FEOS

A new equation-of-state code for hot dense matter

Package Documentation

Version: 16.7 (July 2016)

Dr. Steffen Faik
Institute for Applied Physics
Goethe University Frankfurt am Main, Germany

MPQeos Version 2.0 (09/99):

A.J. Kemp and J. Meyer-ter-Vehn
Max-Planck Institute for Quantum Optics
Garching, Germany

Foreword / Contact

First of all I would like to thank Dr Anna Tauschwitz, Prof Dr Joachim Maruhn (both Goethe University Frankfurt am Main & GSI Darmstadt), Prof Dr Igor Iosilevskiy (Joint Institute for High Temperatures Moscow & GSI Darmstadt), and Prof Dr Mikhail Basko (Institute for Theoretical and Experimental Physics Moscow & GSI Darmstadt) for very helpful and constructive discussions. The whole work was supported by the Extreme Matter Institute EMMI and the Bundesministerium für Bildung und Forschung BMBF (Project 06FY9085).

This present new revised version of the original code MPQeos by A. Kemp [1] was built within the last four years, partially within the framework of my diploma thesis at the university of Frankfurt, Germany. For this reason, the new package is called "Frankfurt equation-of-state (FEOS)". It was mainly built for usage at GSI – Helmholtz Center for Heavy Ion Research in Darmstadt, but is already used by several institutions and working groups around the world.

At this point I want to emphasize that some parts of this documentation are based on the original documentation of MPQeos (version 2.0) by A. Kemp and J. Meyer-ter-Vehn. I updated these parts concerning the new features of FEOS. So, the reader does not have to read the original documentation of MPQeos in order to understand the new manual.

Address of the author of FEOS

Dr. Steffen Faik
Nikolausstr. 10
65936 Frankfurt am Main

Phone: +49 (0)69 30858211
steffen@faik.net
http://physik.faik.net/

Contents

The FEOS Package

1 Introduction / Overview

The FEOS package is a C++ computer code which consists of three parts:

1. the FEOS library (file ”libfeos.a”),
2. the FEOS table generation tool (executable ”feos”),
3. the SHOWEOS table visualization tool (executable ”showeos”).

The main part of the new package, the FEOS library, provides all the routines which are needed to calculate the equation-of-state (EOS) of an, in principle, arbitrary material as a function of density and temperature. The underlying physical model is the ”quotidian equation-of-state model (QEOS)”, described further in Ref. [2]. The background of the original code MPQeos, together with some applications, is described in a correspondig MPQ Report 229 [3]. Having outsourced the main EOS routines into a static library an implementation of the EOS directly into the user's code was made possible. For this purpose the library provides several interfaces, for codes written in C, C++, or Fortran. Furthermore, a file containing a database of material parameters which is accessed by the FEOS library is intended to be exchanged between a growing community of FEOS users.

The second part, the FEOS table generation tool makes use of the FEOS library and can be used to create EOS table files as a function of density and temperature. Several output formats, like the SESAME format, already come with the package, but the user is free to write her/his own output routines. Also, the FEOS table generation tool provides the possibility to perform ”simple” user-defined calculations which require an EOS.

The SHOWEOS table visualization tool can be used to visualize the FEOS (or SESAME) table files, i.e. to make plots of isotherms, isochores, isentropes, Hugoniot curves, or to show three-dimensional phase planes.

Figure 1 gives an overview of the whole package's structure. The documentation explains in detail how to use the three different parts of the package and gives hints how and where to change the source code in order to customize it for the user's own needs. The part which deals with the FEOS

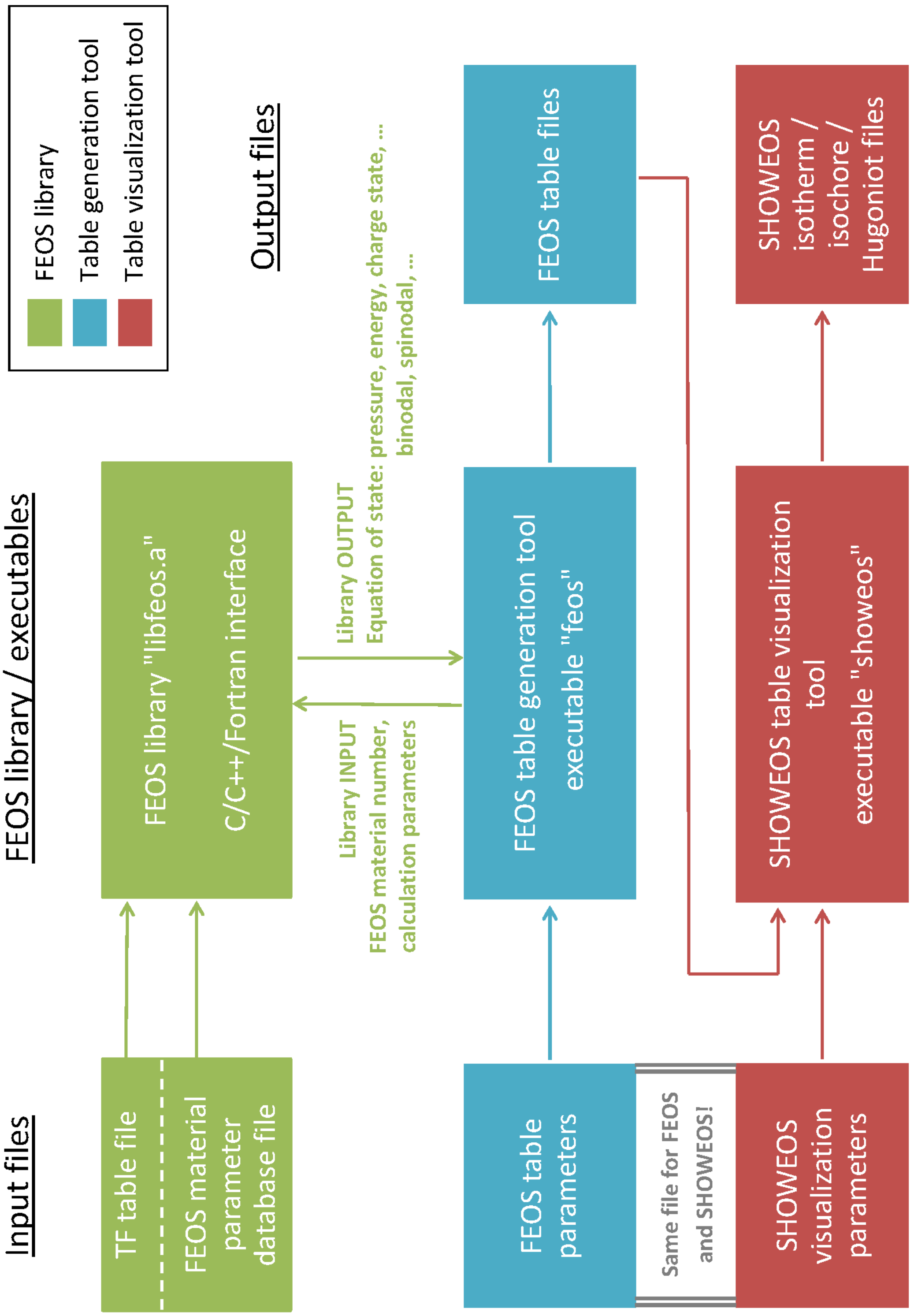

Input files
FEOS library / executables
Output files
🟩 FEOS library
🟦 Table generation tool
🟥 Table visualization tool
TF table file
FEOS material parameter database file
FEOS library "libfeos.a"
C/C++/Fortran interface
Library INPUT
FEOS material number, calculation parameters
Library OUTPUT
Equation of state: pressure, energy, charge state, ... binodal, spinodal, ...
FEOS table parameters
Same file for FEOS and SHOWEOS!
SHOWEOS visualization parameters
FEOS table generation tool
executable "feos"
FEOS table files
SHOWEOS table visualization tool
executable "showeos"
SHOWEOS isotherm / isochore / Hugoniot files

Figure 1: Outline of the FEOS package.

library includes also a short description of the physical model and the improvements and changes applied to the original MPQeos code. Furthermore, the library interface routines and their implementation into any user's code are explained in detail.

1.1 Citation Rules

Whenever the FEOS package or parts of it are used the most recent publication in the journal "Computer Physics Communications" or Reference [4], which contains a short section about the package, must be cited. The user can find links to all publications on the webpage http://physik.faik.net. Also, since the package relies on the QEOS model and on the MPQeos code, one should always cite the original publications (Ref. [1, 2]).

2 Package File Structure

The FEOS package comes in a packed file "FEOS_16.7_20160729.zip". Under Unix this file can be unzipped with the command "unzip". After having unzipped the file, three directories will be present:

1. /Code → contains all source files of the package,
2. /Documents → documentation and QEOS/MPQeos pdf files,
3. /EOS-Data → working directory for table generation and visualization.

The directory /Code contains all the source files of the package. These files are subdivided into four classes. The files which begin with the prefix "COMMON-" (Table 1) contain the routines which are used by all three parts of the FEOS package. The files which begin with the prefix "LIB-" contain all routines of the FEOS library. They are explained in detail in the library part of the documentation. Furthermore, the library interface header file "libfeos.h" also belongs to the FEOS library. The files which begin with the prefixes "FE-" / "SE-" contain the routines of the FEOS table generation / FEOS table visualization tools. Their content is explained in detail in the corresponding parts of the documentation. Finally, the Makefile contains all commands for the compilation of the code (Section 3).

COMMON-00_DEFINITS.H	Conversion factors, physical and mathematical constants, and EOS table storage structures
COMMON-01_UTILITIES.C	Routines for memory allocation and deallocation
COMMON-02_READFILE.C	Routines for reading ASCII files

Table 1: Common source files used by all parts of the FEOS package.

The directory /Documents contains the documentations of the FEOS package and several pdf files about QEOS and MPQeos.

In the directory /EOS-Data all the calculations with the table generation and the table visualization tools, whose exectuables will be present after installation, are supposed to be done. The directory contains the precalculated Thomas-Fermi table file for hydrogen and the material parameter database file. These two files are both read by the FEOS library. The two examplary parameter files for aluminum (Al) and fused silica (SiO_2) contain all the necessary settings for the table generation and table visualization tools. The general structure of the material parameter database and the tools' parameter files will be explained in Section 4. Details on the settings inside these files are given in the corresponding sections of the documentation.

3 Installation

The Makefile contains all information for the compilation of the source code. The whole source code inside the directory /Code is compiled under Unix with the command "make all". By default, the GNU C++ compiler g++ is used. After FEOS has been installed, the FEOS library file "libfeos.a" will be present. Also the executables "feos" and "showeos" of the package tools will be there and automatically copied to the tools' working directory /EOS-Data. In order to reset the package to the uncompiled status, the command "make clean" can be used and all executables and object files will be deleted. In principle, the code can be also compiled under Windows, but then the user has to take care by himself/herself of the compilation.

4 Parameter & Database File Structure

A parameter file ”Materialname.par” and the material parameter database file ”FEOS_Material-DB.dat” are ASCII files devided into several sections by headlines (”Q-table”, ”Material-7386:”, etc.). Each section contains one or several lines with a key-word (like ”Rhonorm”, ”[7386]_A[1]”), a ”=” sign, and some value. Lines can be commented out by a % or # at the beginning. Furthermore, the sections in a parameter file are grouped into two parts. The first part contains the settings for the FEOS table generation tool, the second part those for the SHOWEOS table visualization tool.

5 Package Standard Units

The code-internal units of the whole FEOS package are fixed to cgs-units, except for the temperature which is given in eV. Thus, the following units are used:

- Temperature: eV
- Pressure: dyne/cm^2 or erg/cm^3
- Specific internal energy: erg/g
- Specific Helmholtz free energy: erg/g
- Specific entropy: erg/eVg
- Charge state: multiple of the unit electron charge

The FEOS Library

6 The Physics behind the QEOS Model

The FEOS library contains all the routines which are required to calculate an EOS as a function of density and temperature. Before going deeper into the technical details, this section gives a short overview of the physics of the QEOS model and the improvements which have been applied to the FEOS library with regard to the original MPQeos code. In order to understand all settings of the FEOS library and of the tools, it is inevitable to understand the basical physics behind them. Users which are interested in really all details of the model are referred to Ref. [2, 5].

6.1 General Facts

In the QEOS model the specific Helmholtz free energy $F = E - TS$ is composed of three contributions, an electronic part, an ionic part, and the phenomenological bonding correction:

$$F(\rho, T) = F_e(\rho, T) + F_i(\rho, T) + F_b(\rho, T) \tag{1}$$

In this documentation, all three components of F are assumed to be a function of a single temperature T. However, the electronic part does not depend on the ionic part and in principle there is no difficulty in separating the ion and the electron temperatures with the FEOS library. The thermodynamical quantities like pressure p, specific internal energy E, and specific entropy S are derived from the specific Helmholtz free energy:

$$p_{e,i,b} = \rho^2 \frac{\partial F_{e,i,b}}{\partial \rho}, \quad S_{e,i,b} = -\frac{\partial F_{e,i,b}}{\partial T}, \quad E_{e,i,b} = F_{e,i,b} + TS_{e,i,b} \tag{2}$$

Besides these quantities, the charge state Q is calculated. In the following subsections the three parts of the QEOS model will be described shortly.

To calculate an EOS for any particular element or mixture, the material composition must be specified: atomic number $Z[i]$, atomic weight $A[i]$, and number of atoms $X[i]$ per "molecule" of each single element i. Furthermore, the QEOS model requires two empirical parameters to calibrate the generated EOS, namely, the density ρ_o and the bulk modulus $K_o = \rho \, (\partial p / \partial \rho)$ at a certain reference value of temperature T_o and zero pressure $p = p_o = 0$. For substances that are in the solid state at normal conditions, usually the values $T_o \approx 300$ K are used. For details on the calibration, see Subsection 6.3.

The original MPQeos code was designed to calculate an EOS within the following range:

$$10^{-7} \leq \frac{\rho}{\rho_o} \leq 10^6, \qquad 10^{-4}\ \mathrm{eV} \leq T \leq 10^6\ \mathrm{eV}. \tag{3}$$

In the FEOS library a hard lower limit (numerical floor) was introduced due to numerical limitations and is currently set to (see constants "T_ZERO" and "RHO_ZERO" in "LIB-00_DEFINITS.H"):

$$\rho_{min} = 10^{-50}\frac{\mathrm{g}}{\mathrm{cm}^3}, \qquad T_{min} = 10^{-4}\ \mathrm{eV}. \tag{4}$$

One important feature of the QEOS model is the description of van-der-Waals loops [6] in the liquid-vapor region. This means that liquid-vapor phase coexistence can be described by applying a Maxwell construction. More information about the corresponding procedure can be found in Section 6.6.

6.2 Electronic EOS: TF Model

For the calculation of the electronic contribution of a single element – the most important part of an EOS for hot dense matter – the simple Thomas-Fermi (TF) model [7, 8] is used. In this model the electrons are described as a Fermi gas in the self-consistent electrostatic field of the atom which is produced by the ion mesh and the electrons themselves. Matter is segmented into spherical cells for which the equilibrium electron distribution is calculated by solving the TF equation. For the segmentation Wigner-Seitz cells [9] are used. Hence, the radius r_0 of the cells with atomic mass A, density ρ, and proton mass M_p is choosen in the following way:

$$4\pi r_0^3/3 = AM_p/\rho \tag{5}$$

Since the electrons are assumed to be a Fermi gas in the electrostatic field of the electrons and the ions, there is no distinction between valence electrons and electrons from the inner shells. Instead of this they can be seperated in localized and non-localized electrons. If the electrostatic potential $V(r)$ is normalized to disappear at the cell boundary, those electrons are localized which have negative total energy $\epsilon = p^2/2m_e - eV(r)$ with the classical momentum p. For the free atom for $T = 0$ this is the case for all electrons. Electrons with positive energy are called non-localized. They are responsible for the electron pressure and ionization effects.

TF equation for T = 0

For $T = 0$ [10] the TF equation is:

$$\frac{d^2\chi(x)}{dx^2} = \frac{1}{x^{1/2}}\chi^{3/2} \tag{6}$$

It is a dimensionless function where $x = r/a_0$, and the function χ is defined by the potential $V(r)$ and the chemical potential μ. μ is determined by the condition of charge neutrality of the cell. $Z\chi(r)$ can be regarded as the effective nuclear charge.

$$eV(r) + \mu \equiv \frac{Ze^2}{r}\chi(r) \tag{7}$$

$$a_0 = \frac{1}{2}\left(\frac{3\pi}{4}\right)^{2/3}\frac{\hbar^2}{me^2}Z^{-1/3} \tag{8}$$

The boundary conditions for the TF equation are:

$$\chi(x) \to 1, \quad x \to 0 \tag{9}$$

$$\left[\frac{x}{\chi(x)}\frac{d\chi}{dx}\right]_{x=x_0} = 1 \tag{10}$$

The first boundary condition is a result of the divergence of the potential close to the nucleus. The second condition follows from the charge neutrality of the cell. With other words: the electrical field $E_r(r) = -(\partial\phi/\partial r)$ at the boundary of the neutral cell x_0 must disappear in the spherical case because of the law of Gauß:

$$\left[\frac{d\phi}{dx}\right]_{x=x_0} = 0 \quad \Leftrightarrow \quad \left[\frac{x}{\chi(x)}\frac{d\chi}{dx}\right]_{x=x_0} = 1 \tag{11}$$

TF equation for finite temperatures

For finite temperatures Feynman, Metropolis and Teller [9] derived a similar version of the TF equation

$$\frac{d^2\Psi(\xi)}{d\xi^2} = a\xi F_{1/2}\left[\frac{\Psi(\xi)}{\xi}\right] \tag{12}$$

with the following definitions:

$$\xi = \frac{r}{r_0} \tag{13}$$

$$\Psi(\xi) = \xi \frac{eV + \mu}{kT} \tag{14}$$

$$a = \frac{4\pi m_e e^2 r_0^2 (2m_e kT)^{1/2}}{\pi\hbar^3} \tag{15}$$

r_0 is the cell radius, and $F_{1/2}$ denotes the Fermi-Dirac integral. The boundary conditions for finite T are similar to those for $T = 0$:

$$\Psi(0) = \frac{Ze^2}{kTr_0} \tag{16}$$

$$\Psi'(1) = \Psi(1) \tag{17}$$

Thermodynamic variables

By solving the TF equation one obtains the equilibrium electron distribution in the cell and thereby the energy of the electrons which consists of three contributions: kinetic energy K, potential energy of the electrons among each other U_{ee}, and potential energy of the electrons with the nucleus U_{en}:

$$E_e^{tot} = K + U_{en} + U_{ee} \tag{18}$$

The Helmholtz free energy $F_e = E_e - TS_e$ is obtained by integrating the Gibbs-Helmholtz relation [11]:

$$E_e^{tot} = \frac{\partial}{\partial\beta}\left[\beta F_e\right], \quad \beta = 1/kT \tag{19}$$

The charge state depends on the number of electrons in non-localized states ($g(r,p)$ denotes the Fermi distribution function):

$$Q = \int d^3r \int_{\epsilon>0} \frac{2d^3p}{h^3} g(r,p) \tag{20}$$

Advantages / disadvantages of the simple TF model

The most important advantage of the simple TF model is the fact that calculations are faster than with advanced TF theories because the TF equation and all thermodynamical quantities scale with the atomic number and hence must be calculated only once for e.g. hydrogen. For example the internal energy E of a element (A, Z) at density ρ and temperature T shall be calculated. Then one first has to rescale ρ and T:

$$\rho_1 = \rho/AZ \quad T_1 = t/Z^{4/3} \tag{21}$$

Now the internal energy E_1 is calculated for ρ_1 and T_1 for hydrogen $A = Z = 1$. E is obtained by the proper scaling formula:

$$E(\rho, T) = \frac{Z^{7/3}}{A} E_1(\rho_1, T_1) \tag{22}$$

For all other thermodynamical quantities similar formulas exist. The only restraint for the use of the scaling property is that the original TF table $(A = Z = 1)$ must cover a large density-temperature area which becomes clear when looking at equation (21).

The most important disadvantage of the simple TF model is the negligence of attractive (bonding) forces between neutral atoms. This is the reason for an overestimation of the critical pressure and the critical temperature and an overall overestimation of pressures near normal conditions. The bonding forces originate from quantum effects in the electron-electron interaction. There exist extended TF theories like the Thomas-Fermi-Dirac (TFD) theory [12], the Thomas-Fermi-Kirzhnitz (TSK) theory [13] and the quantum statistical model (QSM) [14, 15]. Unfortunately, they are computationally more intensive and do not contain the scaling property of the simple TF theory.

For more details about the TF theory, its limiting cases, and inter-/extrapolation methods in the QEOS model the reader ist referred to Ref. [3, 2].

6.3 Semiempirical Bonding Correction

The semiempirical bonding correction, the recalibration scheme of QEOS, is added to the total EOS in order to improve the previously mentioned failures of the Thomas-Fermi EOS not to take care of exchange forces and to overestimate the electronic pressure in the solid body. A fully quantum mechanical treatment of the electrons in the solid body would be too elaborate within the framework of the QEOS model. The free energy of the bonding correction is:

$$F_b = E_0 \left\{ 1 - \exp\left(b \left[1 - (\rho_o/\rho)^{1/3} \right] \right) \right\} \tag{23}$$

Since F_b does not depend on the temperature, it follows $E_b = F_b$. The pressure then is:

$$p_b = \rho^2 \frac{\partial E_b}{\partial \rho} = -\left(\frac{E_0 b \rho_o}{3}\right)\left(\frac{\rho}{\rho_o}\right)^{2/3} \exp\left(b\left[1 - \left(\frac{\rho_o}{\rho}\right)^{1/3}\right]\right) \tag{24}$$

The constants E_0 and b determine the bonding correction. They characterize the range and the magnitude of the bonding forces. In order to determine them, two conditions must be fulfilled:

1. The total pressure $p_o = p_{io} + p_{eo} + p_{bo}$ must vanish at the user-defined reference conditions (ρ_o,T_o). For substances that are in the solid state at normal conditions, usually the values $T_o \approx 300$ K are used.

2. The value of the bulk modulus [16] (and thereby the sound speed)

$$K_o = \rho \left(\frac{\partial p}{\partial \rho}\right)_{\rho_o} \tag{25}$$

   at the reference conditions must be equal to the user-defined value. A relatively small difference between the isothermal and isentropic bulk moduli is usually ignored.

The theoretical motivation for the form of the Helmholtz free energy F_b [17] is the Morse potential in a two-atomic molecule with the bonding energy D, the equilibrium distance R_e, and a constant b which must be determined empirically:

$$U_M\left(R\right) = D\left[e^{-2b(R-R_e)} - 2e^{-b(R-R_e)}\right] \tag{26}$$

The bonding correction has the strongest effect on the total EOS near the solid density. At higher densities the TF electronic contribution dominates.

6.4 Ionic EOS: Cowan Model

The thermodynamical properties of the ionic contribution which in the QEOS model is totally independent of the electronic contribution are described through the ionic EOS. The contribution of the ions to the total EOS is significant for temperatures $T < 10$ eV and densities near the reference (solid) density $\rho/\rho_o < 2$. For higher temperatures and/or densities the electronic and bonding contributions dominate. In QEOS the Cowan model is used.

It interpolates between known limiting thermodynamical cases by the aid of empirical formulas. The energy in the model is of purely thermal nature. Coulomb and bonding energies are taken care of in the electronic and the bonding contributions. The following list gives an overview of the limiting physical cases of the Cowan model:

- Ideal gas law [12] (high temperatures and/or low densities):

  $$\begin{aligned} p_i &= \rho kT/AM_p, \quad E_i = \frac{3}{2}kT/AM_p, \\ S_i &= k\left[S_0 + \frac{3}{2}log\left(kT/\rho^{2/3}\right)\right]/AM_p \end{aligned} \tag{27}$$

  S_0 is given through the Sackur-Tetrode formula:

  $$S_0 = \frac{5}{2} + log\left(2AM_p\right) - \frac{3}{2}log\left(h^2/2\pi AM_p\right) \tag{28}$$

- Melting scaling law [18] (energy and density for non-ideal dense liquids):

  $$p_i = \frac{\rho kT}{AM_p}\left[1 + \gamma_F\left(\rho\right) f\left(\frac{T_m}{T}\right)\right] \tag{29}$$

  $$E_i = \frac{3}{2}kT/AM_p\left[1 + f\left(\frac{T_m}{T}\right)\right] \tag{30}$$

  γ_F is determined by the melting temperature $T_m(\rho)$ and $f(T_m/T)$ is a scaling function.

- Lindemann melting law:

  $$T_m\left(\rho\right)/\Theta_D^2\left(\rho\right) = \alpha/\rho^{2/3} \tag{31}$$

  Θ_D – Debye temperature, α – material dependent constant

- Dulong-Petit law ($\Theta_D(\rho) \leq T \leq T_m(\rho)$):

  $$E_i \approx 3kT/AM_p \tag{32}$$

- Grüneisen EOS [12] ($T < T_m(\rho)$):

  $$p_i = \Gamma\left(\rho\right)\rho E_i \tag{33}$$

  Dependence of the Grüneisen parameter Γ on Θ_D [12]:

  $$\Gamma\left(\rho\right) = -\frac{V}{\Theta_D}\frac{\partial\Theta_D}{\partial V} = \frac{\partial\log\Theta_D}{\partial\log\rho} \tag{34}$$

- Debye-Modell:

  $$h\nu_D = k\Theta_D \tag{35}$$

  ν_D – Debye frequency, Θ_D – Debye temperature

  $\Rightarrow$ Specific heat of non-conductors for $T < \Theta_D$:

  $$c_V = \frac{12\pi^4 R}{5}\left(\frac{T}{\Theta_D}\right)^3 \tag{36}$$

- Third law of thermodynamics (Nernst's law):

  $$\lim_{T\to 0} S\left(\rho, T\right) = 0 \tag{37}$$

The Cowan model consists of two independent parts. The empirical part of the model makes estimations for the Debye and the melting temperatures depending on the density. In the structural part the EOS is calculated. Therefore, the scaling variables u and w are introduced:

$$u = \Theta_D\left(\rho\right)/T \tag{38}$$

$$w = T_m\left(\rho\right)/T \tag{39}$$

Cowan defines a scaling function $f(u,w)$ which determines the Helmholtz free energy of the ions:

$$F_i\left(\rho, T\right) = \frac{kT}{AM_p} f\left(u, w\right) \tag{40}$$

There are three different areas in u-w space for which the scaling function is defined. For temperatures above the melting temperature ($T > T_m$) the following empirical formula is used:

$$f\left(u, w\right) = -\frac{11}{2} + \frac{9}{2}w^{1/3} + \frac{3}{2}log\left(\frac{u^2}{w}\right), \quad w \leq 1 \tag{41}$$

For the "hot" solid body ($T_m > T > 3\Theta_D$) holds:

$$f\left(u\right) = -1 + 3\log u + \left(3u^2/40 - u^4/2240\right), \quad w > 1, u < 3 \tag{42}$$

For the "cold" solid body ($T \leq 3\Theta_D$) applies:

$$\begin{aligned} f\left(u\right) =& \frac{9}{8}u + 3\log\left(1 - e^{-u}\right) - \frac{\pi^4}{5u^3} \\ &+ e^{-u}\left(3 + 9u^{-1} + 18u^{-2} + 18u^{-3}\right), \quad w > 1, u \geq 3 \end{aligned} \tag{43}$$

In the Cowan model the Gibbs free energies in the liquid and in the solid phase on the melting curve are exactly the same [3]. This means that the QEOS model does not contain melting.

The task of the empirical part of the Cowan model is to make the Debye and melting temperatures and the Grüneisen parameter available to the structural part as functions of the density. It has to be pointed out that a failure in the estimation of these quantities only has little influence on the pressure and the energy of the ionic EOS. The empirical part should fulfill the formulas for the Grüneisen parameter (34) and the Lindemann melting law (31).

Using a reference density $\rho_{ref} = \left(A/9Z^{0.3}\right)$ g/cm^3 the densities are scaled with $\xi = \rho/\rho_{ref}$. Thereby, the reference density corresponds with a atomic radius of $R_a \approx 1,5 \cdot 10^{-8} Z^{0.1}$ cm. Then Cowan's estimations are:

$$kT_m = 0.32 \frac{\xi^{2b+10/3}}{(1+\xi)^4} \text{ [eV]} \tag{44}$$

$$k\Theta_D = \frac{1,68}{Z+22} \frac{\xi^{b+2}}{(1+\xi)^2} \text{ [eV]} \tag{45}$$

$$\Gamma = b + \frac{2}{1+\xi} \qquad b = 0.6Z^{1/9} \tag{46}$$

Together with $\alpha = 0.0262 \left(A^{2/3} Z^{0.2}\right) (Z+22)^2$ the equations (31) and (34) are fulfilled. Since the formulas of the empirical model are simple, it is clear that no exact estimations can be done, and the calculated quantities have only a qualitative character.

6.5 Homogeneous Mixtures of Elements

Besides pure elements, the QEOS model includes the possibility to calculate the EOS of homogenous mixtures of elements. Unfortunately, the corresponding procedure described in [2] was not implemented in version 2.0 of MPQeos. Hence, the so-called TF-mixing-of-elements method was added to the FEOS library. The ionic contribution and the bonding correction are handled as a single species with mean atomic number $\bar{Z}$ and weight $\bar{A}$. For the electronic contribution of the mixture the densities $\rho[i]$ (effectively the partial volumes) of all species i are iteratively adjusted in order to equilibrate all TF pressures $p_e[i]$ and to fulfill an additive volume rule:

$$1)\, p_e[i]\,(\rho[i],T) = p_e \ \forall i, \qquad 2)\, \frac{\bar{A}}{\rho} = \sum_i x[i]\frac{A[i]}{\rho[i]} \quad (\bar{A} = \sum_i x[i]A[i])\,. \ (47)$$

In these equations $x[i] = X[i]/\sum_i X[i]$ is the number fraction, and $A[i]$ is the atomic weight of species i.

The thermodynamic values for the electronic component of the mixture are finally obtained by summing up the single element values (with densities obtained by the above scheme), each weighted by $x[i]A[i]/\bar{A}$.

The described procedure must be called for every density and temperature, and therefore it is clear that the calculation of mixtures of elements is computationally more intensive than for a single element. Actually, in some few cases the iteration process does not converge. In this case the user is recommended to start the calculation again or to slightly change the density and/or temperature.

6.6 Liquid-Vapor Phase Coexistence

The QEOS model describes van-der-Waals loops [6] (liquid-vapor metastable states) on isotherms below the critical point. Liquid-vapor phase coexistence can be described through a fully equilibrium EOS which is obtained by a Maxwell construction eliminating the van-der-Waals loops. Figure 2 shows a metastable isotherm with a van-der-Waals loop (solid line) as well as the corresponding equilibrium isotherm with saturated vapor / equilibrium pressure p_{sat} (dashed line). The region of liquid-vapor metastable and equilibrium states is delimited by the so-called binodal / vaporization curve. The extrema of the van-der-Waals loops are connected through the so-called spinodal. The question whether one has to use the metastable or the equilibrium EOS below the binodal must be individually answered for each problem under consideration. Ref. [4] gives an answer for hydrodynamic simulations and shows results for volumetrically heated matter.

According to Maxwell's rule the saturated vapor pressures for each isotherm below the critical point are determined by finding such two points along the isotherm with equal pressure $p = p_{sat}$ and equal Gibbs free energy $G = E + pV - TS$ (chemical potential). This rule corresponds to the well known geometrical rule of equal areas between the van-der-Waals loop and

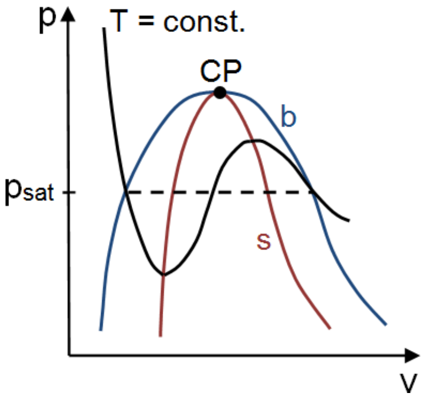


Figure 2: Isotherm with and without the Maxwell construction on the volume-pressure phase plane; the binodal and the spinodal touch each other at the critical point CP.

the equilibrium isotherm:

$$(T = const.) \int_{V_{vap}}^{V_{liq}} pdV = p_{sat} \left(V_{vap} - V_{liq} \right) . \tag{48}$$

In the original MPQeos code the Maxwell construction was calculated with this geometrical rule which is especially for low temperatures in the two-phase region computationally very intensive and imprecise. In FEOS the MPQeos method was replaced by a root finding algorithm which determines the points on the liquid and the vapor sides of the binodal with equal Gibbs' free energies and pressures. Since this "new" method involves no numerical integrations, it is much faster and more accurate.

Nevertheless, there exists one important limitation for the calculation of the Maxwell construction: Since p_{sat} and the density at the vapor branch of the binodal become very low for temperatures around room temperature, the EOS has to be calculated down to very low densities. As already mentioned in Subsection 6.1, computational limits determine the lowest calculatable density. The Maxwell routines of the library will print out warnings if the Maxwell construction cannot be done consistently for a given temperature. Furthermore, the lowest temperature which could be calculated consistently is returned by the library. Note that even if the Maxwell construction can be done consistently one should be carefull to check the correct behaviour of the EOS for such very low densities. In some cases, it can happen that the code violates the ideal gas law limit. This is a construction site for the future.

If the user decides to calculate the Maxwell equilibrium EOS, the library applies two additional steps: In the first step, the critical point is looked up via a bisectioning algorithm. A check is done on loops in each isotherm, until the one with loops at maximum temperature is found. On this isotherm, the density where $d^2p/d\rho^2 = 0$ is determined. In the second step, the vaporisation curve is calculated by applying the Maxwell construction to each temperature specified by the user below the critical point. In order to save computational time, both steps are performed during the material initialization stage.

Later, for computing EOS data at some point inside the two-phase region, the thermodynamic quantities are interpolated linearly. Outside the two-phase region, the computation of the thermodynamic quantities goes as in the normal case. As the user specified temperature grid for the Maxwell construction strongly determines the quality or even the existence of the critical data and vaporisation curve, one should not specify too few temperatures near and inside the critical region.

6.7 Cold Curve Improvement

Despite the existence of the bonding correction (Section 6.3), the QEOS model still overestimates the location of the critical point (pressure p_c and temperature T_c). Furthermore, in a few cases the value of the cohesive energy E_{coh} – better known as enthalpy of sublimation – can become negative. In order to solve this problem in the FEOS library, the TF cold curve and the bonding correction can now be replaced for densities $\rho < \rho_o$ by a soft-sphere function which was proposed by Young et al. [5]:

$$E_{cold}\,(\rho, T = 0) = A\rho^n - B\rho^m + E_{coh}\,. \tag{49}$$

The constants A and B are adjusted so as to make the total pressure and the internal energy be equal to zero at the reference point (ρ_o, T_o):

$$p(\rho_o, T_o) = E(\rho_o, T_o) = 0\,. \tag{50}$$

The free parameters m and n are used to improve the agreement with the experimentally (or theoretically) known critical point. At $\rho = \rho_o$ the sound velocity is allowed to be discontinuous.

The following list demonstrates how the soft-sphere function affects the location of the critical point for aluminum:

Experimental / theoretical values [5]:	$T_c = 5700\,\mathrm{K}$,	$p_c = 1820\,\mathrm{bar}$
Original MPQeos (version 2.0):	$T_c = 13487\,\mathrm{K}$,	$p_c = 23487\,\mathrm{bar}$
FEOS library:	$T_c = 5558\,\mathrm{K}$,	$p_c = 1722\,\mathrm{bar}$
	($m = 0.5$, $n = 2.0$,	$E_{coh} = 12.123\,\mathrm{kJ/g}$)

7 Technical Design of the Library

The static library file "libfeos.a" is basically a collection of object files / routines. Those routines which are intended to be used from outside the library, namely from the user's code or the FEOS table generation tool, form the FEOS library interface. The interface is designed in such a way that all routines can be either called from a C, a C++, or a Fortran code. Instructions concerning the implementation of the library file into the user's code and detailed descriptions of the interface routines are given in Sections 11 and 12.

In principle, the FEOS library can calculate the EOS of an arbitrary number of materials with fixed composition in parallel. The work done by the library routines can be classified into two categories: 1) routines which are responsible for the initialization of the library and of the materials which shall be calculated, and 2) routines which calculate the thermodynamic functions. Analogously, also the interface routines can be basically classified into two categories: 1) the obligatory routines which are responsible for the initialization and finalization, 2) the facultative routines which return the information calculated during the initialization stage and the thermodynamic functions (Section 12).

Figure 3 gives an overview of the logical structure how the library is intended to be implemented into a user's code for three different materials. First, the library itself is initialized with an arbitrary number of material entities. Then, all materials must be initialized with several parameters by the material initialization procedure described in the next subsection. Having initialized a material, one can either retrieve the information about the material which are already present and/or one can calculate the thermodynamic functions (of the full EOS and/or of the EOS contributions with or without a Maxwell construction) as a function of density and temperature (see Section 8 for a list of information provided by the library). Finally, if the library is not to be used anymore, it must be finalized to free the occupied memory.

7.1 Material Initialization Stage

Figure 4 gives an overview of the initialization procedure of a material within the FEOS library. The material initialization is controlled by the interface routine "FEOS_Init_Mat(...)". Several arguments have to be passed which

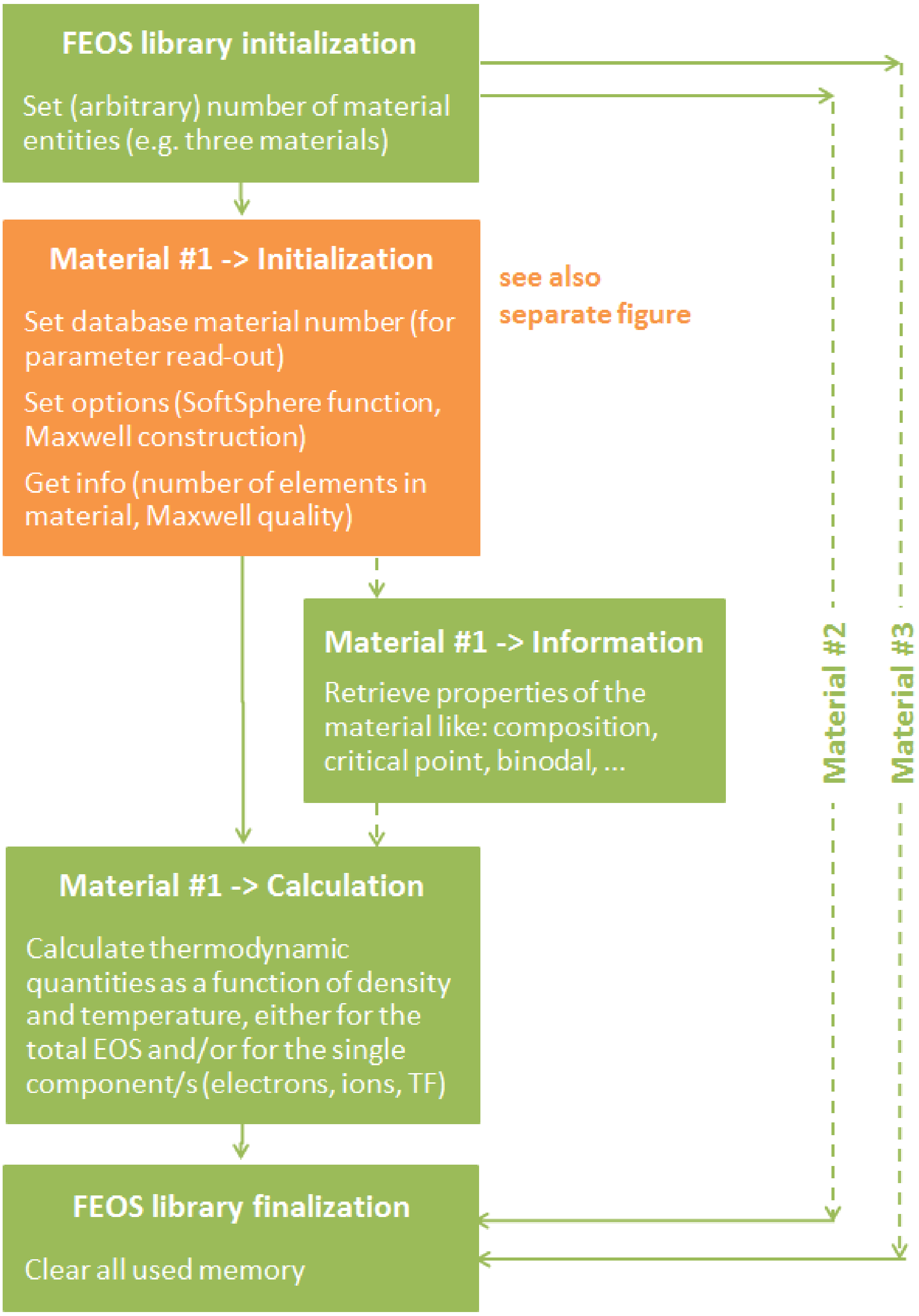
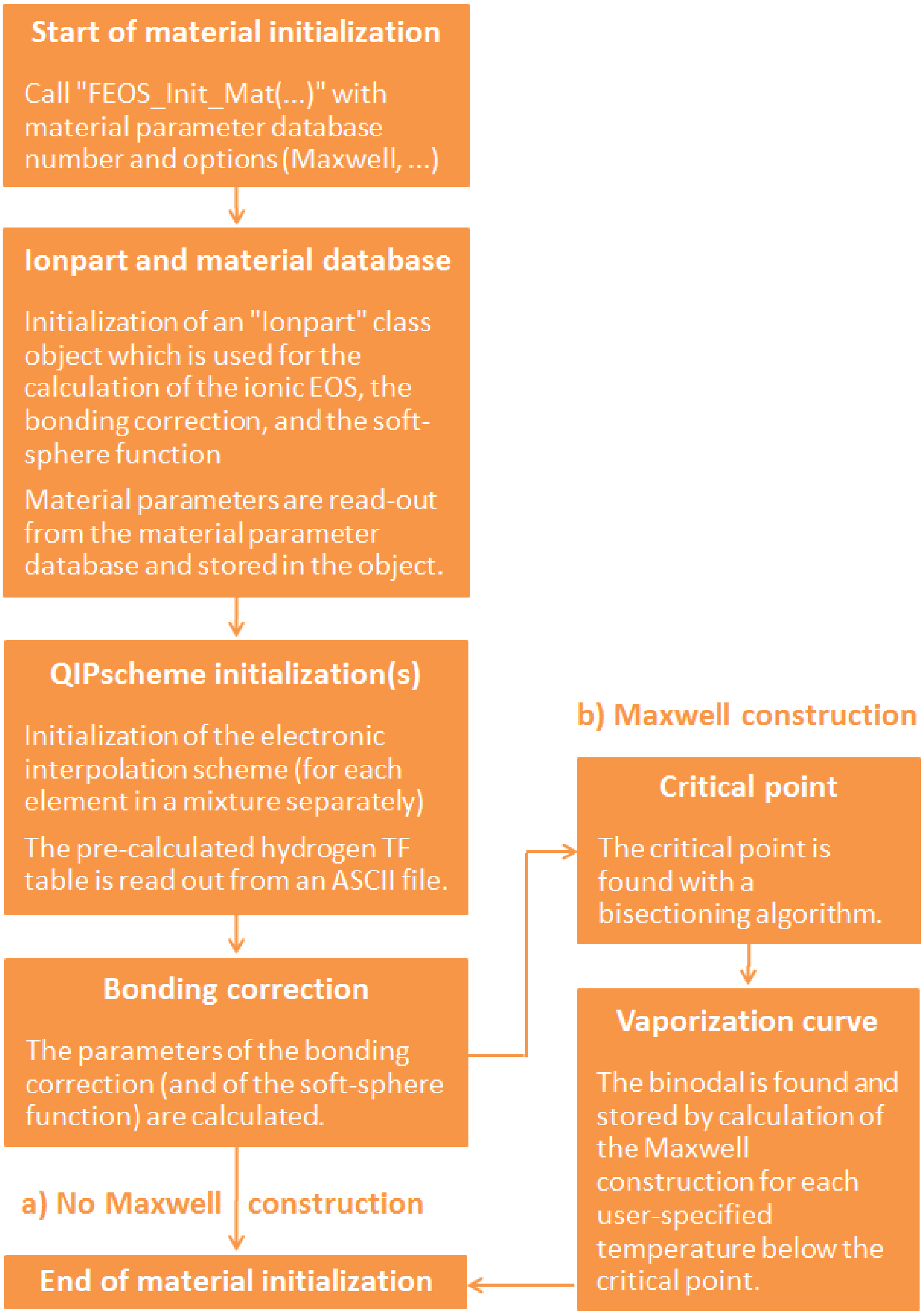

FEOS library initialization
Set (arbitrary) number of material entities (e.g. three materials)
Material #1 -> Initialization
Set database material number (for parameter read-out)
Set options (SoftSphere function, Maxwell construction)
Get info (number of elements in material, Maxwell quality)
see also separate figure
Material #1 -> Information
Retrieve properties of the material like: composition, critical point, binodal, ...
Material #1 -> Calculation
Calculate thermodynamic quantities as a function of density and temperature, either for the total EOS and/or for the single component/s (electrons, ions, TF)
FEOS library finalization
Clear all used memory
Material #2
Material #3

Figure 3: Operational sequence for implementation of the FEOS library.



Start of material initialization

Call "FEOS_Init_Mat(...)" with material parameter database number and options (Maxwell, ...)

Ionpart and material database

Initialization of an "Ionpart" class object which is used for the calculation of the ionic EOS, the bonding correction, and the soft-sphere function

Material parameters are read-out from the material parameter database and stored in the object.

QIPscheme initialization(s)

Initialization of the electronic interpolation scheme (for each element in a mixture separately)

The pre-calculated hydrogen TF table is read out from an ASCII file.

Bonding correction

The parameters of the bonding correction (and of the soft-sphere function) are calculated.

Critical point

The critical point is found with a bisectioning algorithm.

Vaporization curve

The binodal is found and stored by calculation of the Maxwell construction for each user-specified temperature below the critical point.

End of material initialization

Figure 4: Operational sequence of the FEOS library material initialization.

control the following options:

- The number of the material in the FEOS material parameter database file (Section 10) must be defined.
- A flag tells the library if a Maxwell construction shall be initialized for the material. If set to yes, an array with temperatures and its size must be passed. These temperatures control the quality of the Maxwell construction.
- Another flag tells the library if the improved cold curve (Section 6.7) shall be used for the material.

Within the several steps of the initialization procedure the different class objects of the FEOS library are constructed for the material. The FEOS library knows basically three classes: 1) a class "Ionpart" for the storage of the general material parameters and for the routines used for the calculation of the ionic contribution, the bonding correction, and the improved cold curve, 2) a class "QIPscheme" for the electronic contribution / interpolation on the TF table for each single element in a material, and 3) a class "CriticalData" for the Maxwell construction data and routines.

First, the "Ionpart" class object is constructed. In this step the material parameters are read-out from the material parameter database file (Section 10). Then, "QIPscheme" class objects are constructed for each element in the material. Therefore, the precalculated TF table file for hydrogen is read-out and the TF interpolation scheme is set-up for the corresponding atomic numbers and atomic weights. In the third step, the parameters of the bonding correction (and soft sphere function) are determined.

If a Maxwell construction (Section 6.6) is desired, a "CriticalData" class object is constructed and at the same time filled with data. Therefore, first the critical point is determined with a bisectioning procedure. Then, for each defined temperature below the critical point, the Maxwell construction is iteratively calculated and the binodal / vaporization curve is stored. Later on, when calculating a density-temperature point within the two-phase region, this precalculated critical data can be used for linear interpolations.

At the end of a material initialization the routine "FEOS_Init_Mat(...)" returns the number of elements in the choosen material and, if available, the quality of the calculated Maxwell construction data.

8 Information provided by the Library

The FEOS library provides a wide set of information. The following list summarizes all the properties and parameters which can be retrieved with the library interface routines:

- The following thermodynamic functions as a function of density and temperature for the complete EOS and/or for one component (electronic EOS, ionic EOS, or pure Thomas-Fermi EOS), either with or without a Maxwell construction:
  - Pressure
  - Specific internal energy
  - Specific Helmholtz free energy
  - Specific entropy
  - Charge state for each element in a mixture
  - Summed-up (not mean) charge state for a ”molecule”
- All the material parameters which are stored in the material parameter database (Section 10), especially the material's composition
- If initialized, critical point data: temperature, density, pressure, enthalpy, entropy, and compressibility factor at the critical point
- If initialized: temperatures, densities, pressures, specific Gibbs free energies, specific enthalpies, and compressibility factors along the binodal
- If initialized: temperatures, densities, and pressures along the spinodal
- The temperature and density floor values of the library, below which the EOS cannot be calculated for numerical reasons (Section 6.1)
- The energy offsets of the electronic and the ionic EOS which are substracted to zero the complete specific energy at reference conditions

9 Source Code Structure

As the QEOS model itself, also the source code of the FEOS library is divided into several parts. Table 2 gives an overview of the source files of the library.

LIB-00_DEFINITS.H	Numerical constants for the library and paths of the material database and the Thomas-Fermi table files
LIB-01_TF_SERVICES.C	Routines for reading the TF-file and calculating the Fermi-Dirac functions
LIB-02_TF_TABLE_1.C, LIB-02_TF_TABLE_2.C	Routines for calculation of the single element Thomas-Fermi EOS
LIB-03_TF_MIXTURE.C	Routines for calculation of the Thomas-Fermi EOS of a mixture
LIB-04_IONMOD.C	Routines for reading the material parameter database and calculation of the ionic EOS, the bonding correction, and the soft-sphere function
LIB-05_EOS_SERVICES.C	Routines for calculation of the thermodynamic functions without Maxwell construction
LIB-06_MAXWELL.C	Routines for initialization of the Maxwell construction and calculation of the EOS with Maxwell construction
LIB-07_C_INTERFACE.C, LIB-08_FORTRAN_INTERFACE.C	C/C++ and Fortran interface routines

Table 2: Source files of the FEOS library.

The different classes of the code were already explained in Section 7.1. The ”QIPscheme” class for the calculation of the Thomas-Fermi EOS for a single element is defined in the ”LIB-02...” files. The Thomas-Fermi EOS for a mixture of elements (Section 6.5) is obtained in the ”LIB-03...” file. The ”Ionpart” class is defined in the ”LIB-04...” file.

The ”LIB-05...” file combines the different parts of the QEOS model and contains all the routines which calculate the thermodynamic functions without a Maxwell construction. These routines may be called by the ”CriticalData” class, defined in the ”LIB-06...” file, to initialize and to calculate the thermodynamic functions with a Maxwell construction. Finally, the interface routines are defined in the "LIB-07..." and the "LIB-08..." files where the Fortran interface routines simply call the C/C++ interface routines.

10 Material Parameter Database

The material parameter database file "FEOS_Material-DB.dat" contains all collected physical material properties and parameters. The FEOS library routines access this file for each material to be initialized. Each parameter set / section in the file begins with the headline "Material-xxxx:" where xxxx is the material number. Allowed material numbers are 1000-9999. Each parameter inside a section begins with the prefix "[xxxx]_". Table 3 lists all parameters which must be set for a material. The soft-sphere parameters must be only set if the improved cold curve (see Section 6.7) shall be used.

11 Implementation of the Library

The static library file "libfeos.a" is automatically created within the FEOS package installation / compilation process (Section 3). It can be implemented into any C/C++ or Fortran code.

If the library accessing code is compiled with the GNU or Intel C++ compiler, the following options must be applied for the linking stage:

-L. -lfeos   or   libfeos.a

For the GNU or Intel C or Fortran compiler, the corresponding options are:

-L. -lstdc++ -lfeos   or   -lstdc++ libfeos.a

The -L. flag prompts the compiler to search for librariy files in the same directory where it is executed. The -lstdc++ flag tells the compiler to employ the C++ library, and the -lfeos flag finally enables the FEOS library.

If the library shall be used with a C/C++ code, the "libfeos.h" header file must be also copied to the code directory, and the "include "libfeos.h"" statement must be used inside the code.

Once a code is successfully linked together with the FEOS library and the executable file shall be used, one must be careful also to copy the files

SESAME-Number	Number of the material in the SESAME database. If not known, this parameter can be set to 1000. It is only required for the SESAME output format (Section 16.3).
Treference	Reference temperature in eV
Rhoreference	Density at Treference and zero pressure in g/cm^3
Bulk-Modulus	Bulk modulus at Treference and Rhoreference in dyne/cm^2
Number-of-Elements	Number of elements which form a mixture. For single elements this parameter must be set to 1
A[1...Element-Number]	Atomic weights of all included elements
Z[1...Element-Number]	Atomic numbers of all included elements
X[1...Element-Number]	Number of atoms in one ”molecule” of all included elements
Ecohesive	Cohesive energy / enthalphy of sublimation in erg/g; used for the soft-sphere function
Soft-Sphere-m	Parameter m in soft-sphere function
Soft-Sphere-n	Parameter n in soft-sphere function

Table 3: Parameters to be set for each material entry in the material parameter database file. A corresponding prefix [xxxx]_ with the material number must be put in front of each parameter name.

”FEOS_Material-DB.dat” and ”FEOS_TF-Table_1197.dat” from /EOS-Data into the directory where the code is executed. The material parameter database and the precalculated Thomas-Fermi table are read during the initialization stage of any material to be calculated with the library. If a material is not present in the database, a new entry must be created (Section 10).

12 Interface Routines

The FEOS library provides several routines for the usage in any user's code. To calculate an EOS, some of these routines are obligatory. The other routines are facultative and can only be called if the obligatory ones are used. To get an impression, how the interface routines may be used, one can also have a look into the main file of the FEOS table generation tool (Section 14).

All interface routines begin with the prefix "FEOS_". Furthermore, all interface routines are present in two versions. One version is intended to be called from C/C++ codes. The corresponding routines will be marked in blue in this section. The other version is intended to be called from Fortran codes, and the corresponding routines will be marked in red in the following. The file "LIB-07_C_INTERFACE.C" contains all the C/C++ interface routines. In "LIB-08_FORTRAN_INTERFACE.C" the Fortran interface routines are located.

Fortran programmers should take care of the fact that most Fortran compilers add automatically a "_" sign at the end of each subroutine's name during linking. The FEOS library assumes that this "_" sign is present. If one of the library Fortran interface routines is called and the Fortran compiler does not automatically add the "_" sign, then one must manually add it (e.g., instead of using "call FEOS_FINALIZE(...)", the command "call FEOS_FINALIZE_(...)" must be used).

If the interface routines are intended to be called from an OpenMP parallelized code, one must assure that any FEOS interface routine is only called by the OpenMP master thread.

Finally, one should always keep in mind, that any physical quantity argument in the interface routines, no matter if input or output, is or must be given in the FEOS standard units (Section 5).

12.1 Obligatory Interface Routines

void FEOS_Initialize(int entitynumber, int printflag)

FEOS_INITIALIZE(entitynumber, printflag)

integer(4) :: entitynumber, printflag

entitynumber (INPUT): Number of different materials which the EOS shall be calculated for. E.g., to calculate the EOS for Al, Cu and Li, entitynumber must be set to 3 (or any higher integer).

printflag (INPUT): If set to 0, the library will write no output to the console (except for error messages). If set to 1, the library will write full output to the console. For MPI parallelized codes this flag should only be 1 for the MPI master task to reduce the output to the console.

FEOS_Initialize must be called before all the other interface routines. It can only be called again, if one calls FEOS_Finalize before.

---

void FEOS_Finalize( )

FEOS_FINALIZE( )

FEOS_Finalize will free the memory used by the library. Once one has called it, except for FEOS_Initialize, the other interface routines cannot be called anymore.

---

void FEOS_Init_Mat(int entity, int materialnumber, int Maxwellflag,
 int softsphereflag, double* MaxwellTtab, int sizeofMaxwellTtab,
 int* numofelements, int* numofMaxwelliso,
 double* MaxwellTreliable)

FEOS_INIT_MAT(entity, materialnumber, Maxwellflag, softsphereflag,
 MaxwellTtab, sizeofMaxwellTtab, numofelements, numofMaxwelliso,
 MaxwellTreliable)

integer(4) :: entity, materialnumber, Maxwellflag, softsphereflag,
 sizeofMaxwellTtab, numofelements, numofMaxwelliso

real(8) :: MaxwellTtab(sizeofMaxwellTtab), MaxwellTreliable

entity (INPUT): This integer defines a unique code-internal identifier for any material which will be calculated. Every number is only allowed to be initialized once. The allowed range is [1, entitynumber] (see FEOS_Initialize).

materialnumber (INPUT): Number of the material in the FEOS material parameter database file.

Maxwellflag (INPUT): If set to 1, the critical point will be calculated, and the Maxwell construction will be initialized for this material entity. If set to 0, no critical point and no Maxwell construction will be available.

softsphereflag (INPUT): If set to 1, the soft-sphere function will be used. If set to 0, only the bonding correction will be used. The soft-sphere function can only be used if the soft-sphere parameters have been defined in the FEOS material parameter database for the material with number "materialnumber".

MaxwellTtab (INPUT): Array of size "sizeofMaxwellTtab" containing a list of temperatures in eV in ascending order. If the Maxwell construction should be initialized, it will be explicitly calculated for those temperatures. The more temperatures are used, the better the quality of the Maxwell data will be, but the longer the initialization will take. If the library is used to calculate the EOS for a fixed table, you the best result will be achieved if the same temperatures of the EOS table are used for the Maxwell initialization. It must be assured that at least one temperature lies above the critial point!

sizeofMaxwellTtab (INPUT): Number of temperatures in or size of the array "MaxwellTtab".

numofelements (OUTPUT): Number of elements within the material with number "materialnumber".

numofMaxwelliso (OUTPUT): If the Maxwell construction is initialized, this integer will give the number of temperatures which lie below the critical point and for which the Maxwell construction was calculated. Attention: This does not imply that these temperatures are the same temperatures within your input array "MaxwellTtab" up to element "numofMaxwelliso".

MaxwellTreliable (OUTPUT): If the Maxwell construction is initialized, this will represent the lowest temperature for which the Maxwell construction could be calculated consistently without any problems. This means, that the EOS with Maxwell construction is reliable above this temperature.

FEOS_Init_Mat must be called until any other interface routine for a material entity can be called. It must be called for any material the EOS should be calculated for (of course, with different entity numbers between 1 and entitynumber). Until FEOS_Init_Mat can be called, the library must have been initialized with FEOS_Initialize before. For the same entity, FEOS_Init_Mat can only be called again if FEOS_Delete_Mat has been called before in order to delete the material entity, or the whole library has been reinitialized with FEOS_Finalize followed by FEOS_Initialize.

12.2 Facultative Interface Routines

All facultative routines can only be used if the library was (re-)initialized with FEOS_Initialize and not finalized with FEOS_Finalize. Some facultative routines are material-specific or return information only for one entity. These routines can only be called if the material entity has been initialized with FEOS_Init_Mat. Furthermore, some facultative routines require Maxwell-construction data. These routines can only be called for a given material entity if "Maxwellflag" was set to 1 in FEOS_Init_Mat.

```
void FEOS_Get_EOS_All( int entity, int Maxwell,
        double Rho, double T, int* belowbinodal,
        double* p, double* e, double* s, double* f, double* q, double* qtot,
        double* pe, double* ee, double* se, double* fe,
        double* pi, double* ei, double* si, double* fi,
        double* pTF, double* eTF, double* sTF, double* fTF )
```

```
FEOS_GET_EOS_ALL( entity, Maxwell, Rho, T, belowbinodal,
        p, e, s, f, q, qtot, pe, ee, se, fe, pi, ei, si, fi, pTF, eTF, sTF, fTF )
```

```
integer(4) :: entity, Maxwell, belowbinodal
```

```
real(8) :: Rho, T, p, e, s, f, q(numofelements), qtot, pe, ee, se, fe,
           pi, ei, si, fi, pTF, eTF, sTF, fTF
```

entity (INPUT): Defines the material entity the EOS will be calculated for.

Maxwell (INPUT): If set to 1, the EOS will be returned with Maxwell construction. If set to 0, the EOS will be calculated without Maxwell construction. Attention: "Maxwell" = 1 is only allowed if the Maxwell construction has been initialized in FEOS_Init_Mat for the entity under consideration ("Maxwellflag" = 1).

Rho (INPUT): Density in g/cm^3 for which the EOS will be calculated.

T (INPUT): Temperature in eV for which the EOS will be calculated.

belowbinodal (OUTPUT): If "Maxwell" is set to 1, this integer will be 1 if the density-temperature point lies inside the two-phase region, and, thus, the Maxwell construction is used. It will be 0 if the density-temperature point lies outside the two-phase region. If "Maxwell" is set to 0, "belowbinodal" will be always 0, even if the temperature-density point lies inside the two-phase region.

p, e, s, f (OUTPUT): Pressure, specific internal energy, specific entropy, and specific Helmholtz free energy of the total EOS (which is the sum of the electronic and the ionic EOS).

q (OUTPUT): Array of size "numofelements" containing the charge states for every element of the material.

qtot (OUTPUT): Charge state of one "molecule". To get the mean charge state of a mixture, divide "qtot" by "Xtot" (see FEOS_Get_Mat_Par).

pe, ee, se, fe (OUTPUT): Pressure, specific internal energy, specific entropy, and specific Helmholtz free energy of the electronic EOS (which is the sum of the pure Thomas-Fermi contribution and the corrections - the bonding correction and the soft-sphere function).

pi, ei, si, fi (OUTPUT): Pressure, specific internal energy, specific entropy, and specific Helmholtz free energy of the ionic EOS (which is calculated with the Cowan model).

pTF, eTF, sTF, fTF (OUTPUT): Pressure, specific internal energy, specific entropy, and specific Helmholtz free energy of the pure Thomas-Fermi EOS.

Always remember: The total pressure in the FEOS model is set to 0 for a given reference point ρ_o and T_o. The specific internal energy and specific Helmholtz free energy are shifted by an offset such that both become 0 for each component (total EOS, electronic EOS, ionic EOS, and pure Thomas-Fermi EOS) at ρ_o and T_o. The ionic and electronic offsets can be retrieved out of the library for each material entity with FEOS_Get_Energy_Offsets.

If the EOS is calculated with a Maxwell construction, the charge states and all the quantities from the electronic, ionic, and pure Thomas-Fermi EOS become physically meaningless inside the two-phase region.

If one only wants to calculate the electronic EOS, the ionic EOS, or the pure Thomas-Fermi EOS, it is highly advisable to use FEOS_Get_EOS instead of FEOS_Get_EOS_All. This will safe computational time, since the unwanted contributions are not calculated.

```
void FEOS_Get_EOS( int entity, int task, int Maxwell,
        double Rho, double T, int* belowbinodal,
        double* p, double* e, double* s, double* f, double* q, double* qtot )
```

```
FEOS_GET_EOS( entity, task, Maxwell, Rho, T, belowbinodal,
        p, e, s, f, q, qtot )
```

```
integer(4) :: entity, task, Maxwell, belowbinodal
```

```
real(8) :: Rho, T, p, e, s, f, q(numofelements), qtot
```

entity (INPUT): Defines the material entity the EOS will be calculated for.

task (INPUT): Defines the kind of EOS which will be returned.
"task" = 0 → The total EOS will be returned.
"task" = 1 → The ionic EOS will be returned.
"task" = 2 → The electronic EOS will be returned.
"task" = 3 → The pure Thomas-Fermi EOS will be returned.
Note that the total EOS is the sum of the electronic and the ionic EOS. The electronic EOS is the pure Thomas-Fermi EOS including all corrections - the bonding correction and the soft-sphere function.

Maxwell (INPUT): If set to 1, the EOS will be returned with Maxwell construction. If set to 0, the EOS will be calculated without Maxwell construction. Attention: "Maxwell" = 1 is only allowed if the Maxwell construction has been initialized in FEOS_Init_Mat for the entity under consideration ("Maxwellflag" = 1).

Rho (INPUT): Density in g/cm^3 for which the EOS will be calculated.

T (INPUT): Temperature in eV for which the EOS will be calculated.

belowbinodal (OUTPUT): If "Maxwell" is set to 1, this integer will be 1 if the density-temperature point lies inside the two-phase region, and, thus, the Maxwell construction is used. It will be 0 if the density-temperature point lies outside the two-phase region. If "Maxwell" is set to 0, "belowbinodal" will be always 0, even if the temperature-density point lies inside the two-phase region.

p, e, s, f (OUTPUT): Pressure, specific internal energy, specific entropy, and specific Helmholtz free energy of the EOS defined by "task".

q (OUTPUT): Array of size "numofelements" containing the charge states for every element of the material. Note that the charge states are only calculated, if "task" is not equal to 1.

qtot (OUTPUT): Charge state of one "molecule". Note that "qtot" is only calculated, if "task" is not equal to 1. To get the mean charge state of a mixture, divide "qtot" by "Xtot" (see FEOS_Get_Mat_Par).

Always remember: The total pressure in the FEOS model is set to 0 for a given reference point ρ_o and T_o. The specific internal energy and specific Helmholtz free energy are shifted by an offset such that both become 0 for each component (total EOS, electronic EOS, ionic EOS, and pure Thomas-Fermi EOS) at ρ_o and T_o. The ionic and electronic offsets can be retrieved out of the library for each material entity with FEOS_Get_Energy_Offsets.

If the EOS is calculated with a Maxwell construction, the charge states and all the quantities from the electronic, ionic, and pure Thomas-Fermi EOS become physically meaningless inside the two-phase region.

---

void FEOS_Delete_Mat(int entity)

FEOS_DELETE_MAT(entity)

integer(4) :: entity

entity (INPUT): Defines the material entity which shall be deleted.

This routine can be used if a given entity shall be deleted during the runtime in order to initialize the same number afterwards again with FEOS_Mat_Init with a different material or different conditions.

---

void FEOS_Get_Calc_Limits(double* Tzero, double* Rhozero)

```
FEOS_GET_CALC_LIMITS( Tzero, Rhozero )
```

```
real(8) :: Tzero, Rhozero
```

Tzero (OUTPUT): Lower temperature floor of the library.

Rhozero (OUTPUT): Lower density floor of the library.

If one calls any interface routine for a density and/or a temperature below these limits, the library will set the density and/or the temperature to these limits.

```
void FEOS_Get_Mat_Par( int entity, double* A, double* Z, double* X,
        double* Atot, double* Ztot, double* Xtot, double* Tref,
        double* Rhoref, double* BulkModref, int* SESAMEnumber )
```

```
FEOS_GET_MAT_PAR( entity, A, Z, X, Atot, Ztot, Xtot, Tref, Rhoref,
        BulkModref, SESAMEnumber )
```

```
integer(4) :: entity, SESAMEnumber
```

```
real(8) :: A(numofelements), Z(numofelements), X(numofelements),
           Atot, Ztot, Xtot, Tref, Rhoref, BulkModref
```

entity (INPUT): Defines the material entity for which the material parameters will be returned.

A (OUTPUT): Array of size "numofelements" containing the atomic weights of all elements.

Z (OUTPUT): Array of size "numofelements" containing the atomic numbers of all elements.

X (OUTPUT): Array of size "numofelements" containing the number of atoms of all elements within one "molecule".

Atot (OUTPUT): Atomic weight of one "molecule". To get the mean atomic weight of a mixture, "Atot" must be divided by "Xtot".

Ztot (OUTPUT): Atomic number of one "molecule". To get the mean atomic number of a mixture, "Ztot" must be divided by "Xtot".

Xtot (OUTPUT): Number of atoms within one "molecule".

Tref, Rhoref (OUTPUT): Reference temperature and density for which the total pressure is set to zero.

BulkModref (OUTPUT): Reference bulk modulus at reference density and reference temperature.

This routine returns all the basic material parameters from the material parameter database.

```
void FEOS_Get_Crit_Point( int entity, double* T, double* Rho,
    double* P, double* H, double* S, double* Z )
```

```
FEOS_GET_CRIT_POINT( entity, T, Rho, P, H, S, Z )

integer(4) :: entity

real(8) :: T, Rho, P, H, S, Z
```

entity (INPUT): Defines the material entity for which the critical point parameters will be returned.

T (OUTPUT): Critical temperature.

Rho (OUTPUT): Critical density.

P (OUTPUT): Critical pressure.

H (OUTPUT): Critical specific enthalpy.

S (OUTPUT): Critical specific entropy.

Z (OUTPUT): Critical compressibility factor.

This routine returns the location of the critical point. Attention: This data is only available if the Maxwell construction was initialized in FEOS_Init_Mat for the entity under consideration ("Maxwellflag" = 1).

```
void FEOS_Get_SoftSphere_Par( int entity, double* Ecoh, double* softsp_n,
```

double* softsp_m, double* softsp_A, double* softsp_B)

FEOS_GET_SOFTSPHERE_PAR(entity, Ecoh, softsp_n, softsp_m, softsp_A, softsp_B)

integer(4) :: entity

real(8) :: Ecoh, softsp_n, softsp_m, softsp_A, softsp_B

entity (INPUT): Defines the material entity for which the soft-sphere parameters will be returned.

Ecoh (OUTPUT): Cohesive energy in erg/g.

softsp_n (OUTPUT): Parameter n in soft-sphere function.

softsp_m (OUTPUT): Parameter m in soft-sphere function.

softsp_A (OUTPUT): Parameter A in soft-sphere function.

softsp_B (OUTPUT): Parameter B in soft-sphere function.

This routine returns the parameters which are used in and calculated for the soft-sphere function. Attention: This data is only available if the soft-sphere function was initialized in FEOS_Init_Mat for the entity under consideration ("softsphereflag" = 1).

---

void FEOS_Get_Energy_Offsets(int entity, double* ElectronOffs, double* IonOffs)

FEOS_GET_ENERGY_OFFSETS(entity, ElectronOffs, IonOffs)

integer(4) :: entity

real(8) :: ElectronOffs, IonOffs

entity (INPUT): Defines the material entity for which the energy offsets will be returned.

ElectronOffs (OUTPUT): Energy offset of the electronic part of the EOS.

IonOffs (OUTPUT): Energy offset of the ionic part of the EOS.

This routine returns the specific energy and Helmholtz free energy offsets of the electronic and the ionic part of the EOS. These offsets assure that the energies are zero at the reference point ρ_o and T_o.

---

void FEOS_Get_Binodal(int entity, double* T, double* P, double* Rholiq, double* Rhovap, double* Pliq, double* Pvap, double* Gliq, double* Gvap, double* Hliq, double* Hvap, double* Zliq, double* Zvap, double* Tboil)

FEOS_GET_BINODAL(entity, T, P, Rholiq, Rhovap, Pliq, Pvap, Gliq, Gvap, Hliq, Hvap, Zliq, Zvap, Tboil)

integer(4) :: entity

real(8) :: T(numofMaxwelliso), P(numofMaxwelliso), Rholiq(numofMaxwelliso), Rhovap(numofMaxwelliso), Pliq(numofMaxwelliso), Pvap(numofMaxwelliso), Gliq(numofMaxwelliso), Gvap(numofMaxwelliso), Hliq(numofMaxwelliso), Hvap(numofMaxwelliso), Zliq(numofMaxwelliso), Zvap(numofMaxwelliso), Tboil

entity (INPUT): Defines the material entity for which the binodal will be returned.

T (OUTPUT): Array of size ”numofMaxwelliso” which contains all the temperatures for which the Maxwell construction has been calculated.

P (OUTPUT): Array of size ”numofMaxwelliso” containing the calculated saturated vapor pressures.

Rholiq (OUTPUT): Array of size ”numofMaxwelliso” containing the densities on the liquid branch of the binodal.

Rhovap (OUTPUT): Array of size ”numofMaxwelliso” containing the densities on the vapor branch of the binodal.

Pliq (OUTPUT): Array of size ”numofMaxwelliso” containing the pressures on the liquid branch of the binodal. For a Maxwell construction of good quality this array should correspond to the pressures on the vapor branch of the binodal and to the saturated vapor pressures.

Pvap (OUTPUT): Array of size "numofMaxwelliso" containing the pressures on the vapor branch of the binodal. For a Maxwell construction of good quality this array should correspond to the pressures on the liquid branch of the binodal and to the saturated vapor pressures.

Gliq (OUTPUT): Array of size "numofMaxwelliso" containing the specific Gibbs free energies on the liquid branch of the binodal. For a Maxwell construction of good quality this array should correspond to the specific Gibbs free energies on the vapor branch of the binodal.

Gvap (OUTPUT): Array of size "numofMaxwelliso" containing the specific Gibbs free energies on the vapor branch of the binodal. For a Maxwell construction of good quality this array should correspond to the specific Gibbs free energies on the liquid branch of the binodal.

Hliq (OUTPUT): Array of size "numofMaxwelliso" containing the specific enthalpies on the liquid branch of the binodal.

Hvap (OUTPUT): Array of size "numofMaxwelliso" containing the specific enthalpies on the vapor branch of the binodal.

Zliq (OUTPUT): Array of size "numofMaxwelliso" containing the compressibility factors on the liquid branch of the binodal.

Zvap (OUTPUT): Array of size "numofMaxwelliso" containing the compressibility factors on the vapor branch of the binodal.

Tboil (OUTPUT): Boiling temperature which corresponds to the temperature with saturated vapor pressure = 1 bar.

This routine returns several thermodynamic quantities on the liquid-vapor two-phase boundary (binodal). Attention: This data is only available if the Maxwell construction was initialized in FEOS_Init_Mat for the entity under consideration ("Maxwellflag" = 1).

---

void FEOS_Get_Spinodal(int entity, double* T, double* RhoMin, double* RhoMax, double* PMin, double* PMax)

FEOS_GET_SPINODAL(entity, T, RhoMin, RhoMax, PMin, PMax)

integer(4) :: entity

real(8) :: T(numofMaxwelliso), RhoMin(numofMaxwelliso),
RhoMax(numofMaxwelliso), PMin(numofMaxwelliso),
PMax(numofMaxwelliso)

entity (INPUT): Defines the material entity for which the spinodal will be returned.

T (OUTPUT): Array of size "numofMaxwelliso" which contains all the temperatures for which the Maxwell construction has been calculated.

RhoMin (OUTPUT): Array of size "numofMaxwelliso" which contains all the densities of the van der Waals loop minima.

RhoMax (OUTPUT): Array of size "numofMaxwelliso" which contains all the densities of the van der Waals loop maxima.

PMin (OUTPUT): Array of size "numofMaxwelliso" which contains all the pressures of the van der Waals loop minima.

PMax (OUTPUT): Array of size "numofMaxwelliso" which contains all the pressures of the van der Waals loop maxima.

This routine returns the minimal and maximum pressures on the van der Waals loops of the isotherms the Maxwell construction has been calculated for. The corresponding curve is also called the spinodal. Attention: This data is only available if the Maxwell construction was initialized in FEOS_Init_Mat for the entity under consideration ("Maxwellflag" = 1).

---

void FEOS_Get_Pmin_Pmax(int entity, double T, double* RhoMin, double* RhoMax, double* PMin, double* PMax)

FEOS_GET_PMIN_PMAX(entity, T, RhoMin, RhoMax, PMin, PMax)

integer(4) :: entity

real(8) :: T, RhoMin, RhoMax, PMin, PMax

entity (INPUT): Defines the material entity for which the minimum and maximum pressures shall be calculated.

T (INPUT): Temperature for which the minimum and maximum pressures will be calculated.

RhoMin (OUTPUT): Density of the van der Waals loop minima.

RhoMax (OUTPUT): Density of the van der Waals loop maxima.

PMin (OUTPUT): Pressure of the van der Waals loop minima.

PMax (OUTPUT): Pressure of the van der Waals loop maxima.

This routine allows to calculate the minimal and maximum pressures on the van der Waals loop for a given temperature. One has to assure that the input temperature lies below the critical point.

The FEOS Table Generation Tool

13 Generation of EOS Tables

The FEOS table generation tool allows for a quick access to the FEOS library in order to generate a new equation-of-state table of one material. Figure 5 gives an overview of the tool's operational sequence.

The parameters which are required for a calculation must be set in a parameter file (Section 4). Thus, in order to start a calculation one first should make a copy of one of the parameter files given as examples in the directory /EOS-Data. Then this file must be modified according to the desired properties and precision. Obligatory parameters are explained in Section 15.

The new parameter file must be located together with the executable "feos" generated during the FEOS package installation (Section 3), the material parameter database file and the TF table file in one directory (usually in /EOS-Data). The executable is started with the parameter file name (without suffix ".par"!) as argument. As an example, for the parameter file "Aluminum.par" the command "./feos Aluminum" must be used. The name of the parameter file will be also the name of the EOS table output files.

13.1 Additional Features

Besides the calculation of EOS tables the tool allows for two other options:

1. Single Point Information: If the EOS shall be calculated only for one given density-temperature point, then as an example for $\rho = 2.0\,\mathrm{g/cm^3}$, $T = 100\,\mathrm{eV}$, and for the parameter file "Aluminum.par" the user must execute the command "./feos Aluminum 2.0e0 1.0e2". The single point information will be printed out directly onto the console.
2. User-defined calculations: If executed in the table and not in the single point information modus, the code allows to call user-defined routines for simple and short calculations. How and where these routines have to be defined and called is explained in Section 17.

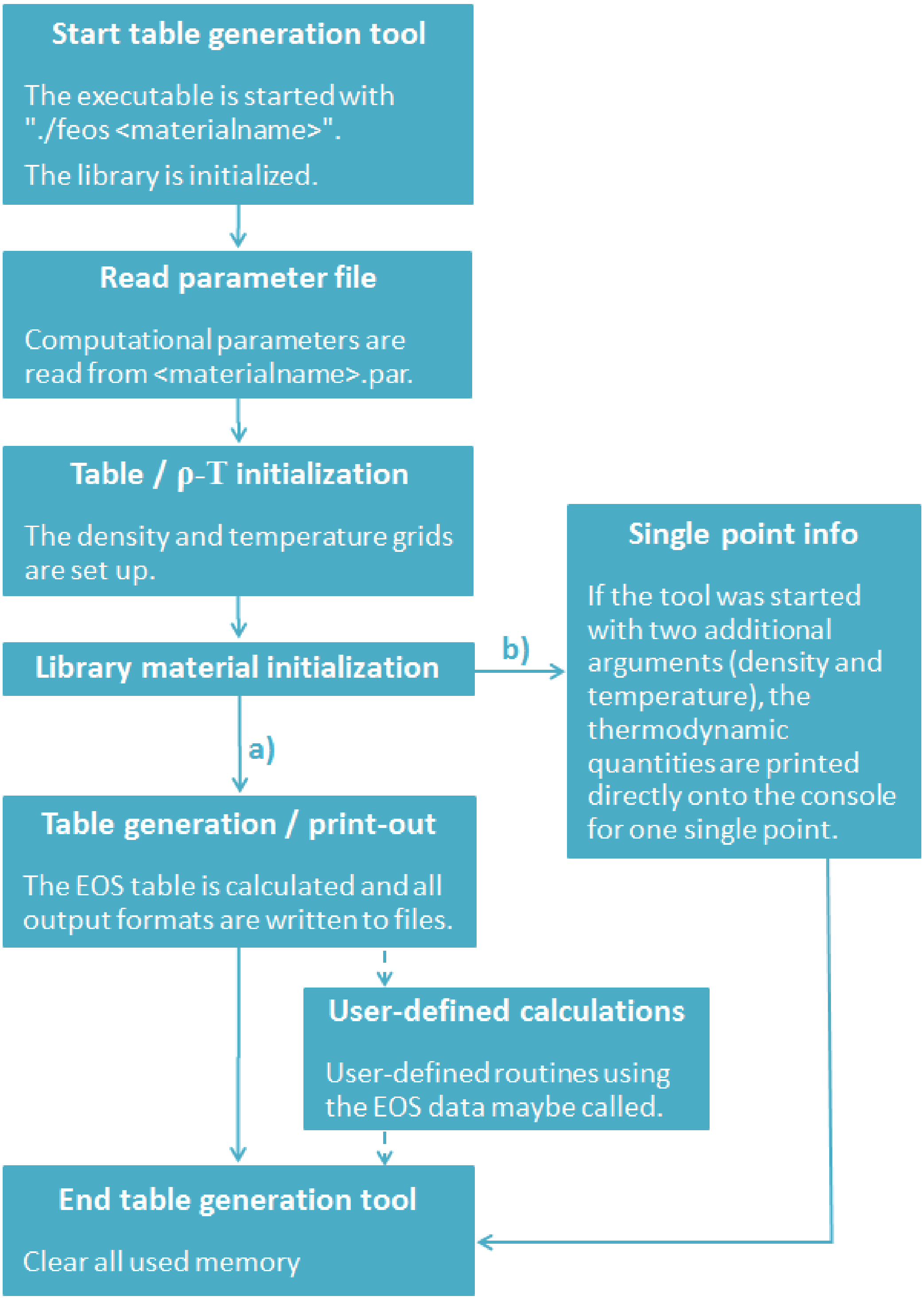

Start table generation tool
The executable is started with "./feos <materialname>".
The library is initialized.
Read parameter file
Computational parameters are read from <materialname>.par.
Table / ρ-T initialization
The density and temperature grids are set up.
Library material initialization
b)
a)
Single point info
If the tool was started with two additional arguments (density and temperature), the thermodynamic quantities are printed directly onto the console for one single point.
Table generation / print-out
The EOS table is calculated and all output formats are written to files.
User-defined calculations
User-defined routines using the EOS data maybe called.
End table generation tool
Clear all used memory

Figure 5: Operational sequence of the FEOS table generation tool.

14 Source Code Structure

The FEOS tool's source code consists of four files listed in Table 4.

FE-00_DEFINITS.H	Definitions used by the FEOS tool
FE-01_TABTOOLS.C	Routines for writing output files and for generating the density and temperature grids
FE-02_CALCULATIONS.C	User-defined routines for "simple" calculations using the FEOS library and/or the calculated tables
FE-03_MAIN.C	Main file of the FEOS tool

Table 4: Source files of the FEOS table generation tool.

The main routine of the tool is located in the source file "FE-03_MAIN.C". Users which want to use the FEOS library directly maybe interested in the main routine of the FEOS tool since it demonstrates the usage of nearly all of the library interface routines.

The file "FE-01_TABTOOLS.C" contains the routines ("getPointDensity(...)" and "makeRTPointArray(...)") which are used to create the density-temperature grid. Furthermore, all the table output file routines (Section 16) are located there.

In the file "FE-02_CALCULATIONS.C" the user can add own routines for (short and simple) calculations accessing the calculated table data or the library interface routines (Section 17).

15 Input Parameters

The variables for a calculation with the FEOS table generation tool are located in the first part (sections "Computation-Settings" and "Q-table") of a parameter file (Section 4). Table 5 lists all necessary settings.

One should keep in mind that for any calculation, the physical parameters of the material must be present with a corresponding material number in the material parameter database (Section 10). Especially, if one wants to use the

Material-Number	Number of the material in the FEOS material parameter database
Maxwell_Flag	Maxwell construction desired? (0: no, 1: yes)
SoftSphere_Flag	Replace TF cold curve and bonding correction for $\rho < \rho_o$ with soft-sphere function? (0: no, 1: yes)
UserCalculations_Flag	Perform user-defined calculations? (0: no, 1: yes)
Rhonorm	Density norm in g/cm^3
Rhoratio x	Number of logarithmically equidistant distributed density points in range Rhonorm$\cdot 10^{x-1}$ to Rhonorm$\cdot 10^{x}$, x=-6...6; Attention: The lower density library limit should not be violated!
Tnorm	Temperature norm in eV
Tratio x	Number of logarithmically equidistant distributed temperature points in range Tnorm$\cdot 10^{x-1}$ to Tnorm$\cdot 10^{x}$, x=-6...6; Attention: The lower temperature library limit should not be violated!

Table 5: Settings in the parameter file for the FEOS table generation tool (sections "Computation-Settings" and "Q-table").

soft-sphere function, the corresponding parameters must be available.

To calculate a table with a Maxwell construction (Section 6.6) one should take care of the quality of the critical data calculation which depends strongly on the choice of the temperature grid density in the two-phase region. One should not be too thrifty spending mesh points. The Maxwell construction is set-up during the library material initialization stage with the same temperatures which are used later on to calculate the EOS table.

If the UserCalculations_Flag is set to 1 the routines defined in the file "FE-02_CALCULATIONS.C" will be performed (Section 17).

Finally, it is important to take care of the lower density and temperature floor of the library (Section 6.1) when defining the density-temperature mesh. If points are defined below these limits, the EOS will not be consistent. Nevertheless, the FEOS table generation tool automatically adds $T = 0.0$ eV and $\rho = 0.0$ g/cm^3 to the density-temperature mesh.

16 EOS Table Structure

16.1 Qtable / CriticalDataTable Structures

The FEOS table generation tool stores the calculated EOS data and the critical data information (thermodynamic quantities on the vaporization curve) in two separate structures. Therefore, these structures contain several arrays and are defined in the file "COMMON-00_DEFINITS.H". The structure type "Qtable" contains the arrays for the calculated EOS table. The name of the most important arrays and their index ranges are listed in Table 6.

NRho, NT, Nelements	Number of densities, temperatures, and elements in the material
A[k], Z[k], X[k]	Atomic weights, atomic numbers and number of atoms per "molecule" for all elements in the material
Atot, Ztot, Xtot	Summed-up values: Atot=$\sum$ X[k] A[k], ...
T[j]	Temperatures
Rho[i]	Densities
P[i][j] (Pe, Pi, PTF)	Pressures (separate for EOS components)
E[i][j] (Ee, Ei, ET)	Specific energies (separate for EOS components)
S[i][j] (Se, Si, STF)	Specific entropies (separate for EOS components)
F[i][j] (Fe, Fi, FTF)	Specific Helmholtz free energies (separate for EOS components)
Q[i][j][k]	Charge states for all elements in the material
Qtot[i][j]	Summed-up (not mean!) charge states for one "molecule" (=$\sum$ X[k] Q[i][j][k])

Table 6: Most important variables and arrays stored in the "Qtable" structure. The indizes cover the following ranges: k=0..Nelements-1, j=0..NT-1, i=0..NRho-1.

The critical data information is stored in a structure type "CriticalDataTable". This type is explained in Table 7.

Niso	Number of calculated isotherms below the critical point
Rhoc, Pc, Tc, Hc	Density, pressure, temperature, and specific enthalpy at the critical point
Tiso[i]	Temperatures
Peq[i]	Saturated vapor pressures
Pmax[i], Pmin[i]	Minimum and maximum pressures on the van-der-Waals loops
Rhomax[i], Rhomin[i]	Densities which correspond to the pressure minima and maxima
Rhol[i], Rhov[i]	Densities on the liquid and the vapor branch of the binodal
Hl[i], Hv[i]	Specific enthalpies on the liquid and the vapor branch of the binodal

Table 7: Most important variables and arrays stored in the ”CriticalDataTable” structure. The index i covers the following range: i=0..Niso-1.

16.2 FEOS Format

The standard table output format of the FEOS table generation tool is a ASCII file with the extension ”.feos”. This type of file contains all information about the composition of the material, the calculation parameters and the calculated EOS table in the FEOS standard units (Section 5) itself. The format was designed to be used by the SHOWEOS table visualization tool and thus, it contains the whole set of calculated EOS data. In contrast to the SESAME format the file contains the complete EOS, all contributions to the EOS (electronic, ionic, Thomas-Fermi), and all thermodynamic functions (also the charge state). The FEOS format is controlled by the routine ”write_FEOS_format(...)” in the source file ”FE-01_TABTOOLS.C”.

16.3 SESAME Formats

The FEOS table generation tool also writes tables in the SESAME-301, -304, and -305 formats with file extensions ”.301”, ”.304”, and ”.305”. The

SESAME EOS library is a widely known set of equation-of-state tables for various materials, produced by the Los Alamons National Laboratory [19]. SESAME-301 tables contain information about the total pressure, specific energy, and specific Helmholtz free energy of a given material as functions of density and temperature. SESAME-304/305 tables contain the same information, but for the electronic/ionic contributions. The tables in the SESAME database are based on experimental results as well as on theoretical models, sometimes on both. Additional information, for example about the vaporisation curve etc., is usually available in extra archive files (in case of FEOS this data is written to other formats).

A typical SESAME table is an ASCII-file consisting of several lines which contain four words each, where each word is a real number. Its first line contains a material code, the solid density of the material described (in cgs units), the number of points on the density mesh (as real number), and the number of points on the temperature mesh (as real number), example:

```
3717           2.70000000e+00 1.92000000e+02 7.30000000e+01
```

The next few lines contain information about the density mesh (Rho[I], I=1...NI) and the temperature mesh (T[J], J=1...NJ):

```
0.00000000e+00 2.70000000e-07 5.81697366e-07 2.70000000e-06 ...
```

The next lines contain all pressure isotherms, ((P[I,J], I=1..NI), J=1..NJ), then specific energy and specific Helmholtz free energy isotherms in the same manner as for the pressure.

The units used in the SESAME library are:

- Pressure: GPa
- Energy: MJ/kg
- Density: g/cm^3
- Temperature: Kelvin

In "FE-01_TABTOOLS.C" the routines "write_301(304,305)_format(...)" are responsible for the file output of the SESAME tables. Furthermore, the routine "write_mexport_format(...)" writes the SESAME mexport format (extension ".mexport"). The mexport format is a database file which usually contains all the EOS tables (301,...) of all materials from the SESAME database. In the case of the FEOS table generation tool the mexport file of course contains only the 301, 304, and 305 tables of the calculated material.

16.4 Other Output Formats

Besides the FEOS and the SESAME output formats, two other formats are written to files. The first one, the txt format with file extension ".data.txt" (routine "write_txt_format" in "FE-01_TABTOOLS.C") contains pressure-energy isotherms as function of mass density. The second one, the Rostock format with file extension ".cst" (routine "write_Rostock_format(...)" in "FE-01_TABTOOLS.C") contains pressure-charge state isotherms as function of particle density and mass density.

If a material is initialized with a Maxwell construction (Section 6.6), the FEOS table generation tool also writes out the information from the CriticalDataTable structure (Section 16.1). The corresponding output file (routine "write_criticaldata(...)" in "FE-01_TABTOOLS.C") has the extension ".critical.dat"and contains the following information:

- Critical point data
- Boiling temperature
- Information about the accuracy of the Maxwell construction
- Binodal and spinodal in ρ-p-plane
- Binodal, spinodal, and diamener curve in T-ρ-plane
- Binodal in T-H-plane (specific enthalpy)
- Evaporation heat ΔH in T-ΔH-plane
- Saturation curve in Arrhenius coordinates
- Compressibility factor on the binodal as function of temperature and pressure

If the user wants to add a own table format to the FEOS table generation tool, a corresponding routine which may access the "Qtable" and/or "CriticalDataTable" structures (Section 16.1) must be added to the source file

"FE-01_TABTOOLS.C" and the corresponding header file. The new routine must be called with appropriate arguments (together with the other output routines) at the end of step 4b in the main function in the source file "FE-03_MAIN.C".

17 User-Defined Calculations

The FEOS table generation tool provides an extra source file for the user reserved for routines which access the calculated EOS table and/or call the FEOS interface routines. The advantage of this special option is that the user can benefit for "simple" calculations from an existing infrastructure for the initialization of the FEOS library and of an EOS table. For more complicated calculations of course it may become necessary that the user writes a fully new program which calls the FEOS library from the beginning.

The user-defined routines are all located and called by the routine "UserCalculations(...)" in the source file "FE-02_CALCULATIONS.C". Four arguments maybe passed by the "UserCalculations(...)" routine, nameley the entitynumber of the material (for the FEOS table generation tool normally always 1), the name of the parameter file, and the stored data in the "Qtable" and "CriticalDataTable" structures (Section 16.1).

If a new routine needs the Maxwell construction data, the user should make sure that it was calculated before by checking the integer "Niso" in the "CriticalDataTable" structure to be greater than zero.

17.1 Isobaric Expansion Data

One (exemplary) application comes already with the delivery version of the FEOS package. The routine "IsobaricExpansion(...)" can be used for the calculation of thermodynamic functions along the isobaric curve $p = 0$ of the metastable EOS (without Maxwell construction) up to the spinodal limit which means: up to that temperature for which the van-der-Waals loop does not cross $p = 0$ anymore. The minimum and maximum temperatures, as well as the number of linearly equidistant distributed temperature points are set at the beginning of the routine "IsobaricExpansion(...)". The following data is calculated along the isobaric curve and written to an output file with extension ".isobaric.dat":

- Temperatures T
- Densities ρ
- Pressures p (for checking accuracy only)
- Thermal expansion coefficients α
- Specific enthalpies H
- Specific isobaric heat capacities C_p

This data, especially the expansion coefficient, can be helpful for the adjustment of the soft-sphere function parameters (Section 6.7) of a new material entry in the material parameter database.

The SHOWEOS Table Visualization Tool

18 Visualization of EOS Tables

SHOWEOS is a tool for calculating plots of the EOS tables generated with the FEOS table generation tool. The tool is designed for table files which are present in the FEOS format (Section 16.2) with extension ".feos". Alternatively, the SESAME file format (Section 16.3) with extensions ".301", ".304", and ".305" can be loaded. The latter feature provides the option to visualize also tables from the SESAME database and to compare them with the corresponding FEOS tables.

Figure 6 shows the general operational sequence of the tool. In order to start a calculation one first must ensure that the executable "showeos" generated during the FEOS package installation (Section 3), the parameter file with

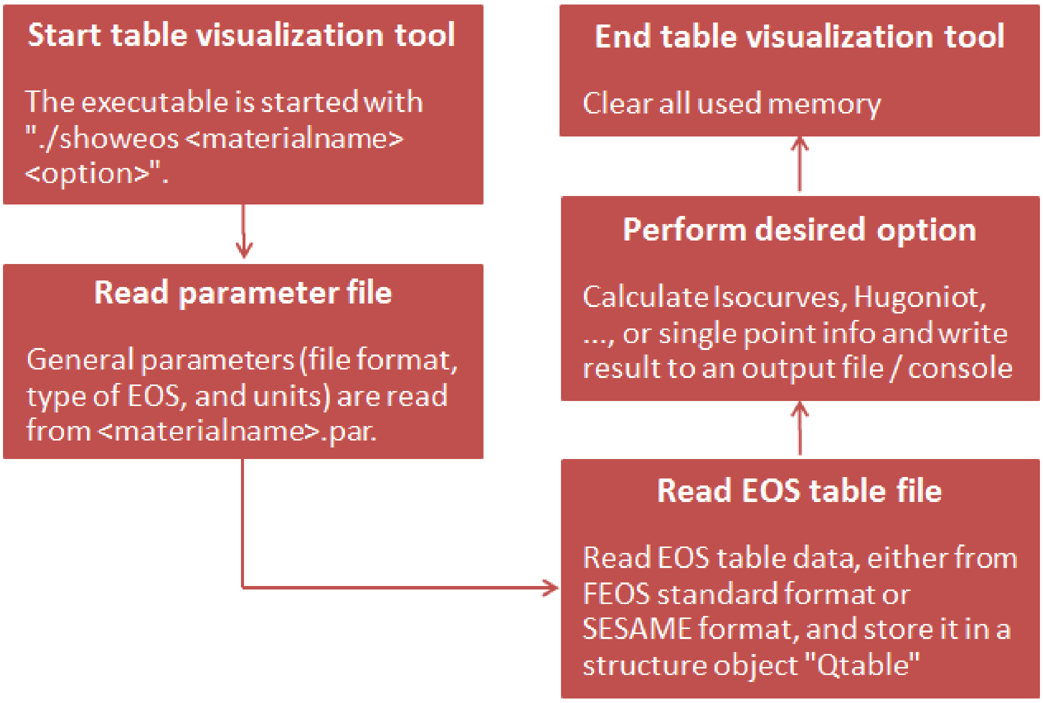

Start table visualization tool
The executable is started with "./showeos <materialname> <option>".
Read parameter file
General parameters (file format, type of EOS, and units) are read from <materialname>.par.
Read EOS table file
Read EOS table data, either from FEOS standard format or SESAME format, and store it in a structure object "Qtable"
Perform desired option
Calculate Isocurves, Hugoniot, ..., or single point info and write result to an output file / console
End table visualization tool
Clear all used memory

Figure 6: Operational sequence of the SHOWEOS table visualization tool.

extension ".par" (Section 4), and the table file, which must have the same name as the parameter file, are located in the same directory (usually /EOS-Data). Usually, the same parameter file which was used for the generation of the table with the FEOS table generation tool contains also the parameters for SHOWEOS. Only if a table from the SESAME database shall be visualized, a new parameter file must be created. The name of the parameter file will be also the name of the output file(s).

The executable "showeos" is started with two arguments. The first argument is the name of the parameter file (without suffix ".par"!). The second argument is an integer number between 1 and 6 which defines the type of the calculation which will be done. As an example, for the parameter file "Aluminum.par" and option number 3 one has to execute the command "./showeos Aluminum 3". Table 8 explains the meaning of the six available options. The corresponding procedures and settings in the parameter file are explained in Section 20. Options 1-3 are collected under the umbrella term "Isocurves".

1	Calculate isotherms (lines of constant temperature)
2	Calculate isochores (lines of constant density)
3	Calculate isentropes (lines of constant specific entropy)
4	Calculate mountain plot (three-dimensional plot of thermodynamic quantity as function of density / specific volume and temperature)
5	Calculate Hugoniot curve [16]
6	Print out single point information

Table 8: Calculational options provided by the SHOWEOS table visualization tool.

Once the tool has been started, first the general parameters (Section 20.1) are read from the parameter file, then the appropriate EOS table file is loaded, and finally the desired option is performed. For options 1-5 the information is printed out into an ASCII file which then again can be loaded by a plotting tool. Here, the user has to take care by himself/herself of the format of the ASCII file by modifying the routine which is responsible for the desired option. Finally, although SHOWEOS, like the other parts of the package, internaly uses the FEOS standard units, the units of the SHOWEOS input parameters and of the quantities written to the output file(s) can be changed and have to be specified in the parameter file (Section 20.1).

19 Source Code Structure

The SHOWEOS tool's source code consists of five files listed in Table 9.

SE-00_DEFINITS.H	Definitions for the output files of the SHOWEOS tool
SE-01_READTABLE.C	Routines for reading EOS table files
SE-02_INTERPOLTOOLS.C	Routines for table interpolations
SE-03_SERVICES.C	"Working" routines for isocurves, Hugoniots, etc.
SE-04_MAIN.C	Main file and main routine of the SHOWEOS tool

Table 9: Source files of the SHOWEOS table visualization tool.

The main routine of the tool is located in the source file "SE-04_MAIN.C". The file "SE-01_READTABLE.C" contains the routines which are used to read and load the EOS tables. In the file "SE-03_SERVICES.C" the "working" routines which are responsible for the different options of the tool are located. In these routines the user must adjust the output file write statements to fit the requirements of the used plotting tool.

20 Input Parameters and Procedures

The variables for a calculation with the SHOWEOS table visualization tool are located in the second part of a parameter file (Section 4). The SHOWEOS parameters are separated into two categories: 1) the obligatory parameters which are used by all options of the tool and are explained in Section 20.1 and 2) the facultative parameters which are only used if the corresponding option is performed (Sections 20.2, 20.3, 20.4, and 20.5).

20.1 General Settings

In a parameter file three sections "General", "Units", and "Rho-T-Mesh" contain parameters which are (partially) used by all options of the SHOWEOS tool. The parameters of the "General" and "Units" sections are explained in Table 10.

File-Format	Type of table file (1: FEOS format, 2: SESAME format)
EOS-Type	Type of EOS (1: total, 2: electronic, 3: ionic, 4: Thomas-Fermi)
Rho_unit	Density unit (1.0 → g/cm^3)
T_unit	Temperature unit (1.0 → eV, 8.61753e-5 → K)
P_unit	Pressure unit (1.0 → dyne/cm^2, 1.0e12 → MBar)
E_unit	Energy unit (1.0 → erg/g, 1.0e10 → MJ/kg)
U_unit	Velocity unit (1.0 → cm/s, 1.0e5 → km/s)

Table 10: Settings in the parameter file for the SHOWEOS table visualization tool (sections "General" and "Units").

The parameter "EOS-Type" controls whether the complete or the electronic / ionic / pure Thomas-Fermi contribution shall be loaded. While the FEOS formatted files (Section 16.2) contain all the required data, for the complete EOS as well as for the contributions, a SESAME table file does not. In this case the SHOWEOS tool automatically chooses the appropriate file extension (".301", ".304", or ".305") depending on the "EOS-Type" parameter. Note, that the option "EOS-Type=4" is not available for SESAME tables. Furthermore, one should keep in mind that the SESAME table files which can be loaded contain no information about the charge state.

The unit parameters are set in such a way that the value corresponds to a multiplication factor of the FEOS standard units (Section 5). All input has to be defined and all output quantities are printed-out in the defined units. The units of the specific Helmholtz free energy, the specific entropy, and the specific volume are determined from the other unit settings. The unit of the charge state is fixed in terms of multiples of the unit electron charge.

Table 11 lists all parameters of the "Rho-T-Mesh" section. This section is important for all SHOWEOS options, except for the single point information and the Hugoniot option. The corresponding parameters define the strucuture of the density-temperature grid. The user can choose whether the original densities and temperatures from the table file shall be used, or a new mesh with linearly or logarithmically equidistant distributed points shall be created.

Rhomin, Tmin	Minimum density / temperature (in units defined in section ”Units”)
Rhomax, Tmax	Maximum density / temperature (in units defined in section ”Units”)
Rhooriginal, Toriginal	Use only original tabulated densities / temperatures? (0: no, 1: yes)
Rhonumber, Tnumber	Number of densities / temperatures if Rhooriginal / Toriginal is set to 0
Rholog, Tlog	Kind of distribution if Rhooriginal / Toriginal is set to 0 (0: linear, 1: logarithmical)
Rhofirst, Tfirst	Only for isotherms, isochores, or Mountain plots: Print out lowest density / temperature of table? (0: no, 1: yes)

Table 11: Settings in the parameter file for the SHOWEOS table visualization tool (section ”Rho-T-Mesh”).

20.2 Isocurves

For the first three options of the SHOWEOS table visualization tool (calculation of isotherms, isochores, or isentropes) the parameters from the parameter file section ”Isocurves” are read. The corresponding settings are explained in Table 12.

Xquantity, Yquantity	Quantity which is printed out on the x-axis / y-axis (1,2,3,...,8 → Table 13)
Xelement, Yelement	For charge state of mixtures (Xquantity=8 / Yquantity=8): element number or 0 for summed-up value

Table 12: Settings in the parameter file for the SHOWEOS table visualization tool (section ”Isocurves”).

The user must choose which quantities shall be plotted on the x- and on the y-axis. Therefore, the different thermodynamic quantities correspond to number between 1 and 8 (Table 13). If the charge state shall be plotted for a mixture, the user must decide whether the charge state shall be plotted for a participating element or summed up for a ”molecule” of the mixture.

Attention: The charge state is only available for the FEOS table format and not for the supported SESAME table files.

1	Density
2	Specific volume
3	Temperature
4	Pressure
5	Specific internal energy
6	Specific Helmholtz free energy
7	Specific entropy
8	Summed-up or single-element charge state

Table 13: Possible quantity choices.

The output file will be an ASCII file with the same name as the parameter file. The first part of the file extension is composed of the names of the plotted quantities on the x- and on the y-axis. The second and last part of the extension denotes the type of isocurve: ".ist" for isotherms, ".isc" for isochores, and ".ise" for isentropes. In the delivery version of the SHOWEOS tool successive isocurves inside the ASCII file are divided by an "&" sign. Each isocurve contains a headline with information of the plotted quantities and units. This sort of file can be graphically visualized with the Unix tool xmgrace, for example.

For isotherms, the plotted temperatures are given by the temperature grid which was set-up in the section "Rho-T-Mesh" (Section 20.1). For isochores, the plotted densities are given by the density grid. For isentropes, the used entropies are calculated for the lowest density and for all temperatures of the density-temperature grid.

20.3 Mountain Plots

For showing total phase planes which means some quantity (parameter "Quantity") as a function of density or specific volume (parameter "Xquantity") and temperature in a three-dimensional diagram, one has to make specifications in the parameter file section "Mountain" (Table 14). The densities and temperatures are given by the density-temperature grid (Section 20.1).

The output file will be an ASCII file with the same name as the parameter file. The file first contains the density and temperature grid and then a list of isochores of the desired quantity. The first part of the file extension is composed of the names of the plotted quantities on the x-, y-, and z-axis. The second and last part of the extension is ".mnt".

Xquantity	Quantity which is printed out on the x-axis (1: density, 2: specific volume)
Quantity	Mountain quantity as function of Xquantity and temperature (4,5,6,7,8 → Table 13)
Element	For charge state of mixtures (Quantity=8): element number or 0 for summed-up value

Table 14: Settings in the parameter file for the SHOWEOS table visualization tool (section "Mountain").

20.4 Hugoniots

For computing a Hugoniot curve [16] the corresponding settings in the parameter file section "Hugoniot" must be specified (Table 15). The user must set the initial density and temperature where the Hugoniot curve calculation shall start. Furthermore, a maximum pressure must be set to limit the calculation. The stepsizes for the calculation and for the print-out can be changed in the source code header file "SE-03_SERVICES.H". The output file will be an ASCII file with the same name as the parameter file and extension ".hug". The file contains the density or specific volume (parameter "Xquantity"), the temperature, the pressure, the specific internal energy, the shock wave velocity, and the particle velocity behind the shock wave.

Rho0	Initial density (in units defined in section "Units")
T0	Initial temperature (in units defined in section "Units")
Pmax	Maximum pressure (in units defined in section "Units")
Xquantity	Quantity which is printed out on the "x-axis" (1: density, 2: specific volume)

Table 15: Settings in the parameter file for the SHOWEOS table visualization tool (section "Hugoniot").

20.5 Single Point Information

A single point information for a given density and temperature can be printed out with the option 6 of the SHOWEOS table visualization tool. The density and the temperature in the units defined in the parameter file section "Units" must be passed as a third and a fourth argument in the command line call of SHOWEOS. As an example, for the parameter file "Aluminum.par", the value 1.0 for the temperature and density units in the parameter file, density 0.3 g/cm^3, and temperature 10 eV one has to execute the command "./showeos Aluminum 6 0.3 1.e2".

References

[1] A. Kemp and J. Meyer-ter Vehn. An equation of state code for hot dense matter, based on the QEOS description. *Nucl. Instrum. Methods A*, 415:674–676, 1998. 2, 8

[2] R. M. More, K. H. Warren, D. A. Young, and G. B. Zimmerman. A new quotidian equation of state (QEOS) for hot dense matter. *Physics of Fluids*, 31:3059, 1988. 6, 8, 11, 15, 19

[3] A. Kemp and J. Meyer-ter Vehn. Das Zustandsgleichungs-Modell QEOS für heisse, dichte Materie. Technical Report 229, Max-Planck-Institut für Quantenoptik, 1998. 6, 15, 19

[4] S. Faik, M. M. Basko, An. Tauschwitz, I. Iosilevskiy, and J. A. Maruhn. Dynamics of volumetrically heated matter passing through the liquid-vapor metastable states. *High Energy Density Physics*, 8(4):349–359, 2012. 8, 20

[5] D. A. Young and E. M. Corey. A new global equation of state model for hot, dense matter. *Journal of Applied Physics*, 78(6):3748, 1995. 11, 22

[6] L. D. Landau and E. M. Lifshitz. *Statistical Physics.* Butterworth Heinemann, 3 edition, 1996. 12, 20

[7] L. H. Thomas. The calculation of atomic fields. *Mathematical Proceedings of the Cambridge Philosophical Society*, 23(5):542–548, 1927. 12

[8] E. Fermi. Eine statistische Methode zur Bestimmung einiger Eigenschaften des Atoms und ihre Anwendung auf die Theorie des periodischen Systems der Elemente. *Zeitschrift für Physik*, 48:73–79, 1928. 12

[9] R. Feynman, N. Metropolis, and E. Teller. Equations of state of elements based on the generalized Thomas-Fermi theory. *Physical Review*, 75:1561, 1949. 12, 13

[10] L. D. Landau and E. M. Lifshitz. *Quantum Mechanics: Non Relativistic Theory.* Butterworth Heinemann, 3 edition, 1981. 13

[11] M. Brachmann. Thermodynamic functions on the generalized Thomas-Fermi theory. *Physical Review*, 84:1263, 1951. 14

[12] S. Eliezer, A. Ghatak, and H. Hora. *Fundamentals of Equations of State.* World Scientific Pub Co, 1 edition, 2002. 15, 17

[13] D. A. Kirzhnits, Y. E. Lozovik, and G. V. Shpatakovskaya. Statistical model of matter. *Soviet Physics Uspekhi*, 18(9):649–672, 1975. 15

[14] B.-G. Englert. *Semiclassical Theory of Atoms (Lecture Notes in Physics)*. Springer, Berlin, 1 edition, 1988. 15

[15] W. Zittel. Elektronische Struktur hochkomprimierter Materie. Technical Report 111, Max-Planck-Institut für Quantenoptik, 1986. 15

[16] Ya. B. Zeldovich and Yu. P. Raizer. *Physics of Shock-Waves and High-Temperature Hydrodynamic Phenomena*. Dover Pubn Inc, illustrated edition, 2002. 16, 56, 61

[17] J. F. Barnes. Statistical Atom Theory and the Equation of State of Solids. *Physical Review*, 153(1):269–275, 1967. 16

[18] M. Ross. Generalized Lindemann Melting Law. *Physical Review*, 184(1):233–242, 1969. 17

[19] T. Group. SESAME Report on the Los Alamos Equation of State Library. Technical Report LALP-83-4, Los Alamos National Laboratory, 1983. 51