208
11-14-78

14. 725-

LA-7313-MS

Informal Report

MASTER

**IONEOS: A Fast, Analytic,
Ion Equation-of-State Routine**

University of California

LOS ALAMOS SCIENTIFIC LABORATORY

Post Office Box 1663 Los Alamos, New Mexico 87545

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

LA-7313-MS
Informal Report
UC-21
Issued: October 1978

IONEOS: A Fast, Analytic, Ion Equation-of-State Routine

C. W. Cranfill
R. More*

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Department of Energy, nor any of their employees nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights.

*Lawrence Livermore Laboratory, Livermore, CA 94550.

IONEOS: A FAST, ANALYTIC, ION EQUATION-OF-STATE ROUTINE

by

C. W. Cranfill and R. More

ABSTRACT

A routine IONEOS has been developed to compute ion pressure and specific energy along with their derivatives. The theoretical model used is due to R. D. Cowan and includes non-ideal-gas behavior. The routine is designed for use in large hydrodynamics codes as an improved procedure for splitting the total pressure and specific energy (which are usually tabulated) into ion and electron components.

I. INTRODUCTION

This report describes the routine IONEOS, which has been developed for the rapid evaluation of ion equation-of-state quantities using a model due to R. D. Cowan. The calculated ion quantities can be subtracted from the total fluid pressure and specific energy to obtain the corresponding electron quantities for partially ionized systems which are characterized by two separate temperatures. While accurate total equation-of-state quantities require very involved calculations and hence are usually tabulated for use in sophisticated hydrodynamics codes, IONEOS has proven fast enough ($\sim 50 \mu s$ per phase-space point on a CDC-7600) to use on-line for splitting off the ion contributions.

In the following sections the units (Section II), formulas (Section III), and code structure (Section IV) of IONEOS are summarized, and the implementation (Section V) in large hydrodynamics codes is outlined.

II. UNITS

The system of units adopted for IONEOS was that commonly used in large hydrodynamics codes at the Los Alamos Scientific and Lawrence Livermore Laboratories. These units are mass in gm, volume in cm^3 , temperature in $\text{keV} = 1.6021 \times 10^{-9} \text{ erg}$, and energy in $\text{Jk} = 10^{16} \text{ erg}$. The following thermodynamic quantities result:

$$\text{mass density} \quad \rho \quad \text{g cm}^{-3} \quad , \quad (1a)$$

$$\text{temperature} \quad T \quad \text{keV} \quad , \quad (1b)$$

$$\text{fluid pressure} \quad P \quad \text{Jk cm}^{-3} \quad , \quad (1c)$$

$$\text{specific energy} \quad E \quad \text{Jk g}^{-1} \quad . \quad (1d)$$

The units of the thermodynamic derivatives $\partial E/\partial T$, $\partial P/\partial T$, $\partial E/\partial \rho$, $\partial P/\partial \rho$ are dictated by these choices.

To illustrate these units, consider the usual ideal-gas equation-of-state:

$$P = \rho RT/A \quad , \quad E = 3RT/2A \quad , \quad (2a,b)$$

where A is the atomic mass and $R = 8.3143 \times 10^7 \text{ erg/g/}^\circ\text{K}$ is the gas constant in cgs units. Changing to IONEOS units is accomplished by simply expressing the gas constant as $R = .096487 \text{ Jk/g/keV}$.

III. FORMULAS

In the approximate theory used in IONEOS, the fluid pressure P and the specific energy E are considered to be composed of totally separate ion and electron parts:

$$P_i = P_i(\rho, T_i) \quad , \quad E_i = E_i(\rho, T_i) \quad ,$$

$$P_e = P_e(\rho, T_e) \quad , \quad E_e = E_e(\rho, T_e) \quad .$$

Here ρ and T are the mass density and temperature, and the subscripts i and e represent ions and electrons. In reality, there is no sharp separation between the ion and electron quantities, which more properly depend on both T_i and T_e . However, such a complicated dependence has been ignored in this treatment to obtain a tractable model.

The ion equation-of-state is described by explicit formulas for P_i and E_i in three different regions of phase space -- one region represents fluids and the other two represent solids. The fluid region corresponds to

temperature above the melting curve and is represented by a scaling formula to model non-ideal-gas behavior near the phase change. The melting temperature is calculated from the Lindemann criterion, which allows a solid to melt when the rms vibration amplitude of the atoms reaches a certain fraction of the atomic separation. The two solid-state regions correspond to high and low temperatures and are represented by the classical and quantum Debye-solid models, respectively. In both cases a Gruneisen parameter gives the pressure from the specific energy.

The ion equation-of-state formulas developed by R. D. Cowan depend on material properties only through the atomic mass A and atomic number Z . It is convenient to define the following:

$$C \equiv .096487/A \quad \text{Jk/g/keV} \quad , \quad (3a)$$

$$\rho_R \equiv A Z^{-0.3}/9 \quad \text{g/cm}^3 \quad , \quad (3b)$$

$$\xi \equiv \rho/\rho_R \quad , \quad \beta \equiv 0.6 Z^{1/9} \quad , \quad (3c,d)$$

$$T_M \equiv .00032(1 + \xi)^{-4} \xi^{4+2\beta-2/3} \text{ keV} \quad , \quad (3e)$$

$$T_D \equiv .00168(Z + 22)^{-1}(1 + \xi)^{-2} \xi^{2+\beta} \text{ keV} \quad . \quad (3f)$$

Here C is a modified gas constant, ρ_R is a reference mass density, ξ is a normalized local mass density, β is a dimensionless constant, T_M is the Lindemann melting temperature, and T_D is the Debye temperature. The appropriate temperature variables and Gruneisen parameters for the fluid and solid phases (subscripted F and S) are

$$\phi_F \equiv (T_M/T_i)^{1/3} \quad , \quad \phi_S \equiv T_D/T_i \quad , \quad (4a,b)$$

$$\gamma_F \equiv 3\beta - 1 + 6(1 + \xi)^{-1} \quad , \quad \gamma_S \equiv \beta + 2(1 + \xi)^{-1} \quad , \quad (4c,d)$$

$$\gamma'_F \equiv \gamma_F + \frac{2}{9}\gamma_F^2 - 6\xi(1 + \xi)^{-2} \quad , \quad \gamma'_S \equiv \beta + 2(1 + \xi)^{-2} \quad . \quad (4e,f)$$

The ion equation-of-state formulas for the three phase-space regions are given in Eqs. (5 - 7). The thermodynamic consistency of these relations has been carefully checked, as has the smooth joining of ion pressure and specific energy at the phase boundaries.* These formulas give reasonable behavior in

* The jump in specific energy across the boundary between the two solid-state regions is at most 3%.

the ideal-gas limit, the solid-state range, and the compressed state up to many thousands of megabars pressure. It is remarkable that simple analytic functions produce a realistic equation-of-state. There may be problems, however, with special materials such as foams or cold porous materials.

A comparison has been made with the nuclear equation-of-state developed by D. T. Warren and used in the code TFCMIX at Lawrence Livermore Laboratory. The Cowan model is much more realistic at low pressures near the solid state. At megabar pressures the two models are in reasonable agreement. At extremely high pressures the Cowan model becomes less accurate (relative to one-component plasma simulations) than the Warren theory. However, at the pressures where this disagreement is large (above 10^4 Mbar), the ion contribution to the total pressure is small for $Z \geq 10$.

Fluid region ($\phi_F < 1$) ---

$$E_i = \frac{3}{2} CT_i(1 + \phi_F) \quad , \quad P_i = \rho CT_i (1 + \gamma_F \phi_F) \quad , \quad (5a,b)$$

$$\frac{\partial E_i}{\partial T_i} = C(\frac{3}{2} + \phi_F) \quad , \quad \frac{\partial P_i}{\partial T_i} = \rho C(1 + \frac{2}{3} \gamma_F \phi_F) \quad , \quad (5c,d)$$

$$\frac{\partial E_i}{\partial \rho} = \rho^{-2}(P_i - T_i \frac{\partial P_i}{\partial T_i}) \quad , \quad \frac{\partial P_i}{\partial \rho} = CT_i(1 + \gamma'_F \phi_F) \quad . \quad (5e,f)$$

High-temperature solid region ($\phi_F \geq 1$, $\phi_S < 3$) ---

$$E_i = 3CT_i(1 + \phi_S^2/20 - \phi_S^4/1680) \quad , \quad (6a)$$

$$P_i = \gamma_S \rho E_i \quad , \quad (6b)$$

$$\frac{\partial E_i}{\partial T_i} = 3C(1 - \phi_S^2/20 + \phi_S^4/560) \quad , \quad (6c)$$

$$\frac{\partial P_i}{\partial T_i} = \gamma_S \rho \frac{\partial E_i}{\partial T_i} \quad , \quad (6d)$$

$$\frac{\partial E_i}{\partial \rho} = \rho^{-2}(P_i - T_i \frac{\partial P_i}{\partial T_i}) \quad , \quad (6e)$$

$$\frac{\partial P_i}{\partial \rho} = \gamma'_S E_i + \gamma_S \rho \frac{\partial E_i}{\partial \rho} \quad . \quad (5f)$$

Low-temperature solid region ($\phi_F \geq 1$, $\phi_S \geq 3$) ---

$$E_i = 3CT_i \left[\frac{3}{8} \phi_S + \frac{1}{5} \pi^4 / \phi_S^3 - (3 + 9/\phi_S + 18/\phi_S^2 + 18/\phi_S^3) \exp(-\phi_S) \right] , \quad (7a)$$

$$P_i = \gamma_S \rho E_i , \quad (7b)$$

$$\frac{\partial E_i}{\partial T_i} = 3C \left[\frac{4}{5} \pi^4 / \phi_S^3 - (3\phi_S + 12 + 36/\phi_S + 72/\phi_S^2 + 72/\phi_S^3) \exp(-\phi_S) \right] , \quad (7c)$$

$$\frac{\partial P_i}{\partial T_i} = \gamma_S \rho \frac{\partial E_i}{\partial T_i} , \quad (7d)$$

$$\frac{\partial E_i}{\partial \rho} = \rho^{-2} (P_i - T_i \frac{\partial P_i}{\partial T_i}) , \quad (7e)$$

$$\frac{\partial P_i}{\partial \rho} = \gamma_S' E_i + \gamma_S \rho \frac{\partial E_i}{\partial \rho} . \quad (7f)$$

IV. CODE STRUCTURE :

Since IONEOS was designed to be a subroutine for sophisticated hydrodynamics codes, its structure was chosen to obtain fast computation speeds. For the CDC-7600 used at the Los Alamos Scientific Laboratory, this involved passing input and output variables through common blocks (rather than argument lists) and minimizing the use of slow operations (such as divides and exponentiations). Furthermore, the various options are organized with the most likely branches occurring first, and corrections to the simpler models are not calculated unless they are significant. When timed while computing ion equation-of-state quantities on a large grid spanning seven decades in both ρ and T_i , IONEOS averaged 50 μ s per grid point on a CDC-7600. A full listing of IONEOS is given in Appendix A.

An example of the results obtained for gold is shown in Fig. 1. Plotted versus density are the ion pressure and specific energy for several temperatures. The deviation from ideal-gas behavior (i.e. from straight lines) is evident even for the 77.4 eV temperature. The asymptotic boundary at high density is the cold curve (0 eV).

Fig. 1. IONEOS for Au

V. IMPLEMENTATION IN LARGE HYDRODYNAMICS CODES

In typical hydrodynamics codes such as those used in laser-fusion target design, the equation-of-state package determines the total fluid pressure P and specific energy E by interpolation from thermodynamic tables generated off-line by separate codes. To model an ionized fluid with distinct temperatures T_i and T_e for ions and electrons, it is necessary to split the total equation-of-state quantities into their ion and electron components.

Previously, codes written to simulate ionized fluids have provided two options for splitting apart the ion and electron equations-of-state:

1st option -

$$P_i = P(\rho, T_i) N_i / (N_i + N_e) \quad , \quad E_i = E(\rho, T_i) N_i / (N_i + N_e) \quad , \quad (8a,b)$$

$$P_e = P(\rho, T_e) N_e / (N_i + N_e) \quad , \quad E_e = E(\rho, T_e) N_e / (N_i + N_e) \quad ; \quad (8c,d)$$

2nd option -

$$P_i = \rho R T_i / A \quad , \quad E_i = 3 R T_i / 2 A \quad , \quad (9a,b)$$

$$P_e = P(\rho, T_e) - \rho R T_e / A \quad , \quad E_e = E(\rho, T_e) - 3 R T_e / 2 A \quad . \quad (9c,d)$$

Here N_i and N_e are the respective ion and electron number densities, and the other quantities are defined in Sections II and III. Note that the 1st splitting assumes the ion and electron contributions are proportional to the partial number densities, while the 2nd splitting assumes the ions behave like an ideal gas. In both cases summing the ion and electron contributions recovers the tabulated values when $T_i = T_e$.

The implementation of subroutine IONEOS is done by adding a third equation-of-state splitting option:

3rd option -

$$P_i = P_i(\rho, T_i) \quad , \quad E_i = E_i(\rho, T_i) \quad , \quad (10 a,b)$$

$$P_e = P(\rho, T_e) - P_i(\rho, T_e) \quad , \quad E_e = E(\rho, T_e) - E_i(\rho, T_e) \quad . \quad (10 c,d)$$

The functions P_i and E_i are computed by IONEOS as described in Section III. To minimize the number of calls to IONEOS, the ion contributions are subtracted from the tabulated equation-of-state values as they are read in during problem set-up. Thus, during problem execution the electron equation-of-state quantities are obtained directly by interpolation from the stored tables, and IONEOS need be called only to calculate the ion equation of state.

More experience with fluid simulations using the IONEOS splitting option is needed before the impact on predictions of fluid behavior can be assessed. However, Fig. 1 shows that the real ion equation-of-state near solid density deviates up to quite high temperatures from the ideal gas behavior assumed in the other splitting options. Hence simulations of such phenomena as the propagation of the first shock through a laser-driven target may be substantially modified, thus altering the conditions for target optimization.

Appendix A. Listing of Subroutine IONEOS.

```

1      SUBROUTINE IONEOS
2      COMMON INPUT RHO,TEMP,ATOMA,ATOMZ
3      COMMON OUTPUT EION,FION,DEIDTI,DFIDTI,DEIDRHO,DFIDRHO
4
5      THIS SUBROUTINE USES THE COWAN NUCLEAR MODEL TO CALCULATE ION
6      CONTRIBUTIONS TO THERMODYNAMIC QUANTITIES IN FLUIDS AND SOLIDS.
7      THE INPUT AND OUTPUT VARIABLES ARE THE FOLLOWING:
8
9      RHO      = MASS DENSITY (GM/CM3)
10     TEMP     = ION TEMPERATURE (KEV)
11     ATOMA    = ATOMIC MASS
12     ATOMZ    = ATOMIC CHARGE
13     EION     = ION SPECIFIC ENERGY (JK/GM)
14     FION     = ION FLUID PRESSURE (JK/CM3)
15     DEIDTI   = DERIVATIVE OF EION WRT TEMP (JK/GM/KEV)
16     DFIDTI   = DERIVATIVE OF FION WRT TEMP (JK/CM3/KEV)
17     DEIDRHO  = DERIVATIVE OF EION WRT RHO (JK/CM3/GM3)
18     DFIDRHO  = DERIVATIVE OF FION WRT RHO (JK/GM)
19
20     COMPUTE USEFUL CONSTANTS.
21     THIRD    = .3333333333
22     THIRD2   = .1111111111
23     CGAS    = 0.096487/ATOMA
24     CTEMP    = CGAS*TEMP
25     IF(RHO.LE.0.) GO TO 10
26     RHOREF   = THIRD2*ATOMA*(ATOMZ**(-0.3))
27     XI       = RHO/RHOREF
28     RXI      = 1./(1.+XI)
29     RXI2     = RXI**2
30     BETA     = 0.6*(ATOMZ**THIRD2)
31     TMELT    = 0.00032*(RXI2**2)*XI**13.+(THIRD+2.)*BETA)
32     IF(TEMP.LE.TMELT) GO TO 30
33
34
35     USE FLUID FORMULAE WHEN TEMP.GT.TMELT.
36
37     PHIF3    = TMELT/TEMP
38     IF(PHIF3.GE.(1.0E-18)) GO TO 20
39     10 CONTINUE
40
41
42     USE IDEAL-GAS EOS IF PHIF3.LT.E-18.
43     EION     = 1.5*CTEMP
44     FION     = RHO*CTEMP
45     DEIDTI   = 1.5*CGAS
46     DFIDTI   = RHO*CGAS
47     DEIDRHO  = 0.
48     DFIDRHO  = CTEMP
49     GO TO 30
50     20 CONTINUE
51
52     USE SCALING FORMULA IF PHIF3.GE.E-18.
53     PHIF     = PHIF3**THIRD
54     GAMF     = 3.*BETA - 1. + 6.*RXI
55     GAMF1    = GAMF*(1.+2.*THIRD2*GAMF) - 6.*XI*RXI2
56     RRHO2    = (1./RHO)**2
57     EION     = 1.5*CTEMP*(1.+PHIF)
58     FION     = RHO*CTEMP*(1.+GAMF*PHIF)
59     DEIDTI   = CGAS*(1.5+PHIF)
60     DFIDTI   = RHO*CGAS*(1.+2.*THIRD2*GAMF*PHIF)
61     DEIDRHO  = RRHO2*(FION-TEMP*DFIDTI)
62     DFIDRHO  = CTEMP*(1.+GAMF1*PHIF)
63     GO TO 30
64     30 CONTINUE

```

```

65 C
66 C
67 C   USE DEBYE-SOLID EOS WHEN TEMP.LE.TMELT.
68 C
69 C
70     GAMB   = IETA + 2.*R*T
71     GAMB1  = IETA + 2.*R*12
72     GRHO   = GAMB*RH0
73     RRHO2   = (1./RH0)**2
74     TDEBYE  = 0.00166*R 12*(V1**2.+IETA)/(ATOMZ+22.)
75     IF(TEMP.LE.(TDEBYE*THIRD)) GO TO 40
76 C
77 C   USE CLASSICAL DEBYE EXPANSION IF TEMP.GT.(TDEBYE*THIRD).
78     PHIE    = TDEBYE/TEMP
79     PHIE2    = PHIE**2
80     EION     = 3.*CTEMP*(1.+0.05*PHIE2*(1.-PHIE2*1.13047619E-2))
81     PION     = GRHO*EION
82     DEIDTI   = 3.*CGAS*(1.-0.05*PHIE2*(1.-PHIE2*3.571428571E-2))
83     DPIDTI   = GRHO*DEIDTI
84     DEIDRHO  = RRHO2*(PION-TEMP*DPIDTI)
85     DRIDRHO  = GAMB1*EION + GRHO*DEIDRHO
86     GO TO 30
87 40  CONTINUE
88 C
89 C   USE QUANTUM DEBYE EXPANSION IF TEMP.LE.(TDEBYE*THIRD).
90     RPHIE    = TEMP/TDEBYE
91     PHIE      = 0.
92     XPHIE     = 0.
93     IF(RPHIE.GT.0.) PHIE = 1./RPHIE
94     IF(RPHIE.GT.(.01*THIRD)) XPHIE = EXPF(-PHIE)
95     ETERM     = 58.4454545*(RPHIE**3) -
96     .         (9. + 27.*RPHIE*(1.+2.*RPHIE*(1.+RPHIE)))*XPHIE
97     EION      = 1.125*CGAS*TDEBYE + CTEMP*ETERM
98     PION      = GRHO*EION
99     DEIDTI    = CGAS*(4.*ETERM - 3.*PHIE*XPHIE)
100    DPIDTI    = GRHO*DEIDTI
101    DEIDRHO   = RRHO2*(PION-TEMP*DPIDTI)
102    DRIDRHO   = GAMB1*EION + GRHO*DEIDRHO
103 90  CONTINUE
104    RETURN
105    END

```