SPHERLSgen

1.0

Generated by Doxygen 1.7.6.1

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1   File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 eos Class Reference

`#include <eos.h>`

Public Member Functions

- eos ()
- eos (int nNumT, int nNumRho)
- eos (const eos &ref)
- ∼eos ()
- eos & operator= (const eos &eosRightSide)
- void readAscii (std::string sFileName)
- void readBobsAscii (std::string sFileName)
- void writeAscii (std::string sFileName)
- void readBin (std::string sFileName) throw (exception2)
- void writeBin (std::string sFileName)
- double dGetPressure (double dT, double dRho)
- double dGetEnergy (double dT, double dRho)
- double dGetOpacity (double dT, double dRho)
- double dDRhoDP (double dT, double dRho)
- double dSoundSpeed (double dT, double dRho)
- void getEKappa (double dT, double dRho, double &dE, double &dKappa)
- void getPEKappa (double dT, double dRho, double &dP, double &dE, double &d-Kappa)
- void getPEKappaGamma (double dT, double dRho, double &dP, double &dE, double &dKappa, double &dGamma)
- void getPEKappaGammaCp (double dT, double dRho, double &dP, double &dE, double &dKappa, double &dGamma, double &dCp)
- void getPKappaGamma (double dT, double dRho, double &dP, double &dKappa, double &dGamma)

- void gamma1DelAdC_v (double dT, double dRho, double &dGamma1, double &dDelAd, double &dC_v)
- void getPAndDRhoDP (double dT, double dRho, double &dP, double &dDRhoDP)
- void getEAndDTDE (double dT, double dRho, double &dE, double &dDTDE)
- void getDlnPDlnTDlnPDlnPDEDT (double dT, double dRho, double &dDlnPDlnT, double &dDlnPDlnRho, double &dDEDT)

Public Attributes

- int nNumRho
- int nNumT
- double dXMassFrac
- double dYMassFrac
- double dLogRhoMin
- double dLogRhoDelta
- double dLogTMin
- double dLogTDelta
- double ∗∗ dLogP
- double ∗∗ dLogE
- double ∗∗ dLogKappa

3.1.1 Detailed Description

This class holds an equation of state as well as many functions useful for manipulating it

3.1.2 Constructor & Destructor Documentation

3.1.2.1 eos::eos ( )

Constructor, doesn't really do anything

References dLogE, dLogKappa, dLogP, nNumRho, and nNumT.

3.1.2.2 eos::eos (int *nNumT,* int *nNumRho*)

Constructor, allocates memory for the 2D arrays

Parameters

in	*nNumT*	number of temperatures in the equaiton of state table
in	*nNumRho*	number of densities in the equaiton of state table

3.1.2.3   eos::eos (const eos & *ref*)

Copy constructor, simply constructs a new eos object from another eos object

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.2.4   eos::∼eos ( )

Destructor, deletes dynamic arrays

References dLogE, dLogKappa, dLogP, and nNumRho.

3.1.3   Member Function Documentation

3.1.3.1   double eos::dDRhoDP (double *dT,* double *dRho*)

This function calculates the partial derivative of density w.r.t. pressure

Parameters

in	*dT*	temperature at which the derivative is to be computed
in	*dRho*	density at which the derivative is to be computed

Returns

the partial derivative of density w.r.t. pressure.

References dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

3.1.3.2   double eos::dGetEnergy (double *dT,* double *dRho*)

This function linearly interpolates the energy to a given temperature and and density. Note that both `dT` and `dRho` are not in log space.

Parameters

in	*dT*	temperature to interpolate to.
in	*dRho*	density to interpolate to.

Returns

the interpolated energy.

References dLogE, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

3.1.3.3 **double eos::dGetOpacity (** double *dT,* double *dRho* **)**

This function linearly interpolates the opacity to a given temperature and and density. Note that both `dT` and `dRho` are not in log space.

Parameters

in	*dT*	temperature to interpolate to.
in	*dRho*	density to interpolate to.

Returns

the interpolated opacity.

References dLogKappa, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNum-Rho, and nNumT.

3.1.3.4 **double eos::dGetPressure (** double *dT,* double *dRho* **)**

This function linearly interpolates the pressure to a given temperature and density. Note that both `dT` and `dRho` are not in log space.

Parameters

in	*dT*	temperature to interpolate to.
in	*dRho*	density to interpolate to.

Returns

the interpolated pressure.

References dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

3.1.3.5 **double eos::dSoundSpeed (** double *dT,* double *dRho* **)**

This function calculates the adiabatic sound speed

Parameters

in	*dT*	temperature at which the derivative is to be computed
in	*dRho*	density at which the derivative is to be computed

Returns

the sound speed.

References dLogE, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, n-NumRho, and nNumT.

3.1.3.6 void eos::gamma1DelAdC_v (double *dT,* double *dRho,* double & *dGamma1,* double & *dDelAd,* double & *dC_v*)

This function calculates gamma1 and the adiabatic gradient

Parameters

in	dT	temperature at which the derivative is to be computed
in	dRho	density at which the derivative is to be computed
out	dGamma1	gamma1
out	dDelAd	adiabatic gradient
out	dC_v	specific heat at constant volume

References dLogE, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, n-NumRho, and nNumT.

3.1.3.7 void eos::getDlnPDlnTDlnPDlnPDEDT (double *dT,* double *dRho,* double & *dDlnPDlnT,* double & *dDlnPDlnRho,* double & *dDEDT*)

This function calculates various partial derivatives

Parameters

in	dT	temperature at which the derivative is to be computed
in	dRho	density at which the derivative is to be computed
out	dDlnPDlnT	derivative of ln(P) w.r.t. ln(T)
out	dDlnPDln-Rho	derivative of ln(P) w.r.t. ln(Rho)
out	dDEDT	derivative of temperature w.r.t. energy at constant density

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, nNumRho, and nNumT.

3.1.3.8 void eos::getEAndDTDE (double *dT,* double *dRho,* double & *dE,* double & *dDTDE*)

This function calculates the partial derivative of temperature w.r.t. energy and the energy

Parameters

in	dT	temperature at which the derivative is to be computed
in	dRho	density at which the derivative is to be computed

out	*dE*	energy at dT and dRho
out	*dDTDE*	derivative of temperature w.r.t. energy at constant density

References dLogE, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

3.1.3.9 void eos::getEKappa (double *dT,* double *dRho,* double & *dE,* double & *dKappa*)

This function linearly interpolates the three dependent quantities (Pressure, Energy , Opacity) to a given temperature and density. Note that both dT and dRho are not in log space.

Parameters

in	*dT*	temperature to interpolate to.
in	*dRho*	density to interpolate to.
out	*dE*	energy at dT and dRho.
out	*dKappa*	opacity at dT and dRho.

References dLogE, dLogKappa, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

3.1.3.10 void eos::getPAndDRhoDP (double *dT,* double *dRho,* double & *dP,* double & *dDRhoDP*)

This function calculates the partial derivative of density w.r.t. pressure and the pressure

Parameters

in	*dT*	temperature at which the derivative is to be computed
in	*dRho*	density at which the derivative is to be computed
out	*dP*	pressure at dT and dRho
out	*dDRhoDP*	derivative of density w.r.t. pressure at conatant temperature

References dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, dLogTMin, nNumRho, and nNumT.

3.1.3.11 void eos::getPEKappa (double *dT,* double *dRho,* double & *dP,* double & *dE,* double & *dKappa*)

This function linearly interpolates the three dependent quantities (Pressure, Energy , Opacity) to a given temperature and density. Note that both dT and dRho are not in log space.

Parameters

in	dT	temperature to interpolate to.
in	dRho	density to interpolate to.
out	dP	pressure at dT and dRho.
out	dE	energy at dT and dRho.
out	dKappa	opacity at dT and dRho.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, nNumRho, and nNumT.

3.1.3.12 void eos::getPEKappaGamma (double *dT,* double *dRho,* double & *dP,* double & *dE,* double & *dKappa,* double & *dGamma*)

This function linearly interpolates the energy and opacity to a given temperature and density. Note that both `dT` and `dRho` are not in log space.

Parameters

in	dT	temperature to interpolate to.
in	dRho	density to interpolate to.
out	dP	pressure at dT and dRho.
out	dE	energy at dT and dRho.
out	dKappa	opacity at dT and dRho.
out	dGamma	adiabatic index at dT and dRho.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, nNumRho, and nNumT.

3.1.3.13 void eos::getPEKappaGammaCp (double *dT,* double *dRho,* double & *dP,* double & *dE,* double & *dKappa,* double & *dGamma,* double & *dCp*)

This function linearly interpolates the energy and opacity to a given temperature and density. Note that both `dT` and `dRho` are not in log space.

Parameters

in	dT	temperature to interpolate to.
in	dRho	density to interpolate to.
out	dP	pressure at dT and dRho.
out	dE	energy at dT and dRho.
out	dKappa	opacity at dT and dRho.
out	dGamma	adiabatic index at dT and dRho.
out	dCp	specific heat at constant pressure at dT and dRho.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, nNumRho, and nNumT.

3.1.3.14 **void eos::getPKappaGamma (** double *dT,* double *dRho,* double & *dP,* double &
dKappa, double & *dGamma* **)**

This function linearly interpolates the energy and opacity to a given temperature and
density. Note that both `dT` and `dRho` are not in log space.

Parameters

in	*dT*	temperature to interpolate to.
in	*dRho*	density to interpolate to.
out	*dP*	pressure at dT and dRho.
out	*dKappa*	opacity at dT and dRho.
out	*dGamma*	adiabatic index at dT and dRho.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-
LogTMin, nNumRho, and nNumT.

3.1.3.15 **eos & eos::operator= (** const **eos &** *eosRightSide* **)**

Assignment operator, assigns one eos object to another.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-
LogTMin, nNumRho, and nNumT.

3.1.3.16 **void eos::readAscii (** std::string *sFileName* **)**

This fuction reads in an ascii file and stores it in the current object.

Parameters

in	*sFileName*	name of the equation of state file to read from.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-
LogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.3.17 **void eos::readBin (** std::string *sFileName* **)** throw **(exception2)**

This fuction reads in a binary file and stores it in the current object.

Parameters

in	*sFileName*	name of the equation of state file to read from.

3.1.3.18 **void eos::readBobsAscii (** std::string *sFileName* **)**

This fuction reads in an ascii file and stores it in the current object. The ascii file is in
Bob's format.

Parameters

in	*sFileName*	name of the equation of state file to read from.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.3.19 void eos::writeAscii (std::string *sFileName*)

This fuction writes the equation of state stored in the current object to an ascii file.

Parameters

in	*sFileName*	name of the file to write the equation of state to.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.3.20 void eos::writeBin (std::string *sFileName*)

This fuction writes the equation of state stored in the current object to a binary file.

Parameters

in	*sFileName*	name of the file to write the equaiton of state to.

References dLogE, dLogKappa, dLogP, dLogRhoDelta, dLogRhoMin, dLogTDelta, d-LogTMin, dXMassFrac, dYMassFrac, nNumRho, and nNumT.

3.1.4 Member Data Documentation

3.1.4.1 double∗∗ eos::dLogE

2D array of log10 energies. dLogE[i][j] gives the log10 energy at log10 density of eos::dLogRhoDelta∗i+eos::dLogRhoMin, and at log10 temperature of eos::dLogTDelta∗j+eos::dLogTMin.

Referenced by dGetEnergy(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getDlnP-DlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPEKappa(), getPEKappa-Gamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), writeAscii(), writeBin(), and ∼eos().

3.1.4.2 double∗∗ eos::dLogKappa

2D array of log10 opacities. dLogKappa[i][j] gives the log10 opacity at log10 density of eos::dLogRhoDelta∗i+eos::dLogRhoMin, and at log10 temperature of eos::dLogTDelta∗j+eos::dLogTMin.

Referenced by dGetOpacity(), eos(), getDlnPDlnTDlnPDlnPDEDT(), getEKappa(), get-PEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), writeAscii(), writeBin(), and ~eos().

3.1.4.3 double∗∗ eos::dLogP

2D array of log10 pressures. dLogP[i][j] gives the log10 pressure at log10 density of eos::dLogRhoDelta∗i+eos::dLogRhoMin, and at log10 temperature of eos::dLogTDelta∗j+eos::dLogTMin.

Referenced by dDRhoDP(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC-_v(), getDlnPDlnTDlnPDlnPDEDT(), getPAndDRhoDP(), getPEKappa(), getPEKappa-Gamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), writeAscii(), writeBin(), and ~eos().

3.1.4.4 double eos::dLogRhoDelta

Increment of the density between table entries in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSound-Speed(), eos(), gamma1DelAdC_v(), getDlnPDlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappa-GammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), write-Ascii(), and writeBin().

3.1.4.5 double eos::dLogRhoMin

Minimum density of the table in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSound-Speed(), eos(), gamma1DelAdC_v(), getDlnPDlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappa-GammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), write-Ascii(), and writeBin().

3.1.4.6 double eos::dLogTDelta

Increment of the temperature between table entries in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSound-Speed(), eos(), gamma1DelAdC_v(), getDlnPDlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappa-GammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), write-Ascii(), and writeBin().

3.1.4.7 double eos::dLogTMin

Minimum temperature of the table in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSound-Speed(), eos(), gamma1DelAdC_v(), getDlnPDlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappa-GammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), write-Ascii(), and writeBin().

3.1.4.8 double **eos::dXMassFrac**

Hydrogen mass fraction of the composition used to generate the equation of state table.

Referenced by eos(), readAscii(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.9 double **eos::dYMassFrac**

Helium mass fraction of the composition used to generate the equation of state table.

Referenced by eos(), readAscii(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.10 int **eos::nNumRho**

Number of densities in the equation of state table

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSound-Speed(), eos(), gamma1DelAdC_v(), getDlnPDlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappa-GammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), write-Ascii(), writeBin(), and ~eos().

3.1.4.11 int **eos::nNumT**

Number of temperatures in the equation of state table

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSound-Speed(), eos(), gamma1DelAdC_v(), getDlnPDlnTDlnPDlnPDEDT(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappa-GammaCp(), getPKappaGamma(), operator=(), readAscii(), readBobsAscii(), write-Ascii(), and writeBin().

The documentation for this class was generated from the following files:

- /home/cgeroux/Documents/WORK/SPHERLS/src/eos.h
- /home/cgeroux/Documents/WORK/SPHERLS/src/eos.cpp

3.2 exception2 Class Reference

The documentation for this class was generated from the following files:

- /home/cgeroux/Documents/WORK/SPHERLS/src/exception2.h
- /home/cgeroux/Documents/WORK/SPHERLS/src/exception2.cpp

Chapter 4

File Documentation

4.1 /home/cgeroux/Documents/WORK/SPHERLS/src/eos.cpp File - Reference

```
#include <string> #include <fstream> #include <sstream>
#include <iostream> #include <cmath> #include <stdlib.-
h> #include "eos.h" #include "exception2.h"
```

4.1.1 Detailed Description

Implements the eos (equation of state) class defined in eos.h

4.2 /home/cgeroux/Documents/WORK/SPHERLS/src/eos.h File - Reference

```
#include <string> #include "exception2.h"
```

Classes

- class eos

4.2.1 Detailed Description

Header file for eos.cpp

Index