 000247

   STORAGE USED: CODE(1) 000315; DATA(0) 000062; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   CHKLH
    0004   WERR2
    0005   PM1
    0006   VM1
    0007   P
    0010   PM2
    0011   V
    0012   VM2
    0013   STACWD
    0014   CHKTIM
    0015   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000031 105L     0001   000035 106L     0001   000040 107L     0001   000100 147G     0001   000141 160G
    0001   000143 163G     0001   000145 167G     0000 R 000025 A1     0000 R 000026 A2     0000 R 000016 A3
    0000 R 000027 B1       0000 R 000030 B2       0000 R 000017 B3     0000 R 000020 C      0000 R 000006 DM
    0000 R 000002 DMAT     0000 R 000011 EPS      0000 R 000025 F      0000 I 000021 I      0000 I 000001 IE
    0000   000042 INJP$    0000 I 000000 IS       0000 I 000022 J      0000 I 000023 K      0000 I 000024 L
    0007 R 000000 P        0005 R 000000 PM1      0010 R 000000 PM2    0000 R 000012 R      0011 R 000000 V
    0006 R 000000 VM1      0012 R 000000 VM2
00101        1*              SUBROUTINE POWERC (IL,IH,TA,TB)                             12000010    000000
00101        2*      C+                                                                  12000020    000000
00101        3*      C                                                                   12000030    000000
00101        4*      C       POWERC      ****** A SUBROUTINE FOR CASPER ******           12000040    000000
00101        5*      C                   AUTHOR      WILLIAM HENRY JONES                 12000050    000000
00101        6*      C                   V01-00      05 FEB 79                           12000060    000000
00101        7*      C                   V01-00A     07 MAR 79                           120A0061    000000
00101        8*      C                   V01-00B     11 FEB 80      ADD 'CHKTIM'                     000000
00101        9*      C                   V01-00C     23 FEB 81      REMOVE TIME FROM COMMON          000000
00101       10*      C                   V01-00D     05 APR 83      P IS DUPLICATE NAME  V01-00D     000000
00101       11*      C                                                                   12000070    000000
00101       12*      C                                                                   12000080    000000
00101       13*      C       DESCRIPTION ******                                          12000090    000000
00101       14*      C                                                                   12000100    000000
00101       15*      C       CALCULATES THE AVERAGE POWER OF COMPRESSION OVER THE PAST   12000110    000000
00101       16*      C       ITERATIONS FOR ALL AEROELEMENTS IN THE RANGE 'IL' TO 'IH'.  12000120    000000
00101       17*      C                                                                   12000130    000000
00101       18*      C       THEORY ******                                               12000140    000000
00101       19*      C                                                                   12000150    000000
00101       20*      C       THE PRESSURE AND VOLUME HISTORIES OF EACH AEROELEMENT ARE   12000160    000000
00101       21*      C       ASSUMED TO BE QUADRATIC IN TIME.                            12000170    000000
00101       22*      C                                                                   12000180    000000
00101       23*      C                   P = A1*T**2 + A2*T + A3                         12000190    000000
00101       24*      C                   V = B1*T**2 + B2*T + B3                         12000200    000000
00101       25*      C                                                                   12000210    000000
00101       26*      C       WE SET PRIMED TIME EQUAL TO ZERO AT THE TIME OF THE PREVIOUS 12000220   000000
00101       27*      C       RECORDED PRESSURES AND VOLUMES (PM1 AND VM1).  FURTHER, THE 12000230    000000
00101       28*      C       CURRENT PRESSURES AND VOLUMES ARE AT RELATIVE TIME TA AND THE           000000
00101       29*      C       SECOND PREVIOUS PRESSURES AND VOLUMES ARE AT TIME TB (WHERE             000000
00101       30*      C       TB IS USUALLY LESS THAN ZERO.  THUS,                                    000000
00101       31*      C                                                                               000000
00101       32*      C                   A3 = PM1 (AT REL T =0)                                      000000
00101       33*      C                   B3 = VM1 (AT REL T =0)                                      000000
00101       34*      C                                                                               000000
00101       35*      C       AND                                                                     000000
00101       36*      C                                                                               000000
00101       37*      C                   P  -PM1 = A1*TA**2 + A2*TA                                  000000
00101       38*      C                   PM2-PM1 = A1*TB**2 + A2*TB                                  000000
00101       39*      C                   V  -VM1 = B1*TA**2 + B2*TA                                  000000
00101       40*      C                   VM2-VM1 = B1*TB**2 + B2*TB.                                 000000
00101       41*      C                                                                               000000
00101       42*      C       BY SIMPLE MATRIX MANIPULATIONS, IT CAN BE SHOWN THAT                    000000
```

Listing 10.—Subroutine POWERC.

```
00101     C                                                                              000000
00101     C                 A1  B1    TB      -TA       P  -PM1  V  -VM1                 000000
00101     C                         =                   *                                000000
00101     C                 A2  B2   -TB**2  TA**2    PM2-PM1  VM2-VM1                   000000
00101     C                            ------------------------------                    000000
00101     C                             (TA**2)*TB - TA*TB**2                            000000
00101     C                                                                              000000
00101     C                                                                   12000350   000000
00101     C      WORK OF COMPRESSION IS THE INTEGRAL OF PRESSURE WITH RESPECT     12000360   000000
00101     C      TO VOLUME.  SINCE PRESSURE AND VOLUME ARE ASSUMED PARAMETRIC     12000370   000000
00101     C      IN TIME, WORK CAN BE CALCULATED AS THE INTEGRAL OF THE QUANTITY  12000380   000000
00101     C      PRESSURE TIMES THE DERIVATIVE WITH RESPECT TO TIME OF VOLUME     12000390   000000
00101     C      (END QUANTITY) WITH RESPECT TO TIME.  ASSUMING THE ABOVE         12000400   000000
00101     C      QUADRATIC EXPRESSIONS FOR PRESSURE AND VOLUME, THE WORK DONE     12000410   000000
00101     C      FOR THE TIME 0 TO T+ BECOMES                                     12000420   000000
00101     C                                                                       12000430   000000
00101     C      W = (A1*B1*0.5)*T+**4 + ((2*A2*B1+A1*B2)/3)*T+**3                12000440   000000
00101     C         + ((2*A3*B1+A2*B2)/2)*T+**2 + A3*B2*T+                        12000450   000000
00101     C                                                                       12000460   000000
00101     C      CASPER ROUTINES REQUIRE POWER OF COMPRESSION RATHER THAN         12000470   000000
00101     C      WORK OF COMPRESSION FIGURES FOR INTEGRATION INTO OTHER           12000480   000000
00101     C      QUANTITIES THAT ARE ONLY CALCULABLE AS POWERS.  THUS THIS        12000490   000000
00101     C      ROUTINE REPORTS THE QUANTITY W/TA.                               12000500   000000
00101     C                                                                       12000510   000000
00101     C                                                                       12000710   000000
00101     C                                                                       12000720   000000
00101     C-                                                                      12000730   000000
00103            PARAMETER OURID=120                 @                                       000000
00104            INTEGER IL,IH,IS,IE                 @                                       000000
00105            REAL DMAT(2,2),DM(3),EPS            @                                       000000
00106            REAL TA,TB,R(2,2)                   @                            V01-00D    000000
00107            REAL F(2,2),A1,B1,A2,B2,A3,B3,C     @                                       000000
00110            EQUIVALENCE (A1,F(1,1)),(A2,F(2,1)) @                                       000000
00111            EQUIVALENCE (B1,F(1,2)),(B2,F(2,2)) @                                       000000
00112            DATA IE/1/                          @                                       000000
00114            DATA EPS/1.0E-20/                   @ SINGULARITY TOLERANCE                 000000
00116            IE=1                                @ JUST IN CASE               V01-00D    000000
00117            CALL CHKLH (IL,IH,IS,OURID,IE)      @ CHECK AEROELEMENT RANGE               000001
00120            IF (IS) 102,106,102                 @ RANGE OK ?                            000010
00123     102    IF (ABS(TA-TB)-EPS) 105,103,103     @ YES, LOOK FOR POTENTIAL ZERO          000012
00126     103    IF (ABS(TA)-EPS) 105,104,104        @  DIVIDES                              000020
00131     104    IF (ABS(TB)-EPS) 105,107,107        @                                       000024
00134     105    CALL WERR2 (OURID,5)                @ OOPS                                  000031
00135     106    RETURN                              @                                       000035
00136     107    C=1.0/(TA*TB*(TA-TB))               @ INVERSE OF DETERMINANT                000040
00137            DMAT(1,1)=C*TB                      @ ESTABLISH MATRIX AS OVERHEAD          000047
00140            DMAT(2,1)=-C*TB*TB                  @  WORK FOR ALL                         000051
00141            DMAT(1,2)=-C*TA                     @                                       000053
00142            DMAT(2,2)=C*TA*TA                   @                                       000056
00143            DM(3)=0.5*TA                        @ 1/2 * TA        (W/T)                 000060
00144            DM(2)=0.666667*DM(3)*TA             @ 1/3 * TA**2                           000063
00145            DM(1)=1.5*DM(2)*TA                  @ 1/2 * TA**3                           000066
00146            DO 143 I=IL,IH,IS                   @ DO EACH AEROELEMENT                   000100
00151            A3=PM1(I)                           @ T=0 SETS A3, B3                       000100
00152            B3=VM1(I)                           @                                       000104
00153            R(1,1)=P(I)-A3                      @ CALC DELTA-P AND DELTA-V     V01-00D  000110
00154            R(2,1)=PM2(I)-A3                    @  RELATIVE TO T=0 P AND V     V01-00D  000115
00155            R(1,2)=V(I)-B3                      @                              V01-00D  000122
00156            R(2,2)=VM2(I)-B3                    @                              V01-00D  000127
00157            DO 137 J=1,2                        @ SIMPLE MATRIX MULTIPLY GETS           000143
00162            DO 137 K=1,2                        @  A1, A2, B1, AND B2                   000143
00165            F(J,K)=0.0                          @                                       000143
00166            DO 136 L=1,2                        @                                       000145
00171     136    F(J,K)=F(J,K)+DMAT(J,L)*R(L,K)      @                              V01-00D  000145
00173     137    CONTINUE                            @                                       000163
00176            C=(A1*B1*DM(1))+((2.0*A2*B1+A1*B2)*DM(2))+((2.0*A3*B1+A2*B2)*DM(3)          000163
00176           1)+(A3*B2)                           @ CALC POWER OF COMPRESSION             000163
00177            CALL STACWD (I,C)                   @ STORE RESULT                          000212
00200            CALL CHKTIM (IL,IH,I)               @ KEEP AN EYE ON THE TIME               000216
00201     143    CONTINUE                            @                                       000225
00203            RETURN                              @                                       000225
00204            END                                                              12001010   000314
END FOR
>
```

Listing 10.—Concluded.

```
@FOR,MS  CASPER5.WRKFDD
FOR   4R1   E -04/13/84-15:48:49 (1,)
>@EOF

   SUBROUTINE WRKFLD     ENTRY POINT 000125


   STORAGE USED: CODE(1) 000145; DATA(0) 000024; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   MU
    0004   MUTHET
    0005   CHKLH
    0006   CHKTIM
    0007   U
    0010   P
    0011   EIJ
    0012   STS
    0013   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000111 110L      0001    000017 112G      0001    000027 116G      0001    000051 122G      0011 R 000000 EIJ
    0000 I 000001 I          0000    000010 INJP$     0000 I 000000 IS         0000 I 000002 J          0000 I 000004 K
    0003 R 000000 MU         0004 R 000000 MUTHET     0010 R 000000 P          0000 R 000003 R          0007 R 000000 U


00101          SUBROUTINE WRKFLD (IL,IH)                                            000000
00101   C+                                                                          000000
00101   C                                                                           000000
00101   C      WRKFLD      ****** A SUBROUTINE FOR CASPER ******                    000000
00101   C                  AUTHOR      WILLIAM HENRY JONES                          000000
00101   C                  V01-00      12 FEB 80                                    000000
00101   C                  V01-00A     15 SEP 80      FUNCTION TYPE STATEMENTS      000000
00101   C                                                                           000000
00101   C                                                                           000000
00101   C      DESCRIPTION ******                                                   000000
00101   C                                                                           000000
00101   C      THIS ROUTINE IS PART OF A PATCH TO THE POWER OF DISTORTION           000000
00101   C      SECTION OF CASPER WORK FLOW CALCULATION.  IT LOADS INTO THE          000000
00101   C      SCRATCH STRING OF EACH ELEMENT IN THE RANGE 'IL' TO 'IH' THE         000000
00101   C      QUANTITY                                                             000000
00101   C                                                                           000000
00101   C      S(I) = U(J)*(-P*D(I,J)+MUTHET*D(I,J)+2*MU*E(I,J)), I=1,3, J=1,3      000000
00101   C                                                                           000000
00101   C      PER THE DERIVATION IN THE CASPER THEORETICAL REPORT.                 000000
00101   C                                                                           000000
00101   C                                                                           000000
00101   C-                                                                          000000
00103          PARAMETER OURID=504                                                  000000
00104          REAL MU,MUTHET                                                       000000
00105          CALL CHKLH (IL,IH,IS,OURID,1)       @ GO VALIDATE DO LOOP RANGE      000000
00106          IF (IS) 102,110,102                 @ IS IT VALID ?                  000006
00111   102    DO 109 I=IL,IH,IS                   @ YES, DO IT                     000010
00114          CALL CHKTIM (IL,IH,I)               @ KEEP AN EYE ON THE TIME        000017
00115          DO 108 J=1,3                        @                                000027
00120          R=U(I,J)*(MUTHET(I)-P(I))           @ DO KRONECKER DELTA TERM        000027
00121          DO 107 K=1,3                        @                                000051
00124   107    R=R+2.0*U(I,K)*MU(I,1)*EIJ(I,J,K)   @ ACCUMULATE SHEAR TERM          000051
00126   108    CALL STS (I,J,R)                    @ STORE RESULT                   000077
00130   109    CONTINUE                            @                                000111
00132   110    RETURN                              @                                000111
00133          END                                 @                                000144
END FOR.
>
```

Listing 11.—Subroutine WRKFLD.

```
@FOR,MS  CASPER5.WRKFED
FOR  4R1   E -04/13/84-15:50:03 (0,)
>@EOF

   SUBROUTINE WRKFLE    ENTRY POINT 000153

   STORAGE USED: CODE(1) 000174; DATA(0) 000046; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   CHKLH
    0004   CHKTIM
    0005   NEI
    0006   S
    0007   INTP
    0010   V
    0011   STS
    0012   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000017 112G      0001   000030 116G      0001   000136 118L      0001   000040 122G      0001   000054 130G
    0001   000056 133G      0001   000074 142G      0001   000077 145G      0000 R 000000 B         0000 R 000024 D
    0000 I 000021 I         0000   000032 INJP$     0007 I 000000 INTP      0000 I 000017 IS        0000 I 000020 I1
    0000 I 000022 J         0000 I 000023 K         0005 I 000000 NEI       0006 R 000000 S         0010 R 000000 V


00101         SUBROUTINE WRKFLE (IL,IH)                                     000000
00101    C+                                                                 000000
00101    C                                                                  000000
00101    C    WRKFLE     ****** A SUBROUTINE FOR CASPER ******               000000
00101    C               AUTHOR      WILLIAM HENRY JONES                    000000
00101    C               V01-00      12 FEB 80                              000000
00101    C                                                                  000000
00101    C                                                                  000000
00101    C    DESCRIPTION ******                                            000000
00101    C                                                                  000000
00101    C    THIS ROUTINE COMPLETES THE POWER OF DISTORTION PATCH BY       000000
00101    C    CALCULATING THE DIVERGENCE OF THE VECTORS CALCULATED BY       000000
00101    C    WRKFLD.                                                       000000
00101    C                                                                  000000
00101    C                                                                  000000
00101    C-                                                                 000000
00103         PARAMETER OURID=505             @                             000000
00104         REAL B(3,5)                     @                             000000
00105         CALL CHKLH (IL,IH,IS,OURID,1)   @ CHECK RANGE LIMITS          000000
00106         IF (IS) 102,118,102             @ VALID RANGE ?               000006
00111    102  DO 117 I1=IL,IH,IS              @ YES, USE IT                 000010
00114         CALL CHKTIM (IL,IH,I1)          @ KEEP AN EYE ON THE TIME     000017
00115         DO 107 I=1,5                    @ PICKUP VALUES FROM THE      000030
00120         J=NEI(I1,I)                     @  NEIGHBORS                  000030
00121         DO 107 K=1,3                    @                             000040
00124    107  B(K,I)=S(J,K)                   @                             000040
00127         DO 110 I=1,5                    @ FORM DIFFERENCES            000056
00132         DO 110 J=1,3                    @                             000056
00135    110  B(J,I)=B(J,I)-S(I1,J)           @                             000056
00140         D=0.0                           @ INIT ACCUMULATOR            000067
00141         DO 114 I=1,3                    @ ACCUMULATE DIVERGENCE       000077
00144         DO 114 J=1,5                    @  (WORK PER UNIT VOLUME)     000077
00147    114  D=D+INTP(I1,I,J)*B(I,J)         @                             000077
00152         D=D*V(I1)                       @ WORK DONE ON THE ELEMENT    000121
00153         CALL STS (I1,4,D)               @ SAVE IN SCRATCH SLOT 4      000126
00154    117  CONTINUE                        @                             000136
00156    118  RETURN                          @                             000136
00157         END                             @                             000173
END FOR
```

Listing 12.—Subroutine WRKFLE.

```
@FOR,MS  CASPER5.WRKFFD
FOR   4R1   E -04/13/84-15:51:27 (1,)
>@EOF

   SUBROUTINE WRKFLF    ENTRY POINT 000220


   STORAGE USED: CODE(1) 000243; DATA(0) 000043; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   CHKLH
    0004   ASCH
    0005   T
    0006   V
    0007   X
    0010   NEI
    0011   STS
    0012   CHKTIM
    0013   XPRR
    0014   SQRT
    0015   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000017 112G      0001   000047 121G      0001   000103 125L      0001   000061 126G      0001   000161 131L
    0001   000071 132G      0001   000203 136L      0001   000106 144G      0004 R 000000 ASCH     0000 R 000000 B
    0000 R 000001 C         0000 R 000002 D         0000 R 000003 E         0000 I 000013 I        0000   000025 INJP$
    0000 I 000011 IS        0000 I 000012 I1        0000 I 000014 J         0000 I 000015 K        0010 I 000000 NEI
    0000 R 000004 R         0000 R 000005 S         0005 R 000000 T         0006 R 000000 V        0007 R 000000 X
    0000 R 000006 X1
00101        1*          SUBROUTINE WRKFLF (IL,IH,GDAST)                                           000000
00101        2*      C+                                                                            000000
00101        3*      C                                                                             000000
00101        4*      C   WRKFLF       ****** A SUBROUTINE FOR CASPER ******                        000000
00101        5*      C                AUTHOR       WILLIAM HENRY JONES                             000000
00101        6*      C                X01.00       12 FEB 80                                       000000
00101        7*      C                X01.00A      06 OCT 81       DISTANCE INTO HEAT FLOW   X01.00A  000000
00101        8*      C                                                                             000000
00101        9*      C                                                                             000000
00101       10*      C   DESCRIPTION ******                                                        000000
00101       11*      C                                                                             000000
00101       12*      C   THIS ROUTINE CALCULATES THE HEAT FLOW BETWEEN NEAREST                     000000
00101       13*      C   NEIGHBORS.  IT IS CALCULATED AS THE AVERAGE COEFFICIENT                   000000
00101       14*      C   OF HEAT TRANSFER BETWEEN THE NEIGHBORS TIMES THE DIFFERENCE               000000
00101       15*      C   IN TEMPERATURE TIMES THE TIME DURATION OF HEAT FLOW TIMES THE   X01.00A   000000
00101       16*      C   AVERAGE CROSS-SECTIONAL AREA OF THE HEAT-EXCHANGING ELEMENTS    X01.00A   000000
00101       17*      C   AND DIVIDED BY THE DISTANCE BETWEEN THE ELEMENTS.               X01.00A   000000
00101       18*      C                                                                             000000
00101       19*      C   CALCULATIONS ARE RECORDED ONLY FOR TARGET AEROELEMENTS                    000000
00101       20*      C   IN THE RANGE 'IL' TO 'IH'.  NEIGHBORS MUST RECIPROCATE                    000000
00101       21*      C   (THE NEIGHBOR MUST LIST THE TARGET AEROELEMENT AS A                       000000
00101       22*      C   NEIGHBOR) FOR THE CALCULATIONS BETWEEN THE PAIRING TO                     000000
00101       23*      C   BE CARRIED OUT.                                                           000000
00101       24*      C                                                                             000000
00101       25*      C                                                                             000000
00101       26*      C-                                                                            000000
00103       27*          PARAMETER OURID=506                                                       000000
00104       28*          REAL B,C,D,E,R,S,X1(3)           @                              X01.00A   000000
00105       29*          CALL CHKLH (IL,IH,IS,OURID,1)    @ CHECK RANGE LIMITS                     000000
00106       30*          IF (IS) 102,136,102              @ VALID RANGE ?                X01.00A   000006
00111       31*      102 DO 135 I1=IL,IH,IS               @ YES, USE IT                  X01.00A   000010
00114       32*          B=0.0                            @ CLEAR ACCUMULATOR            X01.00A   000017
00115       33*          C=ASCH(I1)                       @ TARGET COEFFICIENT,          X01.00A   000020
00116       34*          D=T(I1)                          @  TEMPERATURE,                X01.00A   000024
```

Listing 13.—Subroutine WRKFLF.

```
00117   35*        E=3.141592653*(ABS(7.9577471E-02*V(I1))**0.6666667) @  AREA,       X01.00A    000030
00120   36*        DO 108 I=1,3                          @  POSITION                  X01.00A    000047
00123   37*   108  X1(I)=X(I1,I)                         @                            X01.00A    000047
00125   38*        DO 131 I=1,5                          @ DO EACH NEAREST NEIGHBOR   X01.00A    000061
00130   39*        J=NEI(I1,I)                           @ GET NEIGHBOR ID            X01.00A    000061
00131   40*        DO 113 K=1,5                          @                            X01.00A    000071
00134   41*        IF (NEI(J,K)-I1) 113,125,113          @ DOES NEIGHBOR RECIPROCATE ? X01.00A   000071
00137   42*   113  CONTINUE                              @ NO, NOT YET ANYWAY         X01.00A    000101
00141   43*        GO TO 131                             @ NO, NOT AT ALL             X01.00A    000101
00142   44*   125  R=0.0                                 @ CALC DISTANCE BETWEEN      X01.00A    000103
00143   45*        DO 127 K=1,3                          @  NEIGHBORS                 X01.00A    000106
00146   46*   127  R=R+(X(J,K)-X1(K))**2                 @                            X01.00A    000106
00150   47*        R=SQRT(R)                             @                            X01.00A    000120
00151   48*        S=3.141592653*(ABS(7.9577471E-02*V(J))**0.6666667) @ CALC AREA     X01.00A    000124
00152   49*        B=B+((ASCH(J)+C)*(T(J)-D)*(S+E))/R    @ UN-NORMALIZED HEAT EXCHANGE X01.00A   000140
00153   50*   131  CONTINUE                              @                            X01.00A    000162
00155   51*        B=B*0.25*GDAST                        @ NORMALIZE HEAT EXCHANGE    X01.00A    000162
00156   52*        CALL STS (I1,5,B)                     @ SAVE IN SCRATCH AREA       X01.00A    000166
00157   53*        CALL CHKTIM (IL,IH,I1)                @ KEEP AN EYE ON THE TIME    X01.00A    000173
00160   54*   135  CONTINUE                              @                            X01.00A    000203
00162   55*   136  RETURN                                @                            X01.00A    000203
00163   56*        END                                   @                                       000242
END FOR
>
```

Listing 13.—Concluded.

```
 @FOR,MS  CASPER5.WRKFGD
FOR   4R1   E -04/13/84-15:53:01 (1,)
>@EOF


   SUBROUTINE WRKFLG    ENTRY POINT 000106


   STORAGE USED: CODE(1) 000125; DATA(0) 000020; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   STAT
    0004   CHKLH
    0005   CHKTIM
    0006   S
    0007   ACWDT
    0010   ASCV
    0011   GDACT
    0012   STT
    0013   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000054 107L      0001   000075 110L      0001   000017 113G      0007 R 000000 ACWDT      0010 R 000000 ASCV
    0000 R 000000 B         0011 R 000000 GDACT     0000   000007 INJP$     0000 I 000001 IS         0000 I 000002 I1
    0006 R 000000 S         0003 I 000000 STAT



00101         SUBROUTINE WRKFLG (IL,IH,GDAST)                                              000000
00101    C+                                                                                000000
00101    C                                                                                 000000
00101    C    WRKFLG      ****** A SUBROUTINE FOR CASPER ******                            000000
00101    C                AUTHOR      WILLIAM HENRY JONES                                  000000
00101    C                V01-00      12 FEB 80                                            000000
00101    C                V01-00A     15 SEP 80       FUNCTION TYPE STATEMENT              000000
00101    C                                                                                 000000
00101    C                                                                                 000000
00101    C    DESCRIPTION ******                                                           000000
00101    C                                                                                 000000
00101    C    THIS ROUTINE REVISES THE CURRENT TEMPERATURE OF EACH ELEMENT                 000000
00101    C    IN THE RANGE 'IL' TO 'IH' BASED ON THE FOLLOWING INFORMATION:                000000
00101    C    1) THE CURRENT TEMPERATURE COPY IN GDACT, 2) THE POWER OF                    000000
00101    C    COMPRESSION IN ACWDT, 3) THE VISCOUS DISTORTION WORK IN                      000000
00101    C    S(4), AND 4) THE HEAT TRANSFERED BY CONDUCTION IN S(5).                      000000
00101    C                                                                                 000000
00101    C                                                                                 000000
00101    C-                                                                                000000
00103         PARAMETER OURID=507                                                          000000
00104         INTEGER STAT                                                                 000000
00105         REAL B                                                                       000000
00106         CALL CHKLH (IL,IH,IS,OURID,1)       @ CHECK RANGE LIMITS                     000000
00107         IF (IS) 102,110,102                 @ VALID RANGE ?                          000006
00112    102  DO 109 I1=IL,IH,IS                  @ YES, USE IT                            000010
00115         CALL CHKTIM (IL,IH,I1)              @ KEEP AN EYE ON THE TIME                000017
00116         B=S(I1,4)+S(I1,5)                   @ DISTORTION + CONDUCTION                000024
00117         IF (AND(STAT(I1),2**2)) 107,106,107 @ IS THERE A COMPRESSION TERM?           000037
00122    106  B=B+GDAST*ACWDT(I1)                 @ YES, CONVERT IT TO WORK                000045
00123    107  B=(B/ASCV(I1))+GDACT(I1)            @ CONVERT TO TEMPERATURE                 000054
00124         CALL STT (I1,B)                     @ REVISE ELEMENT'S TEMP                  000066
00125    109  CONTINUE                            @                                        000075
00127    110  RETURN                              @                                        000075
00130         END                                 @                                        000124
END FOR
>
```

Listing 14.—Subroutine WRKFLG.

```
 @FOR,MS  CASPER1.VOLD
FOR   4R1   E -04/17/84-12:49:01 (5,)
>@EOF

   SUBROUTINE VOL       ENTRY POINT 001017

   STORAGE USED: CODE(1) 001040; DATA(0) 000145; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   ISIZEC 000001

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004   STAT
    0005   CHKLH
    0006   SPSHI
    0007   PAAI
    0010   SPSHR
    0011   REQSAF
    0012   SPOPI
    0013   ERROR2
    0014   X
    0015   NEI
    0016   NORMM
    0017   CROSS
    0020   NORM
    0021   DOT
    0022   CROSSM
    0023   ABSARC
    0024   STV
    0025   CHKTIM
    0026   NERR2$
    0027   ACOS
    0030   TAN
    0031   SIN
    0032   SQRT
    0033   COS
    0034   ASIN
    0035   EXP
    0036   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000110 101L     0001   000250 118L     0001   000307 126L     0001   000021 15L      0001   000112 154G
    0001   000125 161G     0001   000135 165G     0001   000030 17L      0001   000153 174G     0001   000175 203G
    0001   000217 211G     0001   000053 22L      0001   000263 226G     0001   000267 232G     0001   000322 245G
    0001   000331 254G     0001   000526 255L     0001   000342 260G     0001   000550 260L     0001   000352 267G
    0001   000073 27L      0001   000572 275L     0001   000716 300L     0001   000721 600L     0001   000771 707L
    0000 R 000104 A        0000 R 000103 ARNM     0000 R 000102 AROM     0000 R 000101 B        0000 R 000100 C
    0000 R 000070 CO       0000 R 000061 CURAV    0000 I 000063 I        0000 I 000056 ID       0000 I 000057 IE
    0000 I 000074 IJK      0000   000131 INJP$    0000 R 000001 IPAV     0000 I 000055 IS       0000 I 000050 ISEN
    0003   000000 ISIZE    0000 I 000060 ISW      0000 I 000062 I1       0000 I 000064 J        0000 I 000065 K
    0000 I 000076 M        0000 I 000000 MIN      0000 I 000075 N        0015 I 000000 NEI      0000 R 000073 OM
    0000 R 000005 P        0000 R 000077 PHI1     0000 R 000067 R        0000 R 000066 RAD      0000 R 000105 RRR
    0000 R 000071 SI       0004 I 000000 STAT     0000 R 000072 TH       0000 R 000043 THETA    0014 R 000000 X
    0000 R 000002 XL       0000 R 000024 Z


00101      1*           SUBROUTINE VOL (IL,IH,PAV,ACAV)                               115A0010    000000
00101      2*     C                                                                   11500020    000000
00101      3*     C                                                                   11500030    000000
00101      4*     C     VOL         ****** A SUBROUTINE FOR CASPER ******             11500040    000000
00101      5*     C                 AUTHOR      WILLIAM HENRY JONES                   11500050    000000
00101      6*     C                 V01-00      06 APR 78                             11500060    000000
00101      7*     C                 V01-01      30 AUG 78                             11510061    000000
```

Listing 15.—Subroutine VOL.

```
00101  C              V01-01A    06 FEB 79                                115A0062   000000
00101  C              V01-01B    08 FEB 80    PAX PROTOCOL ADJUSTMENTS               000000
00101  C              V01-01C    15 SEP 80    FUNCTION TYPE STATEMENT                000000
00101  C              V01-01D    20 APR 83    LITTLE FIXES               V01-01D     000000
00101  C              V01-01E    29 APR 83    LITTLE TYPO                V01-01E     000000
00101  C                                                                 11500070   000000
00101  C   ARGUMENTS IN CASPER 'CACHE' MEMORY ******                     11510074   000000
00101  C                                                                 11510076   000000
00101  C   ARGUMENT   TYPE       DIMENSION    DESCRIPTION                11500080   000000
00101  C   ---------  ---------  ---------    ------------------         11510090   000000
00101  C   X          REAL       1 TO ISIZE   AEROELEMENT POSITIONS      11500100   000000
00101  C                         1 TO 3                                  11500110   000000
00101  C                                                                 11500120   000000
00101  C   NEI        INTEGER    1 TO ISIZE   NEAREST NEIGHBOR LIST      11500130   000000
00101  C                         1 TO 5                                  11500140   000000
00101  C                                                                 11500150   000000
00101  C   V          REAL       1 TO ISIZE   AEROELEMENT VOLUMES        11510160   000000
00101  C                                       -- THESE WILL BE REPLACED 11500170   000000
00101  C                                       WITH ADJUSTED VOLUME      11500180   000000
00101  C                                       ESTIMATES                 11500190   000000
00101  C   STAT       INTEGER    1 TO ISIZE   AEROELEMENT STATUS LIST    11510192   000000
00101  C                                                                 11510194   000000
00101  C                                                                 11500200   000000
00101  C                                                                 11510202   000000
00101  C   ARGUMENTS PASSED IN SUBROUTINE CALL ******                    11510203   000000
00101  C                                                                 11510204   000000
00101  C   ARGUMENT   TYPE       DIMENSION    DESCRIPTION                11510205   000000
00101  C   ---------  ---------  ---------    ------------------         11510206   000000
00101  C   IL         INTEGER    SCALAR       LOW AEROELEMENT ID LIMIT   115A0210   000000
00101  C                                                                 115A0212   000000
00101  C   IH         INTEGER    SCALAR       HIGH AEROELEMENT ID LIMIT  115A0214   000000
00101  C                                                                 115A0216   000000
00101  C   PAV        REAL       SCALAR       PREVIOUS AEROELEMENT SOLID 115A0218   000000
00101  C                                       ANGLE AVERAGE             115A0220   000000
00101  C                                                                 115A0222   000000
00101  C   ACAV       REAL       SCALAR       ACCUMULATION SLOT FOR ALL  115A0224   000000
00101  C                                       SOLID ANGLES CALCULATED   115A0226   000000
00101  C                                       BY THE EXECUTION OF THIS  115A0228   000000
00101  C                                       ROUTINE                   115A0230   000000
00101  C                                                                 11500250   000000
00101  C                                                                 11500260   000000
00101  C                                                                 11500270   000000
00101  C   DESCRIPTION                                                   11500280   000000
00101  C                                                                 11500290   000000
00101  C   THIS ROUTINE CALCULATES NEW ESTIMATED VOLUMES FOR ALL AEROELEMENTS11500300   000000
00101  C   ACCORDING TO FIXED RULES.  THE VOLUME ESTIMATES WILL NOT (IN ALL  11500310   000000
00101  C   PROBABILITY) ADD UP TO EQUAL THE TOTAL VOLUME OF THE PROBLEM AND  11500320   000000
00101  C   MUST BE ADJUSTED FOR THIS OUTSIDE THIS ROUTINE.               11500330   000000
00101  C                                                                 11500340   000000
00101  C   THE VOLUMES CALCULATED BY THIS ROUTINE ARE BASED ON A SPHERE  11500350   000000
00101  C   WHOSE RADIUS IS EQUAL TO THE AVERAGE DISTANCE TO THE AEROELEMENT'S11500360   000000
00101  C   FIVE NEAREST NEIGHBORS.  THIS ESTIMATE IS ADJUSTED BY THE SOLID  11500370   000000
00101  C   ANGLE SUBTENDED BY THE FIVE NEAREST NEIGHBORS.  THE ADJUSTMENT   11500380   000000
00101  C   GIVES MORE VOLUME TO AEROELEMENTS WHOSE SOLID ANGLE RESULTS ARE  11500390   000000
00101  C   LESS THAN AVERAGE.                                            11500400   000000
00101  C                                                                 11510401   000000
00101  C   IF A PARTICULAR AEROELEMENT IS IDENTIFIED AS A 'BOUNDARY'     11510402   000000
00101  C   AEROELEMENT (BIT #11 OF STATUS WORD SET) THE FOLLOWING        11510403   000000
00101  C   ADJUSTMENTS ARE MADE.  ITS SOLID ANGLE RESULT WILL AUTOMATICALLY 11510404   000000
00101  C   BE DOUBLED (SINCE IT HAS ONLY HALF A HORIZON FOR NEIGHBORS)   11510405   000000
00101  C   AND ITS VOLUME ESTIMATE IS HALVED (SINCE IT HAS ONLY HALF A   11510406   000000
00101  C   SPHERE TO RESIDE IN.  THIS PUTS BOUNDARY ELEMENTS ON EQUAL    11510407   000000
00101  C   FOOTING WITH NORMAL ELEMENTS PROVIDED THE BOUNDARY IS QUASI-  11510408   000000
00101  C   FLAT AT THE BOUNDARY ELEMENT SITE WHEN VIEWED IN THE SCALE    11510409   000000
00101  C   OF THE BOUNDARY ELEMENT.                                      11510410   000000
00101  C                                                                 11510411   000000
00101  C   DURING THE EXECUTION OF THE SUBROUTINE THE SUM OF ALL SOLID   115A0420   000000
00101  C   ANGLES CALCULATED (FOR AEROELEMENTS IN THE RANGE 'IL' TO 'IH') 115A0425   000000
00101  C   IS ACCUMULATED.  ON EXIT THE SUM IS LOADED INTO SLOT 'ACAV'   115A0430   000000
00101  C   TO AID THE CALLER IN MAKING ANY NECESSARY ADJUSTMENTS IN 'PAV'. 115A0435   000000
00101  C                                                                 11500450   000000
00101  C                                                                 11500460   000000
```

Listing 15.—Continued.

```
00101   C                                                                       11500470   000000
00101   C     THE CALCULATION OF SOLID ANGLE                                    V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     THE SOLID ANGLE SUBTENDED BY THE FIVE NEAREST NEIGHBORS IS        V01-01D    000000
00101   C     CALCULATED BY SUMMING THE CONTRIBUTIONS OF SPHERICAL TRIANGLES.   V01-01D    000000
00101   C     THREE SPHERICAL TRIANGLES ARE IDENTIFIED: ONE SUBTENDED BY        V01-01D    000000
00101   C     NEIGHBORS 1, 2, AND 3, ONE SUBTENDED BY NEIGHBORS 1, 3, AND       V01-01D    000000
00101   C     4, AND ONE SUBTENDED BY NEIGHBORS 1, 4, AND 5.  THE SPHERICAL     V01-01D    000000
00101   C     TRIANGLES ARE PROJECTED ONTO A SPHERE OF UNIT RADIUS AND ARE      V01-01D    000000
00101   C     IDENTIFIED THROUGH A NEAREST NEIGHBOR LIST THAT HAS BEEN RE-      V01-01D    000000
00101   C     ORDERED TO MAKE A MONOTONICALLY INCREASING ORIENTATION OF NEIGH-  V01-01D    000000
00101   C     BORS WHEN VIEWED FROM A POLE ESTABLISHED BY THE FIRST NEIGHBOR.   V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     THE AREA OF EACH SPHERICAL TRIANGLE IS ESTABLISHED BY DECOM-      V01-01D    000000
00101   C     POSING IT INTO TWO SPHERICAL RIGHT TRIANGLES.  IT CAN BE SHOWN    V01-01D    000000
00101   C     THAT THE AREA OF A SPHERICAL RIGHT TRIANGLE OF BASE THETA.1 AND   V01-01D    000000
00101   C     HEIGHT PHI.1 IS THE INTEGRAL                                      V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C               THETA.1         A*SIN(THETA)                            V01-01D    000000
00101   C       AREA = I         --------------------------    D.THETA          V01-01D    000000
00101   C               0        SQRT(1+(A*SIN(THETA))**2))                     V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                   TAN(PHI.1)                                          V01-01D    000000
00101   C       WHERE A =  ------------                                         V01-01D    000000
00101   C                   SIN(THETA.1)                                        V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     THIS CAN BE INTEGRATED BY THE SUBSTITUTION                        V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C       X = B * COS(THETA)                                              V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C       WHERE B = SQRT((A**2)/(1+(A**2)))                               V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     THIS INTEGRATION GIVES THE RESULT                                 V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C       AREA = ASIN(B) - ASIN(B*COS(THETA.1))                           V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     AN ANOMALLY OF COMPUTATION EXISTS AS PHI.1 APPROACHES PI/2.       V01-01D    000000
00101   C     IN THIS EVENT, B IS SET TO THE LIMIT OF 1.0.  ALTHOUGH A TENDS    V01-01D    000000
00101   C     TO INFINITY AS THETA.1 TENDS TO PI, A COMPUTATIONAL ANOMALLY IS   V01-01D    000000
00101   C     AVOIDED BY CALCULATING B AS                                       V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                             TAN(PHI.1)**2                             V01-01D    000000
00101   C       B = SQRT ( ----------------------------------- )                V01-01D    000000
00101   C                  (SIN(THETA.1)**2) + (TAN(PHI.1)**2)                  V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     THE CALCULATION OF A SPHERICAL TRIANGLE PROCEEDS IN THE FOLLOWING V01-01D    000000
00101   C     MANNER.                                                           V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     1.   A VECTOR TO THE FIRST NEIGHBOR, V.N1, IS ESTABLISHED.  THIS  V01-01D    000000
00101   C          VECTOR DEFINES AN ARBITRARY X' AXIS AND THE LOCATION OF AN   V01-01D    000000
00101   C          ARBITRARY 0 LONGITUDE, 0 LATITUDE POINT ON THE UNIT SPHERE.  V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     2.   A VECTOR TO NEIGHBOR N (N=2, 3, 4), V.NN, IS ESTABLISHED.    V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     3.   A NORTH POLE (Z' AXIS) IS ESTABLISHED AS                     V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                             V.N1 X V.NN                               V01-01D    000000
00101   C                 V.NP = --------------------                           V01-01D    000000
00101   C                         MAG ( V.N1 X V.NN )                           V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     4.   THE EAST LONGITUDE OF NEIGHBOR N IS ESTABLISHED AS           V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C                 THETA.N = ACOS ( V.N1 * V.NN )                        V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     5.   A VECTOR TO NEIGHBOR M (M=N+1), V.NM, IS ESTABLISHED.        V01-01D    000000
00101   C                                                                       V01-01D    000000
00101   C     6.   THE HEIGHT OF THE SPHERICAL TRIANGLES IS ESTABLISHED AS      V01-01D    000000
```

Listing 15.—Continued.

```
00101  C                                                                   V01-01D   000000
00101  C           PSI.1 = ACOS ( V.NP * V.NM )                            V01-01D   000000
00101  C           PHI.1 = ABS ( PI/2 - PSI.1 )                            V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    7.  THE VECTOR TO NEIGHBOR M IS PROJECTED INTO THE EQUATORIAL  V01-01D   000000
00101  C        (X-Y) PLANE AS                                             V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           V.EM = V.NP X ( V.NM X V.NP )                           V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C        AND IS NORMALIZED.                                         V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    8.  THE LENGTH OF A RIGHT SPHERICAL TRIANGLE (WHOSE BASE IS THE V01-01D   000000
00101  C        EQUATOR) RUNNING FROM 0 LONGITUDE TO NEIGHBOR M IS FOUND AS V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           THETA.M = ACOS ( V.N1 * V.EM )                          V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    9.  THE LENGTH OF A RIGHT SPHERICAL TRIANGLE (WHOSE BASE IS THE V01-01D   000000
00101  C        EQUATOR) RUNNING FROM NEIGHBOR N TO NEIGHBOR M IS FOUND AS V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           THETA.NM = ACOS ( V.EM * V.NN )                         V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    10. THE SENSE OF THE ROTATIONS FROM 0 LONGITUDE TO NEIGHBOR N, V01-01D   000000
00101  C        FROM 0 LONGITUDE TO NEIGHBOR M, AND FROM NEIGHBOR N TO     V01-01D   000000
00101  C        NEIGHBOR M IS ESTABLISHED AS                               V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           SENSE.ON = SIGN ( V.NP * ( V.N1 X V.NN ) )              V01-01D   000000
00101  C           SENSE.OM = SIGN ( V.NP * ( V.N1 X V.EM ) )              V01-01D   000000
00101  C           SENSE.NM = SIGN ( V.NP * ( V.NN X V.EM ) )              V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    11. THE AREA (SOLID ANGLE) OF TWO SPHERICAL RIGHT TRIANGLES IS V01-01D   000000
00101  C        CALCULATED USING THE FORMULATION PREVIOUSLY DESCRIBED.  THE V01-01D   000000
00101  C        FIRST TRIANGLE RUNS FROM 0 LONGITUDE TO NEIGHBOR M AND THE V01-01D   000000
00101  C        SECOND TRIANGLE RUNS FROM NEIGHBOR N TO NEIGHBOR M.  THE   V01-01D   000000
00101  C        BASE OF EACH TRIANGLE RUNS ALONG THE EQUATOR.  THUS,       V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           AR.OM = AREA ( THETA.M,  PHI.1 )                        V01-01D   000000
00101  C           AR.NM = AREA ( THETA.NM, PHI.1 )                        V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    12. THE SOLID ANGLE CONTRIBUTION SUBTENDED BY NEIGHBORS N AND  V01-01D   000000
00101  C        M IS CALCULATED ACCORDING TO THE FOLLOWING RULES.          V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C        IF SENSE.OM AND SENSE.NM ARE OF THE SAME SIGN, THEN        V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           ANGLE = ABS ( AR.OM - AR.NM )                           V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C        ELSE                                                       V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C          IF SENSE.ON AND SENSE.NM ARE OF THE SAME SIGN, THEN      V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           ANGLE = ABS ( 2*PI - ( AR.OM + AR.NM ) )                V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C          ELSE                                                     V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C           ANGLE = ABS ( AR.OM + AR.NM )                           V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C          END.IF                                                   V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C        END.IF                                                     V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C                                                                   V01-01D   000000
00101  C    REQUIRED SUBROUTINES                                           11500480  000000
00101  C                                                                   11500490  000000
00101  C    010         ABSARC                                             11500500  000000
00101  C    020         CROSSM                                             11500510  000000
00101  C    021         CROS1M                                             11500520  000000
00101  C    023         DOT1                                               11500530  000000
00101  C    025         NORM1M                                             11500540  000000
00101  C                                                                   11500550  000000
00101  C                                                                   11500560  000000
```

Listing 15.—Continued.

```
00101   C                                                                11500570  000000
00101   C     ERRORS                                                     11500580  000000
00101   C                                                                11500590  000000
00101   C     NONE                                                       11500600  000000
00101   C                                                                11500610  000000
00101   C                                                                11500620  000000
00101   C                                                                11500630  000000
00103         COMMON /ISIZEC/ISIZE                                       11500640  000000
00104         REAL PAV,ACAV                                              115A0680  000000
00105         REAL IPAV                          @                       V01-01D   000000
00106         REAL XL(3)                         @                       V01-01D   000000
00107         REAL P(3,5)                        @                       V01-01D   000000
00110         REAL Z(3,5)                        @                       V01-01D   000000
00111         REAL THETA(5)                      @                       V01-01D   000000
00112         INTEGER ISEN(5)                    @ ROTATION SENSE VALUES V01-01D   000000
00113         INTEGER IL,IH,IS,ID,IE                                     115A0682  000000
00114         INTEGER STAT                                                         000000
00115         DATA ID/115/                                               115A0684  000000
00117         DATA IE/1/                                                 115A0686  000000
00121         ISW=1                              @ INTERNAL RETURN SWITCH   11500710  000000
00122         CURAV=0.0                          @ NEXT AVERAGE ACCUMULATOR 11500720  000001
00123         IPAV=1.0/PAV                       @ FAST DIVIDES          11500730  000002
00124         CALL CHKLH (IL,IH,IS,ID,IE)        @ GO CHECK ID RANGE     115A0740  000005
00125         IF (IS) 14,17,14                   @ VALID ID RANGE ?      115A0745  000014
00130    14   I1=IL-IS                           @ YES, INIT ID DO LOOP  115A0750  000016
00131    15   I1=I1+IS                           @ NEXT AEROELEMENT ID   115A0755  000021
00132         IF (IS*(I1-IH)) 101,101,17         @ END OF LOOP YET ?     115A0760  000023
00135    17   ACAV=CURAV                         @ YES, END OF ROUTINE   115A0765  000030
00136         CALL SPSHI (0)                     @ END-OF-ARGUMENTS FOR SAF        000031
00137         CALL SPSHI (0)                     @  REQUEST                        000034
00140         CALL PAAI (4,I)                    @ PAX SUPPLIED ACCUMULATOR        000037
00141         GO TO (22,27),I                    @ IS IT NON-LITERAL ?             000043
00142    22   CALL SPSHR (CURAV)                 @ YES, SEND OUR SUB-TOTAL         000053
00143         CALL SPSHI (1)                     @                                 000055
00144         CALL SPSHI (7)                     @ LENGTH OF PACKAGE               000060
00145         CALL REQSAF (115,1)                @ ASK FOR OUR ONLY SAF            000063
00146         RETURN                             @ DONE                            000067
00147    27   CALL SPOPI (I)                     @ OOPS, PAX MUST GIVE A           000073
00150         CALL SPOPI (I)                     @  NON-LITERAL                    000075
00151         CALL ERROR2 (115,5)                @                                 000100
00152         RETURN                             @                       115A0770  000104
00152   C                                                                11500790  000104
00152   C     ENTRY TO VOLUME, SOLID ANGLE CALCULATOR                    11500800  000104
00152   C                                                                11500810  000104
00152   C     I1 MUST BE A LEGAL AEROELEMENT ID.                         11500820  000104
00152   C     ISW MUST BE A LEGAL INTERNAL JUMP SWITCH.                  11500830  000104
00152   C     IPAV MUST BE THE AVERAGE (DATUM) SOLID ANGLE INVERSE.      11500840  000104
00152   C                                                                11500850  000104
00152   C                                                                11500860  000104
00153   101   DO 102 I=1,3                       @ PICKUP TARGET ELEMENT'S  V01-01D   000112
00156   102   XL(I)=X(I1,I)                      @  POSITION IN SPACE    V01-01D   000112
00160         DO 104 I=1,5                       @ MAKE AND SAVE VECTORS TO V01-01D   000125
00163         J=NEI(I1,I)                        @  NEAREST NEIGHBORS REFERENCE11500880  000125
00164         DO 104 K=1,3                       @  TO BASE AEROELEMENT  11500890  000135
00167   104   P(K,I)=X(J,K)-XL(K)                @                       V01-01D   000135
00172         RAD=0.0                            @                       11500910  000147
00173         DO 108 I=1,5                       @ NORMALIZE VECTORS, ACCUM 11500920  000153
00176         CALL NORMM  (P(1,I),R)             @  MAGNITUDES           V01-01E   000153
00177    108  RAD=RAD+R                          @                       11500940  000160
00201         RAD=4.1887902*(0.5*RAD)**3         @ CALC VOLUME ESTIMATE  V01-01D   000164
00201   C                                                                11500960  000164
00201   C     NON-FOLDBACK ORDERING STARTS HERE                          11500970  000164
00201   C                                                                11500980  000164
00202         DO 112 I=2,5                       @ CREATE THE ORDERING   11500990  000175
00205         CALL CROSS  (P(1,1),P(1,I),Z(1,I)) @  VECTORS AND NORMALIZE   V01-01D   000175
00206   112   CALL NORM   (Z(1,I))               @  THEM                 V01-01D   000205
00210         DO 119 I=3,5                       @ CALCULATE RELATIVE ROTATIONS11501010  000217
00213         CALL DOT    (Z(1,2),Z(1,I),CO)     @  OF Z3 TO Z5 REFERENCED TO  V01-01D   000217
00214         CALL CROSSM (Z(1,2),Z(1,I),Z(1,1),SI) @  Z2 AND STORE IN THETA  V01-01D   000225
00215         CALL DOT    (P(1,1),Z(1,1),TH)     @                       V01-01D   000235
00216         IF (TH)     ,118,118               @                       11501050  000242
00221         SI=-SI                             @                       11501060  000245
00222    118  CALL ABSARC (TH,SI,CO)             @                       11501070  000250
00223    119  THETA(I)=TH                        @                       11501080  000254
00225         DO 130 I=1,3                       @ REORDER VECTORS IN INCREASE-11501090  000263
00230         MIN=10000.0                        @  ING VALUES OF THETA BY A   11501100  000263
00231         DO 126 J=3,5                       @  SIMPLE MINIMUM SEARCH   11501110  000267
00234         IF (THETA(J)-MIN)    ,    ,126     @                       11501120  000267
00237         MIN=THETA(J)                       @                       11501130  000275
00240         K=J                                @                       11501140  000304
00241    126  CONTINUE                           @                       11501150  000310
00243         THETA(K)=10001.0                   @  DON'T PICK THIS AGAIN   11501160  000310
```

Listing 15.—Continued.

```
00244            DO 129 J=1,3                         @  TRANSFER THE VECTOR      11501170   000313
00247      129   Z(J,I)=P(J,K)                        @                           V01-01D    000322
00251       130  CONTINUE                             @                           11501190   000331
00253            DO 134 I=1,3                         @  PUT VECTORS BACK IN NEW  11501200   000331
00256            K=I+2                                @  ORDER                    11501210   000331
00257            DO 134 J=1,3                         @                           11501220   000334
00262      134   P(J,K)=Z(J,I)                        @                           V01-01D    000342
00262      C                                                                      11501240   000342
00262      C     THIS SECTION CALCULATES THE SOLID ANGLE, OM, SUBTENDED BY THE    11501250   000342
00262      C     FIVE NEAREST NEIGHBORS.  THE FIRST NEAREST NEIGHBOR IS USED AS   11501260   000342
00262      C     A POLE POSITION.                                                 11501270   000342
00262      C                                                                      11501280   000342
00265            OM=0.0                               @ SEE TEXT FOR THEORY       11501290   000346
00266            DO 600 IJK=2,4                       @ ACCUMULATE SOLID ANGLE    V01-01D    000352
00271            N=IJK                                @  CONTRIBUTION FROM EACH OF V01-01D   000352
00272            M=IJK+1                              @  THREE SPHERICAL TRIANGLES V01-01D   000354
00273            CALL CROSS  (P(1,1),P(1,N),Z(1,1))   @ ESTABLISH A NORMALIZED    V01-01D    000356
00274            CALL NORM   (Z(1,1))                 @  NORTH POLE IN Z.1        V01-01D    000370
00275            CALL DOT    (P(1,1),P(1,N),CO)       @ CALCULATE EAST LONGITUDE  V01-01D    000373
00276            THETA(N)=ACOS(CO)                    @  OF NEIGHBOR N            V01-01D    000403
00277            CALL DOT    (Z(1,1),P(1,M),CO)       @ CALCULATE THE HEIGHT OF THE V01-01D  000410
00300            PHI1=ABS(1.5707963-ACOS(CO))         @  SPHERICAL TRIANGLES      V01-01D    000422
00301            IF (ABS(PHI1-1.5707963)-1.0E-03) 500,500,225 @ ANOMALLY ?        V01-01D    000430
00304      225   CALL CROSS  (P(1,M),Z(1,1),Z(1,5))   @ NO, PROJECT VECTOR TO     V01-01D    000435
00305      330*  CALL CROSS  (Z(1,1),Z(1,5),Z(1,2))   @  NEIGHBOR M INTO EQUATORIAL V01-01D  000445
00306            CALL NORM   (Z(1,2))                 @  PLANE AND NORMALIZE IT   V01-01D    000452
00307            CALL DOT    (P(1,1),Z(1,2),CO)       @ LONGITUDE (EAST OR WEST) OF V01-01D  000455
00310            THETA(M)=ACOS(CO)                    @  NEIGHBOR M               V01-01D    000462
00311            CALL DOT    (P(1,N),Z(1,2),CO)       @ LONGITUDE FROM NEIGHBOR N V01-01D    000467
00312            THETA(1)=ACOS(CO)                    @  TO NEIGHBOR M            V01-01D    000476
00313            ISEN(1)=1                            @ ESTABLISH ROTATIONAL SENSE V01-01D   000502
00314            CALL CROSS  (P(1,N),Z(1,2),Z(1,4))   @  FROM NEIGHBOR N TO       V01-01D    000504
00315            CALL DOT    (Z(1,1),Z(1,4),CO)       @  NEIGHBOR M               V01-01D    000513
00316            IF (CO) 254,255,255                  @                           V01-01D    000520
00321      254   ISEN(1)=-1                           @                           V01-01D    000523
00322      255   ISEN(N)=1                            @ ESTABLISH ROTATIONAL SENSE V01-01D   000526
00323            CALL CROSS (P(1,1),P(1,2),Z(1,4))    @  FROM 0 LONGITUDE TO      V01-01D    000530
00324            CALL DOT    (Z(1,1),Z(1,4),CO)       @  NEIGHBOR N               V01-01D    000535
00325            IF (CO) 259,260,260                  @                           V01-01D    000542
00330      259   ISEN(N)=-1                           @                           V01-01D    000545
00331      260   ISEN(M)=1                            @ ESTABLISH ROTATIONAL SENSE V01-01D   000550
00332            CALL CROSS  (P(1,1),Z(1,2),Z(1,4))   @  FROM 0 LONGITUDE TO      V01-01D    000552
00333            CALL DOT    (Z(1,1),Z(1,4),CO)       @  NEIGHBOR M               V01-01D    000557
00334            IF (CO) 264,275,275                  @                           V01-01D    000564
00337      264   ISEN(M)=-1                           @                           V01-01D    000567
00340      275   C=(TAN(PHI1))**2                     @ COMMON TO BOTH TRIANGLES  V01-01D    000572
00341            B=SQRT(C/(((SIN(THETA(M)))**2)+C))   @ AREA FROM 0 LONGITUDE TO  V01-01D    000576
00342            AROM=ABS(ASIN(B)-ASIN(B*COS(THETA(M)))) @  NEIGHBOR M            V01-01D    000615
00343            B=SQRT(C/(((SIN(THETA(1)))**2)+C))   @ AREA FROM NEIGHBOR N TO   V01-01D    000636
00344            ARNM=ABS(ASIN(B)-ASIN(B*COS(THETA(1)))) @  NEIGHBOR M            V01-01D    000652
00345            A=ABS(AROM-ARNM)                     @ ASSUME SIMPLEST CASE      V01-01D    000671
00346            IF (ISEN(1)*ISEN(M)) 302,600,600     @ GOOD ASSUMPTION ?         V01-01D    000674
00351      302   A=ABS(AROM+ARNM)                     @ NO, WE WILL NEED THIS     V01-01D    000700
00352            IF (ISEN(1)*ISEN(N)) 600,304,304     @ GOING OVER THE POLE ?     V01-01D    000704
00355      304   A=ABS(6.2831854-A)                   @ YES, WE CALC'D COMPLEMENT V01-01D    000711
00356            GO TO 600                            @ JOIN AFTER ANOMALLY HANDLER V01-01D  000714
00357      500   A=ABS(THETA(N))                      @ PROPORTION OF HEMISPHERE  V01-01D    000716
00360      600   OM=OM+A                              @ ACCUMULATE SOLID ANGLE    V01-01D    000721
00360      C                                                                      11501580   000721
00360      C     ACCUMULATE AVERAGE AND STORE REVISED VOLUME ESTIMATE             11501590   000721
00360      C                                                                      11501600   000721
00362            OM=ABS(OM)                           @ RECOGNIZE ALWAYS POSITIVE 11501610   000725
00363            CURAV=CURAV+OM                       @ ACCUM AVERAGE FOR NEXT TIME 11501620 000727
00364            RRR=RAD*EXP(1.0-OM*IPAV)             @ ADJUST VOLUME BY HORIZON  11511630   000731
00365            IF (AND(STAT(I1),2048)) 704,707,704  @ BR IF NOT BOUNDARY ELEMENT V01-01D   000742
00370      704   CURAV=CURAV+OM                       @ PUT BOUNDARY ELEMENTS ON  V01-01D    000750
00371            OM=OM+OM                             @  EQUAL TERMS              11511636   000753
00372            RRR=0.5*RAD*EXP(1.0-OM*IPAV)         @ REVISE VOLUME ESTIMATE    11511640   000756
00373      707   CALL STV (I1,RRR)                    @ SAVE VOLUME ESTIMATE      V01-01D    000771
00374            CALL CHKTIM (IL,IH,I1)               @ KEEP AN EYE ON THE TIME   V01-01D    000774
00375            GO TO 15                             @ NO ALTERNATE RETURNS YET  115A1650   001001
00376            END                                                              11501660   001037
END FOR
>
```

Listing 15.—Concluded.

```
 @FOR,MS  CASPER1.VSUMD
FOR   4R1   E -04/17/84-12:56:31 (1,)
>@EOF

   SUBROUTINE VSUM      ENTRY POINT 000121


   STORAGE USED: CODE(1) 000143; DATA(0) 000021; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   CHKLH
    0004   CHKTIM
    0005   V
    0006   SPSHI
    0007   PAAI
    0010   SPSHR
    0011   REQSAF
    0012   SPOPI
    0013   ERROR2
    0014   NERR2$
    0015   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000020 117G      0001   000056 24L       0001   000076 29L       0001   000110 32L       0000 I 000003 I
    0000 I 000001 ID        0000 I 000002 IE        0000   000012 INJP$     0000 I 000000 IS        0005 R 000000 V



00101      1*          SUBROUTINE VSUM (IL,IH,VL)                                          10000010    000000
00101      2*     C+                                                                      10000020    000000
00101      3*     C                                                                       10000030    000000
00101      4*     C    VSUM        ****** A SUBROUTINE FOR CASPER ******                  10000040    000000
00101      5*     C                AUTHOR      WILLIAM HENRY JONES                        10000050    000000
00101      6*     C                V01-00      02 FEB 79                                  10000060    000000
00101      7*     C                V01-00A     06 FEB 80     PAX PROTOCOL ADJUSTMENTS                 000000
00101      8*     C                                                                       10000070    000000
00101      9*     C                                                                       10000080    000000
00101     10*     C    DESCRIPTION ******                                                 10000090    000000
00101     11*     C                                                                       10000100    000000
00101     12*     C    SUMS UP THE VOLUME ESTIMATES OF ALL THE AEROELEMENTS BETWEEN       10000110    000000
00101     13*     C    'IL' AND 'IH'.                                                     10000120    000000
00101     14*     C                                                                       10000130    000000
00101     15*     C-                                                                      10000140    000000
00103     16*          INTEGER IL,IH,IS,ID,IE                                             10000150    000000
00104     17*          REAL VL                                                            10000160    000000
00105     18*          DATA ID/100/                                                       10000170    000000
00107     19*          DATA IE/1/                                                         10000180    000000
00111     20*          CALL CHKLH (IL,IH,IS,ID,IE)        @ CHECK AEROELEMENT RANGE       10000190    000000
00112     21*          IF (IS) 17,32,17                   @ VALIC RANGE ?                 10000200    000006
00115     22*      17  VL=0.0                             @ YES, DO THE SUMATION          10000210    000010
00116     23*          DO 19 I=IL,IH,IS                   @                               10000220    000020
00121     24*          CALL CHKTIM (IL,IH,I)              @ KEEP AN EYE ON THE TIME                   000020
00122     25*      19  VL=VL+V(I)                         @                               10000230    000025
00124     26*          CALL SPSHI (0)                     @ PUT END-OF-ARGUMENTS ON                   000034
00125     27*          CALL SPSHI (0)                     @  THE STACK                                000037
00126     28*          CALL PAAI (3,I)                    @ USE 'VL' AS MASTER ACCUM                  000042
00127     29*          GO TO (24,29),I                    @ PAX GIVE US VALID 'VL' ?                  000046
00130     30*     24   CALL SPSHR (VL)                    @ YES, LOAD OUR SUB-TOTAL                   000056
00131     31*          CALL SPSHI (1)                     @  AS A LITERAL                             000060
00132     32*          CALL SPSHI (7)                     @ ITEMS ON STACK                            000063
00133     33*          CALL REQSAF (100,1)                @ LET PAX ADD THE SUB-TOTALS                000066
00134     34*          RETURN                             @                                           000072
00135     35*     29   CALL SPOPI (I)                     @ CLEAN UP STACK                            000076
00136     36*          CALL SPOPI (I)                     @                                           000100
00137     37*          CALL ERROR2 (100,5)                @ REPORT BAD ARGUMENT ERROR                 000103
00140     38*     32   RETURN                             @                                           000110
00141     39*          END                                                                10000250    000142
END FOR
>
```

Listing 16.—Subroutine VSUM..

```
 @FOR,MS  CASPER1.VCORD
FOR   4R1   E -04/17/84-12:48:11 (1,)
>@EOF

   SUBROUTINE VCOR      ENTRY POINT 000051

   STORAGE USED: CODE(1) 000070; DATA(0) 000015; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   CHKLH
    0004   CHKTIM
    0005   V
    0006   STAESC
    0007   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000017 116G      0001   000040 20L       0000 I 000004 I         0000 I 000001 ID         0000 I 000002 IE
    0000   000005 INJP$     0000 I 000000 IS        0000 R 000003 R         0005 R 000000 V


00101       1*         SUBROUTINE VCOR (IL,IH,VL)                                            13200010     000000
00101       2*    C+                                                                         13200020     000000
00101       3*    C                                                                          13200030     000000
00101       4*    C    VCOR        ****** A SUBROUTINE FOR CASPER ******                     13200040     000000
00101       5*    C                AUTHOR      WILLIAM HENRY JONES                           13200050     000000
00101       6*    C                V01-00      02 FEB 79                                     13200060     000000
00101       7*    C                V01-00A     08 FEB 80      SEPARATES INPUTS AND OUTPUT                 000000
00101       8*    C                                                                          13200070     000000
00101       9*    C                                                                          13200080     000000
00101      10*    C    DESCRIPTION ******                                                    13200090     000000
00101      11*    C                                                                          13200100     000000
00101      12*    C    APPLIES A SUPPLIED MULTIPLICATIVE CORRECTION TO THE VOLUME            13200110     000000
00101      13*    C    ESTIMATES OF ALL AEROELEMENTS IN THE RANGE 'IL' TO 'IH'.              13200120     000000
00101      14*    C                                                                          13200130     000000
00101      15*    C-                                                                         13200140     000000
00103      16*         INTEGER IL,IH,IS,ID,IE                                                13200150     000000
00104      17*         REAL VL,R                                                             13200160     000000
00105      18*         DATA ID/132/                                                          13200170     000000
00107      19*         DATA IE/1/                                                            13200180     000000
00111      20*         CALL CHKLH (IL,IH,IS,ID,IE)        @ CHECK AEROELEMENT RANGE          13200190     000000
00112      21*         IF (IS) 17,20,17                   @ VALID RANGE ?                    13200200     000006
00115      22*      17 DO 19 I=IL,IH,IS                   @ YES, APPLY CORRECTION            13200210     000010
00120      23*         CALL CHKTIM (IL,IH,I)              @ KEEP AN EYE ON THE TIME                       000017
00121      24*         R=VL*V(I)                          @                                  13200220     000024
00122      25*    19   CALL STAESC (I,R)                  @ RESULT TO SCRATCH SLOT                        000031
00124      26*     20  RETURN                             @                                  13200240     000040
00125      27*         END                                                                   13200250     000067
END FOR
>
```

Listing 17.—Subroutine VCOR.

```
 @FOR,MS  CASPER1.RHOPRD
FOR   4R1   E -04/13/84-16:06:31 (2,)
>@EOF

   SUBROUTINE RHOPRS    ENTRY POINT 000073


   STORAGE USED: CODE(1) 000110; DATA(0) 000017; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   M
    0004   CHKLH
    0005   CHKTIM
    0006   V
    0007   STRHO
    0010   T
    0011   GASLAW
    0012   STP
    0013   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000017 116G       0001   000062 22L       0000 R 000004 DENS       0000 I 000003 I          0000 I 000001 ID
    0000 I 000002 IE         0000   000007 INJP$     0000 I 000000 IS         0003 R 000000 M          0000 R 000006 PRES
    0010 R 000000 T          0000 R 000005 TEMP      0006 R 000000 V



00101      1*          SUBROUTINE RHOPRS (IL,IH)                                          17600010   000000
00101      2*     C+                                                                       17600020   000000
00101      3*     C                                                                        17600030   000000
00101      4*     C    RHOPRS      ****** A SUBROUTINE FOR CASPER ******                   17600040   000000
00101      5*     C                AUTHOR      WILLIAM HENRY JONES                         17600050   000000
00101      6*     C                V01-00      02 FEB 79                                   17600060   000000
00101      7*     C                V01-00A     11 FEB 80      ADD TIME CHECK                          000000
00101      8*     C                V01-00B     15 SEP 80      FUNCTION TYPE STATEMENT                 000000
00101      9*     C                                                                        17600070   000000
00101     10*     C                                                                        17600080   000000
00101     11*     C    DESCRIPTION ******                                                  17600090   000000
00101     12*     C                                                                        17600100   000000
00101     13*     C    FOR ALL AEROELEMENTS IN THE RANGE 'IL' TO 'IH', 1) CALCULATES       17600110   000000
00101     14*     C    DENSITY BASED ON MASS AND VOLUME, AND 2) CALCULATES PRESSURE        17600120   000000
00101     15*     C    BASED ON TEMPERATURE AND DENSITY AND A CALL TO SUBROUTINE           17600130   000000
00101     16*     C    'GASLAW' (WHICH MAY BE USER SUPPLIED).                              17600140   000000
00101     17*     C                                                                        17600150   000000
00101     18*     C-                                                                       17600160   000000
00103     19*          INTEGER IL,IH,IS,ID,IE                                              17600170   000000
00104     20*          REAL M                                                                         000000
00105     21*          DATA ID/176/                                                        17600180   000000
00107     22*          DATA IE/1/                                                          17600190   000000
00111     23*          CALL CHKLH (IL,IH,IS,ID,IE)         @ CHECK AEROELEMENT RANGE       17600200   000000
00112     24*          IF (IS) 16,22,16                    @ VALID RANGE ?                 17600210   000006
00115     25*      16  DO 21 I=IL,IH,IS                    @ YES, DO THEM                  17600220   000010
00120     26*          CALL CHKTIM (IL,IH,I)               @ KEEP AN EYE ON THE TIME                  000017
00121     27*          DENS=M(I)/V(I)                      @ CALC DENSITY                  17600230   000024
00122     28*          CALL STRHO (I,DENS)                 @ UPDATE DATA BASE              17600240   000035
00123     29*          TEMP=T(I)                           @ TEMPERATURE                   17600250   000041
00124     30*          CALL GASLAW (PRES, DENS, TEMP, I)   @ USE SOMEBODY'S GAS LAW        17600260   000045
00125     31*      21  CALL STP (I,PRES)                   @ UPDATE DATA BASE              17600270   000053
00127     32*      22  RETURN                              @                               17600280   000062
00130     33*          END                                                                 17600290   000107
END FOR
>
```

Listing 18.—Subroutine RHOPRS.

```
@FOR,MS  CASPER1.GASLAD
FOR   4R1   E -04/17/84-12:58:57 (0,)
>@EOF

   SUBROUTINE GASLAW    ENTRY POINT 000012


   STORAGE USED: CODE(1) 000016; DATA(0) 000004; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003   GASCON 000001


   EXTERNAL REFERENCES (BLOCK, NAME)

    0004   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000   000000 INJP$     0003 R 000000 R



00101                 SUBROUTINE GASLAW (PR,RH,TP,ID)                                    000000
00101           C+                                                                       000000
00101           C                                                                        000000
00101           C     GASLAW      ****** A SUBROUTINE FOR CASPER ******                  000000
00101           C                 AUTHOR      WILLIAM HENRY JONES                        000000
00101           C                 V01-00      22 OCT 80                                  000000
00101           C                                                                        000000
00101           C                                                                        000000
00101           C     DESCRIPTION ******                                                 000000
00101           C                                                                        000000
00101           C     THIS ROUTINE CALCULATES THE IDEAL GAS LAW.                         000000
00101           C                                                                        000000
00101           C     PR          PRESSURE                                               000000
00101           C     RH          DENSITY                                                000000
00101           C     TP          TEMPERATURE                                            000000
00101           C     ID          ELEMENT ID (IGNORED)                                   000000
00101           C                                                                        000000
00101           C     THUS                                                               000000
00101           C                                                                        000000
00101           C     PR = RH * TP * R                                                   000000
00101           C                                                                        000000
00101           C     WHERE R IS THE IDEAL GAS CONSTANT.  THE FOLLOWING ARE APPROPRIATE  000000
00101           C     CONSTANTS FOR VARIOUS UNITS OF MEASURE:                            000000
00101           C                                                                        000000
00101           C     53.352000    (FT-LBF)/(LBM-DEGREES R)                              000000
00101           C     1716.5473    (FT-LBF)/(SLUG-DEGREES R)                             000000
00101           C     88.595808    (NT-M)/(KG-DEGRESS K)                                 000000
00101           C                                                                        000000
00101           C                                                                        000000
00101           C     THIS ROUTINE IS INITIALIZED FOR (NT-M)/(KG-DEGRESS K); HOWEVER,    000000
00101           C     THIS MAY BE ALTERED BY THE PAX/CASPER WORKER STARTUP               000000
00101           C     INITIALIZATION PROCESS.                                            000000
00101           C                                                                        000000
00101           C-                                                                       000000
00103                 COMMON /GASCON/R                   @ FOR EASY INITIALIZATION       000000
00104                 REAL PR,RH,TP,R                    @ SO EASY I ONLY HAD TO TRY     000000
00105                 DATA R/88.595808/                  @  3 TIMES TO GET IT RIGHT      000000
00107                 PR=RH*TP*R                         @                               000000
00110                 RETURN                             @                               000003
00111                 END                                @                               000015
END FOR
>
```

Listing 19.—Subroutine GASLAW.

```
 @FOR,MS  CASPER1.FNBNED
FOR   4R1   E -04/13/84-16:24:54 (2,)
>@EOF

   SUBROUTINE FNBNE     ENTRY POINT 000707


   STORAGE USED: CODE(1) 000736; DATA(0) 030116; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   ELNK
    0004   FZ
    0005   FNEIZN
    0006   CHKLH
    0007   WERR2
    0010   TREE2
    0011   X
    0012   NEI
    0013   IPLZN
    0014   NEIZN
    0015   STNEI
    0016   CHKTIM
    0017   NERR2$
    0020   SQRT
    0021   NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000047 1000L     0001   000110 1010L     0001   000147 1025L     0001   000152 1050L     0001   000200 1055L
    0001   000207 1075L     0001   000267 1088L     0001   000276 1100L     0001   000324 1300L     0001   000345 1303L
    0001   000402 1311L     0001   000410 1400L     0001   000055 170G      0001   000432 2000L     0001   000473 2012L
    0001   000512 2025L     0001   000542 2032L     0001   000112 204G      0001   000600 2040L     0001   000615 2044L
    0001   000623 2046L     0001   000124 212G      0001   000023 225L      0001   000035 228L      0001   000225 245G
    0001   000245 255G      0001   000356 312G      0001   000434 340G      0001   000476 363G      0001   000521 373G
    0001   000533 400G      0001   000551 412G      0001   000567 421G      0001   000610 434G      0001   000627 446G
    0001   000656 7L        0000 R 006405 AET       0003 I 000000 ELNK      0004 I 000000 FZ        0000 I 006367 I
    0000 I 006362 ICC       0000 I 006370 IEP       0000 I 006361 IERC      0000 I 006373 IFZBAS    0000   030074 INJP$
    0013 I 000000 IPLZN     0000 I 006363 IST       0000 I 006365 ITAR      0000 I 006364 ITARG     0000 I 006402 IZEP
    0000 I 006401 IZLC      0000 I 006400 IZLP      0000 I 006371 J         0000 I 006372 JECNT     0000 I 006375 JENEP
    0000 I 006374 JENEW     0000 I 000000 JES       0000 I 006366 JESP      0000 I 006405 JET       0000 I 006200 JZS
    0000 I 006403 JZSP      0000 I 006376 K         0000 I 006377 L         0000 I 006344 LNEI      0012 I 000000 NEI
    0014 I 000000 NEIZN     0005 I 000000 FNEIZN    0000 R 006404 R         0000 R 006354 RNEI      0011 R 000000 X
    0000 R 006351 XBAS


 00101               SUBROUTINE FNBNE (ILO,IHI,JCC,IR)                                          000000
 00101         C+                                                                               000000
 00101         C                                                                                000000
 00101         C     FNBNE      ****** A SUBROUTINE FOR CASPER ******                           000000
 00101         C                AUTHOR      WILLIAM HENRY JONES                                 000000
 00101         C     1015       X01.00      28 JAN 83                                           000000
 00101         C                X01.00A     15 FEB 83     TYPO                      X01.00A     000000
 00101         C                X01.00B     17 MAR 83     INDEX TYPO                X01.00B     000000
 00101         C                                                                                000000
 00101         C                                                                                000000
 00101         C     DESCRIPTION ******                                                         000000
 00101         C                                                                                000000
 00101         C     FNBNE (FIND BY NEIGHBOR ELEMENTS) UPDATES THE NEAREST NEIGHBORS            000000
 00101         C     LIST FOR EACH AEROELEMENT IN THE RANGE ILO TO IHI.  NEAREST                000000
 00101         C     NEIGHBOR CANDIDATES ARE TAKEN FROM THE TARGET ELEMENT'S LISTED             000000
 00101         C     NEIGHBORS, THE NEIGHBORS OF THE NEIGHBORS, AND SO ON, UNTIL AT             000000
 00101         C     LEAST JCC (CANDIDATE COUNT) INDEPENDENT (NON-REDUNDANT) ELEMENTS           000000
 00101         C     HAVE BEEN CONSIDERED.                                                      000000
 00101         C                                                                                000000
 00101         C     IT IS POSSIBLE THAT THE NEIGHBORS-OF-NEIGHBORS-OF-NEIGHBORS                000000
 00101         C     SCHEME WILL FAIL TO PRODUCE JCC CANDIDATE ELEMENTS.  IN THIS               000000
```

Listing 20.—Subroutine FNBNE.

```
00101   C   EVENT, CANDIDATE ELEMENTS ARE DRAWN FROM THE TARGET ELEMENT'S                    000000
00101   C   FLOW ZONE AND, THEN, FROM THAT FLOW ZONE'S NEIGHBORS.  IF                        000000
00101   C   THIS IS NECESSARY, THE REQUIREMENT FOR JCC CANDIDATE ELEMENTS                    000000
00101   C   IS WAIVED.                                                                       000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C   INPUT INFORMATION ******                                                         000000
00101   C                                                                                    000000
00101   C   QUANTITY    TYPE        DIMENSION     DESCRIPTION                                000000
00101   C   ---------   ---------   ---------     ---------------------------                000000
00101   C   X           REAL        1 TO ISIZE    AEROELEMENT POSITION IN                    000000
00101   C                           1 TO 3         SPACE                                     000000
00101   C                                                                                    000000
00101   C   NEI         INTEGER     1 TO ISIZE    AEROELEMENT NEAREST                        000000
00101   C                           1 TO 5         NEIGHBOR LIST                             000000
00101   C                                                                                    000000
00101   C   ELNK        INTEGER     1 TO ISIZE    AEROELEMENT LINKAGES                       000000
00101   C                           1 TO 3         ( ,1) POINTER TO NEXT                     000000
00101   C                                                ELEMENT IN FLOW                     000000
00101   C                                                ZONE                                000000
00101   C                                                                                    000000
00101   C   IPLZN       INTEGER     1 TO NZN      FLOW ZONE RESIDENT ELEMENT                 000000
00101   C                           1 TO 6         LINKAGE HEAD                              000000
00101   C                                          ( ,4) PTR TO FIRST ELEMENT                000000
00101   C                                          ( ,5) NUMBER OF ELEMENTS                  000000
00101   C                                          ( ,6) PTR TO LAST ELEMENT                 000000
00101   C                                                                                    000000
00101   C   FZ          INTEGER     1 TO ISIZE    FLOW ZONE OF ELEMENT                       000000
00101   C                                                                                    000000
00101   C   NEIZN       INTEGER     1 TO NEIZSZ   FLOW ZONE NEIGHBOR LISTS                   000000
00101   C                                                                                    000000
00101   C   PNEIZN      INTEGER     1 TO NZN      NEIZN CONTROL INFO                         000000
00101   C                                          ( ,1) STARTING POINT                      000000
00101   C                                          ( ,2) NUMBER OF NEIGHBORS                 000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C   ERRORS REPORTED BY THIS ROUTINE                                                  000000
00101   C                                                                                    000000
00101   C   CODE        ERROR                                                                000000
00101   C   ---------   ---------------------------------------------------                  000000
00101   C   1           ILO <1                                                               000000
00101   C                                                                                    000000
00101   C   2           ILO >ISIZE                                                           000000
00101   C                                                                                    000000
00101   C   3           IHI <1                                                               000000
00101   C                                                                                    000000
00101   C   4           IHI >ISIZE                                                           000000
00101   C                                                                                    000000
00101   C   5           CANDIDATE COUNT TOO LOW (OPERATION CONTINUES)                        000000
00101   C                                                                                    000000
00101   C   6           CANDIDATE COUNT TOO HIGH (OPERATION CONTINUES)                       000000
00101   C                                                                                    000000
00101   C   7           COULD NOT INSERT TARGET AEROELEMENT IN AEROELEMENT                   000000
00101   C                TREE (TREE MECHANISMS FAULT).                                       000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C                                                                                    000000
00101   C   DEFINED RETURN SWITCH (IR) VALUES                                                000000
00101   C                                                                                    000000
00101   C   VALUE       MEANING                                                              000000
00101   C   ---------   ---------------------------------------------------                  000000
00101   C   1           SUCCESS                                                              000000
00101   C   2           FAILURE                                                              000000
00101   C                                                                                    000000
00101   C   FORMAT OF AEROELEMENT TREE                                                       000000
00101   C                                                                                    000000
00101   C   A BINARY TREE IS CONSTRUCTED FOR THE PURPOSE OF ELIMINATING                      000000
```

Listing 20.—Continued.

```
00101   C     REDUNDANT ELEMENTS.  ITS FORMAT IS AS FOLLOWS:                       000000
00101   C                                                                          000000
00101   C     INDEX       CONTENTS                                                 000000
00101   C     ---------   ---------------------------------------------------      000000
00101   C     1           LENGTH OF AN ENTRY                                       000000
00101   C     2           POINTER TO FIRST FREE ENTRY                              000000
00101   C     3           NUMBER OF FREE ENTRIES REMAINING                         000000
00101   C     4           COMPARISON CONTROL CODE                                  000000
00101   C     5           POINTER TO FIRST ENTRY, IPTR, 0 => NONE                  000000
00101   C     ...                                                                  000000
00101   C     IPTR ->     < BRANCH POINTER, 0 => NONE                              000000
00101   C      +1         > BRANCH POINTER, 0 => NONE                              000000
00101   C      +2         PREVIOUS ENTRY POINTER, 0 => NONE                        000000
00101   C      +3         INFORMATION WORD                                         000000
00101   C      +4         ELEMENT ID                                               000000
00101   C      +5         ELEMENT X( ,1)                                           000000
00101   C      +6         ELEMENT X( ,2)                                           000000
00101   C      +7         ELEMENT X( ,3)                                           000000
00101   C      +8         ELEMENT'S FLOW ZONE ID                                   000000
00101   C                                                                          000000
00101   C                                                                          000000
00101   C-                                                                         000000
00103         PARAMETER OURID=106      @ CASPER CATALOG ID                         000000
00104         PARAMETER NNNEI=5        @ NUMBER OF NEAREST NEIGHBORS               000000
00105         INCLUDE TREE             @                                           000000
00120         PARAMETER NNNEIM=NNNEI-1 @                                           000000
00121         PARAMETER ERCCL=5        @ ERROR -- CANDIDATE COUNT TOO LOW          000000
00122         PARAMETER ERCCH=6        @ ERROR -- CANDIDATE COUNT TOO HIGH         000000
00123         PARAMETER JETNE=1000     @ ELEMENT TREE -- NUMBER OF ENTRIES         000000
00124         PARAMETER JETFF=6        @ ELEMENT TREE -- FIRST FREE ENTRY          000000
00125         PARAMETER JETLE=9        @ ELEMENT TREE -- LENGTH OF ENTRY           000000
00126         PARAMETER JETAL=(JETNE*JETLE)+JETFF-1 @ LENGTH OF JET ARRAY          000000
00127         PARAMETER JESAL=3200     @ SIZE OF ELEMENT STACK                     000000
00130         PARAMETER JZSAL=100      @ TEMPORARY ZONE STORAGE                    000000
00131         INTEGER JET(JETAL)       @ ELEMENT TREE TABLE                        000000
00132         INTEGER JES(JESAL)       @ ELEMENT STACK                             000000
00133         INTEGER JZS(JZSAL)       @                                           000000
00134         INTEGER ELNK             @                                           000000
00135         INTEGER LNEI(NNNEI)      @ LOCAL NEAREST NEIGHBORS LIST              000000
00136         INTEGER FZ               @                                           000000
00137         INTEGER PNEIZN           @                                           000000
00140         REAL XBAS(3)             @ TARGET AEROELEMENT'S POSITION             000000
00141         REAL RNEI(NNNEI)         @ LOCAL NEAREST NEIGHBOR RADIAL DISTANCES   000000
00142         REAL AET(JETAL)          @ REAL ACCESS TO JET                        000000
00143         EQUIVALENCE (JET(1),AET(1))         @                                000000
00144         IERC=1                              @ INIT ERROR COUNTER             000000
00145         IR=1                                @ ASSUME SUCCESS                 000001
00146         ICC=JCC                             @ DUMMY UP                       000002
00147         CALL CHKLH (ILO,IHI,IST,OURID,IERC) @ CHECK DO-LOOP RANGE            000004
00150         IERC=1                              @ RESET ERROR COUNTER            000013
00151         IF (IST)225,202,225                 @ ANY ERROR ?                    000015
00154   202   RETURN                              @ YES                            000017
00155   225   IF (ICC-6) 226,228,228              @ ENFORCE A MINIMUM CANDIDATE    000023
00160   226   ICC=6                               @  COUNT                         000026
00161         CALL WERR2 (OURID,ERCCL)            @                                000030
00162   228   IF (ICC+1-JETNE) 1000,1000,229      @ ALWAYS LEAVE SPACE IN TREE     000035
00165   229   ICC=JETNE-1                         @  FOR TARGET ELEMENT ENTRY TO   000040
00166         CALL WERR2 (OURID,ERCCH)            @  AVERT TARGET-TO-TARGET        000042
00167   1000  DO 2100 ITARG=ILO,IHI,IST           @ DO EACH ELEMENT IN RANGE       000047
00172         ITAR=ITARG                          @ DUMMY UP FOR DEC               000055
00173         JET(IETLE)=JETLE                    @ INIT TREE -- ENTRY LENGTH      000057
00174         JET(IETFF)=JETFF                    @  FIRST FREE ENTRY              000061
00175         JET(IETFC)=JETNE                    @  FREE ENTRY COUNT              000063
00176         JET(IETCC)=-1                       @  COMPARE BY INTEGERS           000065
00177         JET(IETVE)=0                        @  NO ENTRIES YET                000067
00200         JESP=0                               @ RESET ELEMENT STACK           000070
00201         CALL TREE2 (I,IEP,ITAR,JET)         @ ENTER TARGET ELEMENT           000071
00202         GO TO (1010,1010,7),I               @ CHECK RESULT JUST IN CASE      000077
00203   1010  DO 1012 I=1,3                       @ PICK UP AEROELEMENT'S          000112
00206         XBAS(I)=X(ITAR,I)                   @  POSITION             X01.00B  000112
00207   1012  CONTINUE                            @                                000124
00211         DO 1017 I=1,5                       @ PICK UP AEROELEMENT'S          000124
```

Listing 20.—Continued.

```
00214            J=I                                @ NEAREST NEIGHBORS AND PUT            000124
00215            JESP=JESP+1                        @  ON POSSIBILITY STACK                000126
00216            JES(JESP)=NEI(ITAR,J)              @                                      000131
00217      1017  CONTINUE                           @                                      000141
00221            JECNT=0                            @ CLEAR ENTRY COUNT                    000141
00222            IFZBAS=FZ(ITAR)                    @ RECORD TARGET'S FLOW ZONE            000142
00223      1025  JENEW=0                            @ CLEAR NEW ENTRY COUNT                000147
00224            JENEP=JET(IETFF)                   @ POINT TO FIRST NEW ENTRY             000147
00225      1050  IF (JESP) 1075,1075,1051           @ ANYTHING ON STACK ?                  000152
00230      1051  I=JES(JESP)                        @ YES, POP ELEMENT POSSIBILITY         000155
00231            JESP=JESP-1                        @  ID OFF THE STACK                    000157
00232            CALL TREE2 (J,IEP,I,JET)           @ INSERT IN THE TREE IF NEW            000161
00233            GO TO (1050,1055,2000),J           @ WAS IT A NEW ENTRY ?                 000167
00234      1055  JENEW=JENEW+1                      @ YES, BUMP COUNT OF NEW               000200
00235            JECNT=JECNT+1                      @ BUMP OVERALL COUNT, TOO              000202
00236            GO TO 1050                         @                                      000205
00237      1075  IF (JENEW) 1100,1100,1076          @ ANY NEW ELEMENTS LEFT ?              000207
00242      1076  I=JENEP+IEVA                       @ YES, POINT TO ELEMENT ID             000211
00243            J=JET(I)                           @ GET ELEMENT ID              X01.00A  000216
00244            DO 1081 K=1,3                      @ GET ELEMENT'S POSITION               000225
00247            L=K                                @  AND RECORD IN THE ENTRY             000225
00250            AET(I+K)=X(J,L)                    @                                      000227
00251      1081  CONTINUE                           @                                      000236
00253            JET(I+4)=FZ(J)                     @ RECORD ZONE, TOO                     000236
00254            DO 1087 K=1,5                      @ TRANSFER THE ELEMENT'S               000245
00257            L=K                                @  NEIGHBORS TO THE CANDIDATE          000245
00260            IF (JESP-JESAL) 1085,1088,1088     @  STACK (UNTIL THE STACK              000247
00263      1085  JESP=JESP+1                        @  FILLS UP, VERY UNLIKELY)            000253
00264            JES(JESP)=NEI(J,L)                 @                                      000256
00265      1087  CONTINUE                           @                                      000267
00267      1088  JENEW=JENEW-1                      @ ONE LESS 'NEW' ENTRY                 000267
00270            JENEP=JENEP+JET(IETLE)             @ POINT TO NEXT ENTRY                  000271
00271            GO TO 1075                         @ GO THROUGH ALL 'NEW' ENTRIES         000274
00272      1100  IF (JECNT-ICC) 1101,2000,2000      @ ENOUGH CANDIDATES ?                  000276
00275      1101  IF (JESP) 1200,1200,1025           @ NO, PROCESS STACK IF NONNULL         000301
00275      C+                                                                              000301
00275      C     THIS SECTION LOCATES CANDIDATE ELEMENTS BY SEARCHING FLOW ZONE            000301
00275      C     RESIDENT LISTS.  THE FIRST FLOW ZONE USED IS THE TARGET ELEMENT'S         000301
00275      C     FLOW ZONE OF RESIDENCE.  THEN A FLOW ZONE NEIGHBORS OF NEIGHBORS          000301
00275      C     OF NEIGHBORS SCHEME IS USED TO LOCATE ADDITIONAL FLOW ZONES.              000301
00275      C-                                                                              000301
00300      1200  IZLP=PNEIZN(IFZBAS,1)              @ INIT ZONE'S NEIGHBOR CONTROL         000304
00301            IZLC=PNEIZN(IFZBAS,2)              @                                      000311
00302            IZEP=IPLZN(IFZBAS,4)               @ POINT TO ZONE'S FIRST EL             000316
00303      1300  IF (IZEP) 1400,1400,1301           @ GOT AN ELEMENT ?                     000324
00306      1301  CALL TREE2 (J,I,IZEP,JET)          @ YES, PUT IT IN THE TREE              000326
00307            GO TO (1311,1303,2000),J           @ A NEW TREE ENTRY ?                   000334
00310      1303  J=I+IEVA                           @ YES                                  000345
00311            DO 1307 K=1,3                      @ TRANSFER ITS POSITION TO             000347
00314            L=K                                @  THE TREE ENTRY                      000356
00315            AET(J+K)=X(IZEP,L)                 @                                      000360
00316      1307  CONTINUE                           @                                      000367
00320            JET(J+4)=FZ(IZEP)                  @ RECORD ZONE, TOO                     000367
00321            JECNT=JECNT+1                      @ BUMP ENTRY COUNT                     000373
00322            IF (JECNT-ICC) 1311,2000,2000      @ ENOUGH ENTRIES ?                     000376
00325      1311  IZEP=ELNK(IZEP,1)                  @ NO, LOOK FOR NEXT ELEMENT            000402
00326            GO TO 1300                         @                                      000406
00327      1400  IF (IZLC) 2000,2000,1401           @ ANY MORE NEIGHBOR ZONES ?            000410
00332      1401  IZLC=IZLC-1                        @ YES, ONE LESS NOW                    000412
00333            I=NEIZN(IZLP)                      @ GET NEIGHBOR'S ID                    000414
00334            IZLP=IZLP+1                        @ POINT TO NEXT                        000420
00335            IZEP=IPLZN(I,4)                    @ POINT TO FIRST ELEMENT               000423
00336            GO TO 1300                         @                                      000430
00336      C+                                                                              000430
00336      C     THE FOLLOWING CODE DOES THE FINAL SELECTION OF NEAREST NEIGHBORS          000430
00336      C     FROM THE POPULATION IN THE ELEMENT TREE.                                  000430
00336      C-                                                                              000430
00337      2000  DO 2003 I=1,NNNEI                  @ INIT LOCAL NEIGHBOR                  000434
00342            LNEI(I)=I                          @  SELECTION TABLES                    000434
00343            RNEI(I)=1.0E+30                    @                                      000436
00344      2003  CONTINUE                           @                                      000442
00346            JENEP=JETFF+JETLE                  @ SKIP OVER TARGET'S ENTRY             000442
```

Listing 20.—Continued.

```
00347              JZSP=1                                   @ PUT TARGET AEROELEMENT'S          000444
00350              JZS(JZSP)=IFZBAS                         @  FLOW ZONE ON ZONE STACK          000447
00351              J=PNEIZN(IFZBAS,1)                       @ PUT AS MANY NEIGHBORS OF          000451
00352              K=PNEIZN(IFZBAS,2)                       @  THAT ZONE ON THE STACK           000456
00353              IF (K) 2025,2025,2010                    @  AS WILL FIT                      000463
00356        2010  IF (K-JZSAL) 2012,2011,2011              @                                   000465
00361        2011  K=JZSAL-1                                @                                   000470
00362        2012  DO 2016 L=1,K                            @                                   000473
00365              JZSP=JZSP+1                              @                                   000476
00366              JZS(JZSP)=NEIZN(I)                       @                                   000500
00367              I=I+1                                    @                                   000505
00370        2016  CONTINUE                                 @                                   000512
00372        2025  DO 2046 I=1,JECNT                        @ LOOK AT EACH CANDIDATE            000512
00375              J=JENEP+IEVA                             @ POINTER CANDIDATE'S ID            000521
00376              JENEP=JENEP+JETLE                        @ POINT TO NEXT ENTRY               000524
00377              DO 2030 K=1,JZSP                         @ CANDIDATE MUST BE IN A GOOD       000533
00402              IF (JZS(K)-JET(J+4)) 2030,2032,2030      @  FLOW ZONE TO BE CONSIDERED       000533
00405        2030  CONTINUE                                 @                                   000540
00407              GO TO 2046                               @ OOPS, CLOSE YOUR EYES, MARY       000540
00410        2032  R=0.0                                    @ ZAP RADIAL DISTANCE               000542
00411              DO 2034 K=1,3                            @ CALC DISTANCE FROM TARGET         000542
00414        2034  R=R+((AET(J+K)-XBAS(K))**2)              @  TO CANDIDATE                     000552
00416              R=SQRT(R)                                @                                   000556
00417              L=NNNEI+1                                @ L BECOMES POSITION NUMBER OF      000562
00420              DO 2039 K=NNNEI,1,-1                     @  CLOSEST NEIGHBOR BESTED BY       000567
00423              IF (R-RNEI(K)) 2039,2039,2040            @  THE CANDIDATE POSITION           000567
00426        2039  L=K                                      @                                   000573
00430        2040  IF (L-NNNEI) 2041,2044,2046              @ ANY REPLACEMENT ?                 000600
00433        2041  DO 2043 K=NNNEIM,L,-1                    @ YES, LEFT FURTHER NEIGHBORS       000603
00436              LNEI(K+1)=LNEI(K)                        @  UP ONE                           000610
00437        2043  RNEI(K+1)=RNEI(K)                        @                                   000611
00441        2044  LNEI(L)=JET(J)                           @ SLIP NEW NEIGHBOR INTO            000615
00442              RNEI(L)=R                                @  POSITION                         000620
00443        2046  CONTINUE                                 @                                   000627
00445              DO 2053 I=1,5                            @ TRANSFER RESULTS TO TARGET        000627
00450              J=I                                      @  ELEMENT'S STORAGE AREA           000627
00451              CALL STNEI (ITAR,J,LNEI(J))              @                                   000631
00452        2053  CONTINUE                                 @                                   000643
00454              CALL CHKTIM (ILO,IHI,ITAR)               @ KEEP AN EYE ON THE TIME           000643
00455        2100  CONTINUE                                 @ END OF TARGET LOOP                000652
00457              RETURN                                   @ THAT'S ALL !!                     000652
00457        C+                                                                                 000652
00457        C     ERROR REPORTING                                                              000652
00457        C-                                                                                 000652
00460        7     IERC=IERC+6            @ FAILED TO INSERT TARGET ELEMENT                     000656
00461              IR=2                   @ SET FAILURE RETURN                                  000660
00462              CALL WERR2 (OURID,IERC) @ REPORT ERROR                                       000662
00463              RETURN                 @                                                     000666
00464              END                    @                                                     000735
END FOR
>
```

Listing 20.—Concluded.

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
NASA TP-2418		

4. Title and Subtitle

Combined Aerodynamic and Structural Dynamic Problem Emulating Routines (CASPER) - Theory and Implementation

5. Report Date

February 1985

6. Performing Organization Code

505-40-5A

7. Author(s)

William H. Jones

8. Performing Organization Report No.

E-2278

10. Work Unit No.

9. Performing Organization Name and Address

National Aeronautics and Space Administration
Lewis Research Center
Cleveland, Ohio 44135

11. Contract or Grant No.

13. Type of Report and Period Covered

Technical Paper

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, D.C. 20546

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The Combined Aerodynamic and Structural Dynamic Problem Emulating Routines (CASPER) is a collection of data-base modification computer routines that can be used to simulate Navier-Stokes flow through realistic, time-varying internal flow fields. The Navier-Stokes equation used involves calculations in all three dimensions and retains all viscous terms. The only term neglected in the current implementation is the gravitational term. The solution approach is of an iterative, time-marching nature. Calculations are based on Lagrangian aerodynamic elements (aeroelements). It is assumed that the relationships between a particular aeroelement and its five nearest neighbor aeroelements are sufficient to make a valid simulation of Navier-Stokes flow on a small scale and that the collection of all small-scale simulations makes a valid simulation of a large-scale flow. In keeping with these assumptions, it must be noted that CASPER produces an imitation or simulation of Navier-Stokes flow rather than a strict numerical solution of the Navier-Stokes equation. CASPER is written to operate under the Parallel, Asynchronous Executive (PAX), which is described in a separate report.

17. Key Words (Suggested by Author(s))

Computational fluid mechanics
Navier-Stokes solutions
Navier-Stokes particle tracking

18. Distribution Statement

[redacted]
Until February 1987

STAR Category 02

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	74	