SPHERLSgen Reference Manual

1.0

Generated by Doxygen 1.3.9.1

Tue Jan 10 13:58:06 2012

Contents

Chapter 1

SPHERLSgen Class Index

1.1 SPHERLSgen Class List

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

Chapter 2

SPHERLSgen File Index

2.1 SPHERLSgen File List

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

Chapter 3

SPHERLSgen 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 &dKappa)
- void getPEKappaGamma (double dT, double dRho, double &dP, double &dE, double &dKappa, double &dGamma)
- 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)

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

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

Constructor, allocates memory for the 2D arrays

Parameters:

← ***nNumT*** number of temperatures in the equaiton of state table

← ***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

3.1.2.4 eos::∼eos ()

Destructor, delets dynamic arrays

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:

← ***dT*** temperature at which the derivative is to be computed

← ***dRho*** density at which the derivative is to be computed

Returns:

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

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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

Returns:

the interpolated energy.

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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

Returns:

the interpolated opacity.

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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

Returns:

the interpolated pressure.

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

This function calculates the adiabatic sound speed

Parameters:

← ***dT*** temperature at which the derivative is to be computed

← ***dRho*** density at which the derivative is to be computed

Returns:

the sound speed.

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:

← ***dT*** temperature at which the derivative is to be computed

← ***dRho*** density at which the derivative is to be computed

→ ***dGamma1*** gamma1

→ ***dDelAd*** adiabatic gradient

→ ***dC_v*** specific heat at constant volume

3.1.3.7 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:

← ***dT*** temperature at which the derivative is to be computed

← ***dRho*** density at which the derivative is to be computed

→ ***dE*** energy at dT and dRho

→ ***dDTDE*** derivative of temperature w.r.t. energy at constant density

3.1.3.8 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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

→ ***dE*** energy at dT and dRho.

→ ***dKappa*** opacity at dT and dRho.

3.1.3.9 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:

← ***dT*** temperature at which the derivative is to be computed

← ***dRho*** density at which the derivative is to be computed

→ ***dP*** pressure at dT and dRho

→ ***dDRhoDP*** derivative of density w.r.t. pressure at conatant temperature

3.1.3.10 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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

→ ***dP*** pressure at dT and dRho.

→ ***dE*** energy at dT and dRho.

→ ***dKappa*** opacity at dT and dRho.

3.1.3.11 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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

→ ***dP*** pressure at dT and dRho.

→ ***dE*** energy at dT and dRho.

→ ***dKappa*** opacity at dT and dRho.

→ ***dGamma*** adiabatic index at dT and dRho.

3.1.3.12 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:

← ***dT*** temperature to interpolate to.

← ***dRho*** density to interpolate to.

→ ***dP*** pressure at dT and dRho.

→ ***dKappa*** opacity at dT and dRho.

→ ***dGamma*** adiabatic index at dT and dRho.

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

Assignment operator, assigns one eos object to another.

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

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

Parameters:

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

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

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

Parameters:

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

3.1.3.16 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:

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

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

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

Parameters:

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

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

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

Parameters:

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

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.

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.

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.

3.1.4.4 double eos::dLogRhoDelta

Increment of the density between table entries in log10.

3.1.4.5 double eos::dLogRhoMin

Minimum density of the table in log10.

3.1.4.6 double eos::dLogTDelta

Increment of the temperature between table entries in log10.

3.1.4.7 double eos::dLogTMin

Minimum temperature of the table in log10.

3.1.4.8 double eos::dXMassFrac

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

3.1.4.9 double eos::dYMassFrac

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

3.1.4.10 int eos::nNumRho

Number of densities in the equation of state table

3.1.4.11 int eos::nNumT

Number of temperatures in the equation of state table

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

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

Chapter 4

SPHERLSgen File Documentation

4.1 /home/cgeroux/SPHERLS/src/eos.cpp File Reference

```
#include <string>
#include <fstream>
#include <sstream>
#include <iostream>
#include <cmath>
#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/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