

LA-9861-M

Manual

C.2

Los Alamos National Laboratory is operated by the University of California for the United States Department of Energy under contract W-7405-ENG-36.

For Reference

Not to be taken from this room



*MIXPAC: A Subroutine Package
for Calculating Equations of State
for Equilibrium Mixtures of Materials*

Los Alamos Los Alamos National Laboratory
Los Alamos, New Mexico 87545

An Affirmative Action/Equal Opportunity Employer

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of 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. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

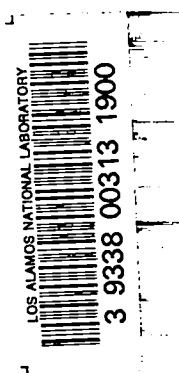
LA-9861-M
Manual

UC-34
Issued: August 1983



MIXPAC: A Subroutine Package for Calculating Equations of State for Equilibrium Mixtures of Materials

C. W. Cranfill



Contents

Abstract.	1
0. Introduction.	1
1. Equations of State for Equilibrium Mixtures.	3
2. Mixture Equations of State Calculated by MIXPAC.	7
2.1 Thermodynamic Properties.	8
2.2 Transport Coefficients.	9
3. The MIXPAC Package.	10
3.1 MIXPAC Implementation.	11
3.2 MIXPAC Subroutines.	12
Acknowledgements.	17
References.	18

MIXPAC: A Subroutine Package for Calculating Equations of State
for Equilibrium Mixtures of Materials

by

C. W. Cranfill

Abstract

This manual describes MIXPAC, a subroutine package for calculating equations of state (i.e., thermodynamic and transport properties) for plasmas composed of equilibrium mixtures of materials. The package is vectorized for the Los Alamos Cray-1 computers and uses EOSPAC, another vectorized subroutine package, to access the Los Alamos Sesame EOS data library. Each mixture is forced to be in equilibrium through the constraints that all its constituents have the same values for two state functions (e.g., temperature and pressure). The desired equations of state (including first partial derivatives) are then calculated for the mixture consistent with these constraints. All equations of state available for pure materials through EOSPAC are available for equilibrium mixtures through MIXPAC.

0. Introduction.

The dynamical state of a plasma is described by certain equations of state (EOS) which give thermodynamic properties (e.g., pressure and specific energy) and transport coefficients (e.g., opacity and conductivity) as functions of the ambient plasma conditions. The atomic-physics models for EOS are sufficiently complex that the results are often tabulated off-line for use in dynamics calculations. The Los Alamos Sesame EOS data library is such a tabulation, giving the values of several state functions for various materials, mass densities, and temperatures.

To get the EOS for a plasma composed of a mixture of materials, it is necessary to combine the EOS for the constituents subject to some equilibrium constraints. It is impractical to tabulate off-line the EOS for all possible mixtures, so if the material composition is to change during a dynamics calculation, the EOS for the constituents must be combined in-line to give the EOS for each new mixture.

This manual describes MIXPAC, a subroutine package for calculating in-line the EOS for an equilibrium mixture given its composition and the EOS for its constituents. MIXPAC is vectorized for the Los Alamos CRAY-1 computers and uses EOSPAC (Ref. 1), another vectorized subroutine package, to access the Los Alamos Sesame EOS data library to obtain the EOS for the constituents. Each mixture is forced to be in equilibrium through the constraints that all its constituents have the same values for two state functions (e.g., temperature and pressure).

Section 1 gives the basic relations needed to calculate the EOS for an equilibrium mixture given its composition and the EOS for its constituents. Section 2 gives the specific mixture equations of state calculated by MIXPAC. Section 3 gives a description of the implementation and subroutines of the MIXPAC package.

1. Equations of State for Equilibrium Mixtures.

Consider a plasma occupying a volume V and containing several ionic constituents, where constituent α consists of N_α ions having charge state z_α and atomic mass m_α . The concentration of constituent α is $c_\alpha \equiv N_\alpha/N$, where $N \equiv \sum_\alpha N_\alpha$, and the average charge state and atomic mass for the mixture are

$$z \equiv \sum_\alpha c_\alpha z_\alpha \quad , \quad m \equiv \sum_\alpha c_\alpha m_\alpha \quad .$$

The ion and electron number densities and the mass density are then

$$n_i = N/V \quad , \quad n_e = zN/V \quad , \quad \rho = mN/V \quad .$$

It is desired to approximate a state function S for the mixture given the state functions S_α for all constituents.

Every state function is assumed to have as independent variables the specific volume $v \equiv V/N$ and a quantity y which is one of the following:

temperature	T	for temperature-based functions,
specific pressure	$p \equiv Pv$	for pressure-based functions,
specific energy	$e \equiv Ev$	for energy-based functions.

Also, every state function for the mixture is assumed to be given by

$$S = \sum_\alpha c_\alpha S_\alpha(v_\alpha, y_\alpha) \quad , \quad v = \sum_\alpha c_\alpha v_\alpha \quad , \quad y = \sum_\alpha c_\alpha y_\alpha \quad ,$$

where the v_α, y_α are chosen to force the mixture into some equilibrium. Specifically, the v_α, y_α are chosen so that all constituents have the same values for two state functions -- the temperature T_α and pressure P_α :

$$T_\alpha(v_\alpha, y_\alpha) = T(v, y) \quad , \quad P_\alpha(v_\alpha, y_\alpha) = P(v, y) \quad , \quad \forall \alpha, v, y \quad .$$

These assumptions are sufficient to specify the first partial derivatives of a state function as well as the function itself for the mixture. It is convenient to introduce the Jacobian for any two functions $f(v,y)$ and $g(v,y)$:

$$\frac{\partial(f,g)}{\partial(v,y)} \equiv \frac{\partial f}{\partial v} \Big|_y \frac{\partial g}{\partial y} \Big|_v - \frac{\partial g}{\partial v} \Big|_y \frac{\partial f}{\partial y} \Big|_v = - \frac{\partial(g,f)}{\partial(v,y)} .$$

Every state function $S = \sum_{\alpha} c_{\alpha} S_{\alpha}(v_{\alpha}, y_{\alpha})$ then satisfies

$$\begin{aligned} \frac{\partial S}{\partial v} \Big|_y &= \frac{\partial T}{\partial v} \Big|_y \frac{\partial S}{\partial T} \Big|_P + \frac{\partial P}{\partial v} \Big|_y \frac{\partial S}{\partial P} \Big|_T , & \frac{\partial S}{\partial y} \Big|_v &= \frac{\partial T}{\partial y} \Big|_v \frac{\partial S}{\partial T} \Big|_P + \frac{\partial P}{\partial y} \Big|_v \frac{\partial S}{\partial P} \Big|_T , \\ \frac{\partial S}{\partial T} \Big|_P &= \sum_{\alpha} c_{\alpha} \frac{\partial(S_{\alpha}, P_{\alpha})}{\partial(v_{\alpha}, y_{\alpha})} / \frac{\partial(T_{\alpha}, P_{\alpha})}{\partial(v_{\alpha}, y_{\alpha})} = \sum_{\alpha} c_{\alpha} \frac{\partial S_{\alpha}}{\partial T_{\alpha}} \Big|_{P_{\alpha}} , \\ \frac{\partial S}{\partial P} \Big|_T &= - \sum_{\alpha} c_{\alpha} \frac{\partial(S_{\alpha}, T_{\alpha})}{\partial(v_{\alpha}, y_{\alpha})} / \frac{\partial(T_{\alpha}, P_{\alpha})}{\partial(v_{\alpha}, y_{\alpha})} = \sum_{\alpha} c_{\alpha} \frac{\partial S_{\alpha}}{\partial P_{\alpha}} \Big|_{T_{\alpha}} . \end{aligned}$$

Applying these relations to the cases $S = v$ and $S = y$ gives

$$\begin{aligned} \frac{\partial T}{\partial v} \Big|_y &= \frac{\partial y}{\partial P} \Big|_T / \frac{\partial(v,y)}{\partial(T,P)} , & \frac{\partial T}{\partial y} \Big|_v &= - \frac{\partial v}{\partial P} \Big|_T / \frac{\partial(v,y)}{\partial(T,P)} , \\ \frac{\partial P}{\partial v} \Big|_y &= - \frac{\partial y}{\partial T} \Big|_P / \frac{\partial(v,y)}{\partial(T,P)} , & \frac{\partial P}{\partial y} \Big|_v &= \frac{\partial v}{\partial T} \Big|_P / \frac{\partial(v,y)}{\partial(T,P)} . \end{aligned}$$

Also, applying these relations to the specific energy, $S = e$, verifies that the mixture is thermodynamically consistent if its constituents are --

$$\frac{\partial e}{\partial P} \Big|_T = -T \frac{\partial v}{\partial T} \Big|_P - P \frac{\partial v}{\partial P} \Big|_T .$$

Unless the T_{α}, P_{α} are given by simple analytic expressions, the equilibrium constraints determine the v_{α}, y_{α} only implicitly and must be inverted numerically. A suitable inversion technique is Newton iteration starting with $v_{\alpha}^{(0)} = v, y_{\alpha}^{(0)} = y$ and requiring at each iteration n that

$$\sum_{\alpha} c_{\alpha} v_{\alpha}^{(n)} = v , \quad \sum_{\alpha} c_{\alpha} y_{\alpha}^{(n)} = y .$$

The general iteration algorithm is as follows:

$$T_{\alpha}^{(n)} \equiv T_{\alpha}(v_{\alpha}^{(n)}, y_{\alpha}^{(n)}) \quad , \quad P_{\alpha}^{(n)} \equiv P_{\alpha}(v_{\alpha}^{(n)}, y_{\alpha}^{(n)}) \quad ,$$

$$v_{\alpha}^{(n+1)} \equiv v_{\alpha}^{(n)} + \left[(T^{(n)} - T_{\alpha}^{(n)}) \frac{\partial P_{\alpha}^{(n)}}{\partial y_{\alpha}} \Big|_{v_{\alpha}} - (P^{(n)} - P_{\alpha}^{(n)}) \frac{\partial T_{\alpha}^{(n)}}{\partial y_{\alpha}} \Big|_{v_{\alpha}} \right] / \frac{\partial (T_{\alpha}^{(n)}, P_{\alpha}^{(n)})}{\partial (v_{\alpha}, y_{\alpha})} \quad ,$$

$$y_{\alpha}^{(n+1)} \equiv y_{\alpha}^{(n)} - \left[(T^{(n)} - T_{\alpha}^{(n)}) \frac{\partial P_{\alpha}^{(n)}}{\partial v_{\alpha}} \Big|_{y_{\alpha}} - (P^{(n)} - P_{\alpha}^{(n)}) \frac{\partial T_{\alpha}^{(n)}}{\partial v_{\alpha}} \Big|_{y_{\alpha}} \right] / \frac{\partial (T_{\alpha}^{(n)}, P_{\alpha}^{(n)})}{\partial (v_{\alpha}, y_{\alpha})} \quad .$$

The average quantities $T^{(n)}, P^{(n)}$ converge to the values T, P for the mixture and are chosen so that $\sum_{\alpha} c_{\alpha} (v_{\alpha}^{(n+1)} - v_{\alpha}^{(n)}) = 0 = \sum_{\alpha} c_{\alpha} (y_{\alpha}^{(n+1)} - y_{\alpha}^{(n)})$.

The iteration algorithm simplifies for temperature-based functions, which have the y_{α} given explicitly by $y_{\alpha} \equiv T_{\alpha} = T \equiv y$. The v_{α} are determined by Newton iteration starting with $v_{\alpha}^{(0)} = v$ and proceeding at each iteration n as follows:

$$P_{\alpha}^{(n)} \equiv P_{\alpha}(v_{\alpha}^{(n)}, T_{\alpha}) \quad , \quad v_{\alpha}^{(n+1)} \equiv v_{\alpha}^{(n)} + (P^{(n)} - P_{\alpha}^{(n)}) / \frac{\partial P_{\alpha}^{(n)}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} \quad .$$

The average quantity $P^{(n)}$ is chosen so that $\sum_{\alpha} c_{\alpha} (v_{\alpha}^{(n+1)} - v_{\alpha}^{(n)}) = 0$. In this case, the expressions for the first partial derivatives of P and any other state function $S = \sum_{\alpha} c_{\alpha} S_{\alpha}(v_{\alpha}, T_{\alpha})$ can be simplified to give

$$1 = \frac{\partial P}{\partial v} \Big|_T \sum_{\alpha} c_{\alpha} / \frac{\partial P_{\alpha}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} \quad , \quad \frac{\partial P}{\partial T} \Big|_v = \frac{\partial P}{\partial v} \Big|_T \sum_{\alpha} c_{\alpha} \frac{\partial P_{\alpha}}{\partial T_{\alpha}} \Big|_{v_{\alpha}} / \frac{\partial P_{\alpha}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} \quad ,$$

$$\frac{\partial S}{\partial v} \Big|_T = \frac{\partial P}{\partial v} \Big|_T \sum_{\alpha} c_{\alpha} \frac{\partial S_{\alpha}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} / \frac{\partial P_{\alpha}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} \quad ,$$

$$\frac{\partial S}{\partial T} \Big|_v = \sum_{\alpha} c_{\alpha} \left[\frac{\partial S_{\alpha}}{\partial T_{\alpha}} \Big|_{v_{\alpha}} + \left(\frac{\partial P}{\partial T} \Big|_v - \frac{\partial P_{\alpha}}{\partial T_{\alpha}} \Big|_{v_{\alpha}} \right) \frac{\partial S_{\alpha}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} / \frac{\partial P_{\alpha}}{\partial v_{\alpha}} \Big|_{T_{\alpha}} \right] \quad .$$

The iteration algorithm also simplifies for pressure-based functions, which have the y_α given explicitly by $y_\alpha \equiv p_\alpha = P_\alpha v_\alpha = P v_\alpha = p v_\alpha / v \equiv y v_\alpha / v$. The v_α are determined by Newton iteration starting with $v_\alpha^{(0)} = v$ and proceeding at each iteration n as follows:

$$p_\alpha^{(n)} \equiv p v_\alpha^{(n)} / v, \quad T_\alpha^{(n)} \equiv T_\alpha(v_\alpha^{(n)}, p_\alpha^{(n)}),$$

$$v_\alpha^{(n+1)} \equiv v_\alpha^{(n)} + (T^{(n)} - T_\alpha^{(n)}) / \left(\left. \frac{\partial T_\alpha^{(n)}}{\partial v_\alpha} \right|_{p_\alpha} + \frac{p}{v} \left. \frac{\partial T_\alpha^{(n)}}{\partial p_\alpha} \right|_{v_\alpha} \right).$$

The average quantity $T^{(n)}$ is chosen so that $\sum_\alpha c_\alpha (v_\alpha^{(n+1)} - v_\alpha^{(n)}) = 0$. In this case, the expressions for the first partial derivatives of T and any other state function $S = \sum_\alpha c_\alpha S_\alpha(v_\alpha, p_\alpha)$ can be simplified to give

$$\left. \frac{\partial}{\partial v} \right|_p = \left. \frac{\partial}{\partial v} \right|_p + \frac{p}{v} \left. \frac{\partial}{\partial p} \right|_v, \quad 1 = \left. \frac{\partial T}{\partial v} \right|_p \sum_\alpha c_\alpha / \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha},$$

$$v \left. \frac{\partial T}{\partial v} \right|_p = \left. \frac{\partial T}{\partial v} \right|_p \sum_\alpha c_\alpha v_\alpha \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha} / \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha},$$

$$p \left. \frac{\partial T}{\partial p} \right|_v = \left. \frac{\partial T}{\partial v} \right|_p \sum_\alpha c_\alpha p_\alpha \left. \frac{\partial T_\alpha}{\partial p_\alpha} \right|_{v_\alpha} / \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha},$$

$$v \left. \frac{\partial S}{\partial v} \right|_p = \sum_\alpha c_\alpha \left[v_\alpha \left. \frac{\partial S_\alpha}{\partial v_\alpha} \right|_{p_\alpha} + \left(v \left. \frac{\partial T}{\partial v} \right|_p - v_\alpha \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha} \right) \left. \frac{\partial S_\alpha}{\partial v_\alpha} \right|_{p_\alpha} / \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha} \right],$$

$$p \left. \frac{\partial S}{\partial p} \right|_v = \sum_\alpha c_\alpha \left[p_\alpha \left. \frac{\partial S_\alpha}{\partial p_\alpha} \right|_{v_\alpha} + \left(p \left. \frac{\partial T}{\partial p} \right|_v - p_\alpha \left. \frac{\partial T_\alpha}{\partial p_\alpha} \right|_{v_\alpha} \right) \left. \frac{\partial S_\alpha}{\partial v_\alpha} \right|_{p_\alpha} / \left. \frac{\partial T_\alpha}{\partial v_\alpha} \right|_{p_\alpha} \right].$$

2. Mixture Equations of State Calculated by MIXPAC.

All equations of state available for pure materials through the EOSPAC package (Ref. 1) are available for equilibrium mixtures through the MIXPAC package. MIXPAC attempts to minimize the Gibbs free energy for each mixture by balancing the temperatures and pressures for all constituents. The temperatures are readily available for all constituents as either the independent or dependent variables of the EOS data tables accessed through EOSPAC. Pressures consistent with the thermodynamic properties are also readily available for all constituents through EOSPAC, but pressures consistent with the transport coefficients must be approximated using ideal Fermi gas relationships. The mixture equations of state calculated by MIXPAC are described in the following subsections. They should be fairly realistic for monatomic gases and plasmas, but since they ignore chemical reactions, they are less reliable for molecular gases and plasmas.

2.1 Thermodynamic Properties.

The thermodynamic properties calculated for equilibrium mixtures by MIXPAC are

temperature-based pressure P and specific energy E/ρ ,
pressure-based temperature T and specific energy E/ρ ,
energy-based temperature T and pressure P .

These properties are available either for the total plasma or for the separate ion and electron components. The independent variables are the mass density ρ and a quantity η which is one of the following:

temperature T for temperature-based functions,
specific pressure $\pi \equiv P/\rho$ for pressure-based functions,
specific energy $\epsilon \equiv E/\rho$ for energy-based functions.

Note that these variables were chosen because of historical precedents rather than convenience. In its internal calculations, MIXPAC replaces ρ , π , ϵ with $v \equiv m/\rho$, $p \equiv m\pi$, $e \equiv m\epsilon$ as described in Section 1.

For thermodynamic properties, MIXPAC requires as inputs the values ρ , η , c_α and $m_\alpha \forall \alpha$, and the functions $T_\alpha(\rho', \eta')$ and $P_\alpha(\rho', \eta') \forall \alpha, \rho', \eta'$, where the functions are in the form of data tables accessed through the EOSPAC package (Ref. 1). Following the mixture algorithm of Section 1, MIXPAC iterates to insure that each mixture satisfies the temperature- and pressure-balance equilibrium constraints. Maximum computational speed is achieved for temperature- or pressure-based functions, which satisfy the respective temperature- or pressure-balance constraints explicitly without iteration.

2.2 Transport Coefficients.

The transport coefficients calculated for equilibrium mixtures by MIXPAC are

Rosseland- and Planck-mean photon opacities k_R and k_P ,
electron-conductive opacity k_c and ionic charge state z ,
electron electrical and thermal conductivities σ and κ ,
electron thermoelectric coefficient β .

The independent variables are the mass density ρ and the temperature T . In its internal calculations, MIXPAC replaces ρ with $v \equiv m/\rho$ and treats as state functions (see Section 1) the quantities mk_R , mk_P , zk_c , z , $z\sigma^{-1}$, $z\kappa^{-1}$, $z\beta^{-1}$.

For transport coefficients, MIXPAC requires as inputs the values ρ , T , c_α and $m_\alpha v_\alpha$, and the functions $z_\alpha(\rho', T')$ $\forall \alpha, \rho', T'$, where the functions are in the form of data tables accessed through the EOSPAC package (Ref. 1). Since pressures consistent with the transport coefficients are not readily available, they are approximated using ideal Fermi gas relationships --

$$P_\alpha \approx n_\alpha k T_\alpha f_\alpha(n_\alpha, T_\alpha) \quad , \quad n_\alpha \equiv z_\alpha / v_\alpha \quad ,$$

where f_α is a ratio of Fermi integrals. Following the mixture algorithm of Section 1, MIXPAC iterates to insure that each mixture satisfies the pressure-balance equilibrium constraint, which in this approximation reduces to an equal-number-density condition -- $n_\alpha = n \forall \alpha$. Being temperature-based, the transport coefficients satisfy the temperature-balance equilibrium constraint explicitly without iteration.

3. The MIXPAC Package.

The MIXPAC package is a set of FORTRAN subroutines that can be used to calculate the EOS for an equilibrium mixture of materials. This package is designed for use in dynamical hydrocode calculations on the Los Alamos Cray-1 computers and uses the EOSPAC package (Ref. 1) to access the Los Alamos Sesame EOS data library to obtain the EOS for a mixture's constituents. Like EOSPAC, MIXPAC assumes that the host code deals with a physical problem consisting of discrete elements called zones which are organized into groups called regions. The zones and regions must both be indexed with consecutive integers 1,2,... . Each region is associated with a single primary material and with some number of other regions whose primary materials mix with it. The host code must specify the indices of all regions whose materials are to mix in each region and the concentrations of all materials in each zone.

The implementation and subroutines of the MIXPAC package are described in the following subsections. For convenient access, MIXPAC has been appended to the EOSPAC package. The Cray FORTRAN source code and binary load library for both packages are stored on the Common File System (CFS) at Los Alamos and can be obtained by any validated user knowing the password:

```
MASS GET PW=%% /RUBY/ S/EOSPACS or L/EOSPACL .
```

EOSPAC requires the Los Alamos CFTMATH and CALMATH function libraries.

3.1 MIXPAC Implementation.

The MIXPAC package has been designed for easy implementation in most host codes using the EOSPAC package (Ref. 1) to access the Los Alamos Sesame EOS data library. It is assumed that the EOSPAC set-up phase is performed to read the necessary data tables into an EOS table storage area and convert them into a form suitable for the host code. MIXPAC then replaces the EOSPAC search-interpolate phase to use the stored data tables to compute the mixture EOS values required by the host code.

MIXPAC contains an executive controller MX1VALS which is meant to directly replace the EOSPAC executive controller ES1VALS in controlling all search-interpolate operations. Like ES1VALS, MX1VALS is heavily vectorized and receives all necessary information through its subroutine arguments (see Section 3.2):

```
CALL MX1VALS (IEOSF, IDRVS, INTRP, KTABS,  
             NZONS, NMIXR, IMIXR, CMIXR,  
             IREGS, XVALS, YVALS, FVALS, IERR1) .
```

Note that the subroutine arguments are the same as for ES1VALS except that three have been added to describe the mixture:

NMIXR is the maximum number of regions whose materials mix in any region,

IMIXR is an array containing the indices of all regions whose materials mix in each region,

CMIXR is an array containing the concentrations in all zones of each material.

3.2 MIXPAC Subroutines.

(1) SUBROUTINE MXIVALS (IEOSF, IDRVS, INTRP, KTABS,
NZONS, NMIXR, IMIXR, CMIXR,
IREGS, XVALS, YVALS, FVALS, IERR1)

Purpose -- Controls all search-interpolate operations necessary to use the data tables stored in the EOS table storage area to compute the EOS values required by the host code.

Comments -- The computation is heavily vectorized and can produce the interpolated values of an EOS function $F(X,Y)$ and its derivatives with respect to $\ln(X,Y)$.

Arguments --

IEOSF = input Hollerith name of the EOS function to be computed. (A list of recognized names is given below.)
IDRVS = input Hollerith name of the derivatives to be computed. "FUNC" gives the EOS function $F(X,Y)$ only, "DERIVS" gives $F(X,Y)$ plus the derivs wrt $\ln(X,Y)$.
INTRP = input Hollerith name of the interpolator to be used. "BILINE" gives 4-point bi-linear, "BIQUAD" gives 6-point bi-quadratic, "BIRATF" gives 12-point bi-rational-function.
KTABS = input integer pointer to the EOS table storage area.
NZONS = input integer number of zones to be processed.
NMIXR = input integer maximum number of regions whose materials mix in any region.
IMIXR = input integer array having dimension NMIXR, NREGS and containing the indices of all regions whose materials mix in each region. (NREGS is the total number of regions.)
CMIXR = input real array having dimension NZONS, NMIXR and containing the concentrations in all zones of each material.
IREGS = input integer array having dimension NZONS and containing the region index for each zone.
XVALS = input real array having dimension NZONS and containing the value of the X independent variable for each zone.
YVALS = input real array having dimension NZONS and containing the value of the Y independent variable for each zone.
FVALS = output real array having dimension NZONS, NVALS and containing the EOS value storage area. (NVALS = 1 or 3.)
IERR1 = output integer error indicator. The possible values are
00 if there are no errors,
21 if the EOS table storage area has not been initialized,
23 if the region index is out of bounds,
24 if the data table is not in the EOS table storage area,
25 if the derivative is not recognized,
26 if the interpolator is not recognized,
27 if the iteration did not converge,
31 if the EOS data type is not recognized.

The independent variables of the EOS functions are

X variable = mass density.

Y variable = temperature if temperature-based,
thermal pressure/mass density if pressure-based,
thermal specific energy if energy-based.

The recognized names of category 1 EOS functions are

"PRTOT " = temperature-based total pressure.
"ENTOT " = temperature-based total specific energy.
"TPTOT " = pressure -based total temperature.
"TNTOT " = energy -based total temperature.
"PNTOT " = energy -based total pressure.
"EPTOT " = pressure -based total specific energy.
"PRION " = temperature-based ion pressure.
"ENION " = temperature-based ion specific energy.
"TPION " = pressure -based ion temperature.
"TNION " = energy -based ion temperature.
"PNION " = energy -based ion pressure.
"EPION " = pressure -based ion specific energy.
"PRELC " = temperature-based electron pressure.
"ENELC " = temperature-based electron specific energy.
"TPELC " = pressure -based electron temperature.
"TNELC " = energy -based electron temperature.
"PNELC " = energy -based electron pressure.
"EPELC " = pressure -based electron specific energy.
"PRCLD " = temperature-based cold curve pressure.
"ENCLD " = temperature-based cold curve specific energy.

The recognized names of category 2 EOS functions are

"OPACR " = temperature-based Rosseland-mean opacity.
"OPACC2" = temperature-based electron conductive opacity.
"ZFREE2" = temperature-based mean ionic charge.
"OPACP " = temperature-based Planck-mean opacity.

The recognized names of category 3 EOS functions are

"ZFREE3" = temperature-based mean ionic charge.
"ECONDE" = temperature-based electron electrical conductivity.
"TCONDE" = temperature-based electron thermal conductivity.
"THERME" = temperature-based electron thermoelectric coefficient.
"OPACC3" = temperature-based electron conductive opacity.

(2) SUBROUTINE MX2FUNC (IEOSF, IDRVS, INTRP, KTABS, ITYPS, ISFUN,
ICATE, IBASE, LLOGS, NVALS, IINTP, IERR2)

Purpose -- Gets the attributes of a function F(X,Y) for an EOS data type in a designated EOS table storage area.

Comments -- The function attributes are needed by subroutines MX2LOOK, MX2TBAS, MX2PBAS, and MX2EBAS.

Arguments --

IEOSF = input Hollerith name of the EOS function to be computed. (A list of recognized names is given above.)
IDRVS = input Hollerith name of the derivatives to be computed. (See description of MX1VALS.)
INTRP = input Hollerith name of the interpolator to be used. (See description of MX1VALS.)
KTABS = input integer pointer to the EOS table storage area.
ITYPS = output integer array having dimension 3 and containing the indices of the EOS data type and its temperature- and pressure-balance functions.
ISFUN = output integer state-function type for the EOS data type.
ICATE = output integer data category for the EOS data type.
IBASE = output integer base indicator for the EOS data type.
LLOGS = output logical log indicator for the EOS data type.
NVALS = output integer number of values to be computed per zone. (See description of MX1VALS.)
IINTP = output integer index of the interpolator to be used.
1 gives 4-point bi-linear,
2 gives 6-point bi-quadratic,
3 gives 12-point bi-rational-function.
IERR2 = output integer error indicator. The possible values are 00, 21, 24, 25, 26, 31. (See description of MX1VALS.)

(3) SUBROUTINE MX2LOOK (NZONS, IREGS, IZONE, NVECT, IERR2)

Purpose -- Performs a vectorized look-ahead to determine the number of consecutive zones with the same or zero region index.

Comments -- Subroutine ES2FUNC must have been called before this subroutine is called. The look-ahead determines the number of zones that can be processed as a vector, and this vector length is constrained to be at most 64.

Arguments --

NZONS = input integer number of zones to be processed.
IREGS = input integer array having dimension NZONS and containing the region index for each zone.
IZONE = in/out integer index of the next zone to be processed.
NVECT = output integer length of the search vectors.
IERR2 = output integer error indicator. The possible values are 00, 23. (See description of MX1VALS.)

(4) SUBROUTINE MX2TBAS (ITYPS, IREGN,
ISFUN, ICATE, LLOGS, NVALS, IINTP,
NZONS, NVECT, NMIXR, IMIXR, CMIXR,
XVALV, YVALV, FVALV, IERR2)

Purpose -- Iterates a vectorized interpolation to compute a temperature-based EOS function $F(X,Y)$ and its derivatives with respect to $\ln(X,Y)$ for a mixture of materials in pressure-balance.

Comments -- Subroutine MX2LOOK must have been called before this subroutine is called. The iteration is by Newton's method.

Arguments --

ITYPS = input integer array having dimension 3 and containing the indices of the EOS data type and its temperature- and pressure-balance functions.

IREGN = input integer index of the region.

ISFUN = input integer state-function type for the EOS data type.

ICATE = input integer data category for the EOS data type.

LLOGS = input logical log indicator for the EOS data type.

NVALS = input integer number of values to be computed per zone.
(See description of MX1VALS.)

IINTP = input integer index of the interpolator to be used.
1 gives 4-point bi-linear,
2 gives 6-point bi-quadratic,
3 gives 12-point bi-rational-function.

NZONS = input integer number of zones to be processed.

NVECT = input integer length of the search vectors.

NMIXR = input integer maximum number of regions whose materials mix in any region.

IMIXR = input integer array having dimension NMIXR, NREGS and containing the indices of all regions whose materials mix in each region. (NREGS is the total number of regions.)

CMIXR = input real array having dimension NZONS, NMIXR and containing the concentrations in all zones of each material.

XVALV = input real search vector having dimension NVECT and containing the X value for each zone.

YVALV = input real search vector having dimension NVECT and containing the Y value for each zone.

FVALV = output real array having dimension NZONS, NVALS and containing the interpolated values of the function $F(X,Y)$ and its derivatives wrt X,Y.

IERR2 = output integer error indicator. The possible values are 00, 27. (See description of MX1VALS.)

(5) SUBROUTINE MX2PBAS (ITYPS, IREGN,
ISFUN, ICATE, LLOGS, NVALS, IINTP,
NZONS, NVECT, NMIXR, IMIXR, CMIXR,
XVALV, YVALV, FVALV, IERR2)

Purpose -- Iterates a vectorized interpolation to compute a pressure-based EOS function $F(X,Y)$ and its derivatives with respect to $\ln(X,Y)$ for a mixture of materials in temperature-balance.

Comments -- Subroutine MX2LOOK must have been called before this subroutine is called. The iteration is by Newton's method.

Arguments -- Same as for subroutine MX2TBAS.

(6) SUBROUTINE MX2EBAS (ITYPS, IREGN,
ISFUN, ICATE, LLOGS, NVALS, IINTP,
NZONS, NVECT, NMIXR, IMIXR, CMIXR,
XVALV, YVALV, FVALV, IERR2)

Purpose -- Iterates a vectorized interpolation to compute an energy-based EOS function $F(X,Y)$ and its derivatives with respect to $\ln(X,Y)$ for a mixture of materials in pressure- and temperature-balance.

Comments -- Subroutine MX2LOOK must have been called before this subroutine is called. The iteration is by Newton's method.

Arguments -- Same as for subroutine MX2TBAS.

(7) SUBROUTINE MX3NAME (INAME, ITEMP, IPRES, ISFUN, ICATE, IBASE,
IERR3)

Purpose -- Translates the name of an EOS data type into the information about the corresponding data table.

Comments -- This subroutine is called by MX2FUNC.

Arguments --

INAME = input Hollerith name of the EOS data type.
ITEMP = output Hollerith name of the temperature-balance function.
IPRES = output Hollerith name of the pressure-balance function.
ISFUN = output integer state-function type for the EOS data type.
ICATE = output integer data category for the EOS data type.
IBASE = output integer base indicator for the EOS data type.
IERR3 = output integer error indicator. The possible values are 00, 31. (See description of MX1VALS.)

Acknowledgements

Several discussions with Gerald I. Kerley were very helpful in the development of the prescription for mixture equations of state described in Section 1 of this manual.

References

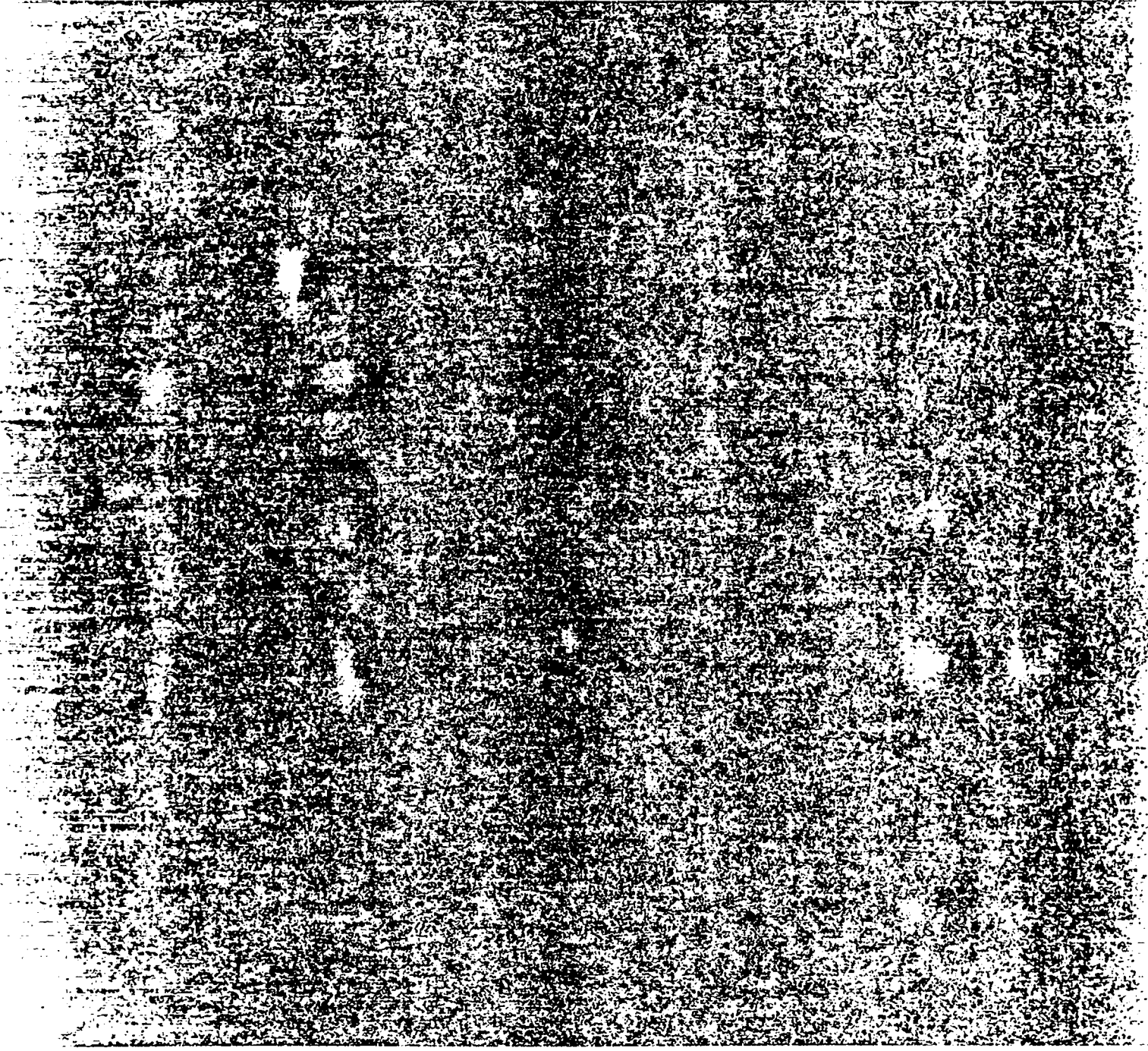
1. C. W. Cranfill, "EOSPAC: A Subroutine Package for Accessing the Los Alamos Sesame EOS Data Library", Los Alamos National Laboratory report LA-9728-M (August 1983).

Printed in the United States of America
Available from
National Technical Information Service
US Department of Commerce
5285 Port Royal Road
Springfield, VA 22161

Microfiche (A01)

<u>Page Range</u>	<u>NTIS Price Code</u>	<u>Page Range</u>	<u>NTIS Price Code</u>	<u>Page Range</u>	<u>NTIS Price Code</u>	<u>Page Range</u>	<u>NTIS Price Code</u>
001-025	A02	151-175	A08	301-325	A14	451-475	A20
026-050	A03	176-200	A09	326-350	A15	476-500	A21
051-075	A04	201-225	A10	351-375	A16	501-525	A22
076-100	A05	226-250	A11	376-400	A17	526-550	A23
101-125	A06	251-275	A12	401-425	A18	551-575	A24
126-150	A07	276-300	A13	426-450	A19	576-600	A25
						601-up*	A99

*Contact NTIS for a price quote.



Los Alamos