

SPHERLSgen

1.0

Generated by Doxygen 1.5.6

Fri Apr 20 13:37:49 2012

Contents

1	Class Index	1
1.1	Class List	1
2	File Index	3
2.1	File List	3
3	Class Documentation	5
3.1	eos Class Reference	5
3.1.1	Detailed Description	6
3.1.2	Constructor & Destructor Documentation	6
3.1.2.1	eos	6
3.1.2.2	eos	6
3.1.2.3	eos	6
3.1.2.4	~eos	6
3.1.3	Member Function Documentation	7
3.1.3.1	operator=	7
3.1.3.2	readAscii	7
3.1.3.3	readBobsAscii	7
3.1.3.4	writeAscii	7
3.1.3.5	readBin	7
3.1.3.6	writeBin	8
3.1.3.7	dGetPressure	8
3.1.3.8	dGetEnergy	8
3.1.3.9	dGetOpacity	8
3.1.3.10	dDRhoDP	9
3.1.3.11	dSoundSpeed	9
3.1.3.12	getEKappa	9
3.1.3.13	getPEKappa	10
3.1.3.14	getPEKappaGamma	10

3.1.3.15	getPEKappaGammaCp	10
3.1.3.16	getPKappaGamma	11
3.1.3.17	gamma1DelAdC_v	11
3.1.3.18	getPAndDRhoDP	11
3.1.3.19	getEAndDTDE	12
3.1.4	Member Data Documentation	12
3.1.4.1	nNumRho	12
3.1.4.2	nNumT	12
3.1.4.3	dXMassFrac	12
3.1.4.4	dYMassFrac	12
3.1.4.5	dLogRhoMin	13
3.1.4.6	dLogRhoDelta	13
3.1.4.7	dLogTMin	13
3.1.4.8	dLogTDelta	13
3.1.4.9	dLogP	13
3.1.4.10	dLogE	13
3.1.4.11	dLogKappa	14
4	File Documentation	15
4.1	/home/cgeroux/SPHERLS/src/eos.cpp File Reference	15
4.1.1	Detailed Description	15
4.2	/home/cgeroux/SPHERLS/src/eos.h File Reference	16
4.2.1	Detailed Description	16

Chapter 1

Class Index

1.1 Class List

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

eos	5
---------------------	-------	-------------------

Chapter 2

File Index

2.1 File List

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

/home/cgeroux/SPHERLS/src/ eos.cpp	15
/home/cgeroux/SPHERLS/src/ eos.h	16
/home/cgeroux/SPHERLS/src/ exception2.h	??
/home/cgeroux/SPHERLS/src/ xmlFunctions.h	??

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 &dKappa)
- 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)

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:

- ← ***nNumT*** number of temperatures in the equation of state table
- ← ***nNumRho*** number of densities in the equation 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](#), [dLogTMin](#), [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 eos & eos::operator= (const eos & *eosRightSide*)

Assignment operator, assigns one eos object to another.

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

3.1.3.2 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.

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

3.1.3.3 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.

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

3.1.3.4 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.

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

3.1.3.5 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.

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

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

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

Parameters:

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

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

3.1.3.7 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.

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

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

This function linearly interpolates the energy 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 energy.

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

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

This function linearly interpolates the 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.

Returns:

the interpolated opacity.

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

3.1.3.10 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.

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

3.1.3.11 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.

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

3.1.3.12 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*.

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

3.1.3.13 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*.

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

3.1.3.14 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*.

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

3.1.3.15 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:

- ← *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*.
- *dCp* specific heat at constant pressure at *dT* and *dRho*.

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

3.1.3.16 `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*.

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

3.1.3.17 `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

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

3.1.3.18 `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 constant temperature

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

3.1.3.19 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

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

3.1.4 Member Data Documentation

3.1.4.1 int eos::nNumRho

Number of densities in the equation of state table

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), writeBin(), and ~eos().

3.1.4.2 int eos::nNumT

Number of temperatures in the equation of state table

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.3 double eos::dXMassFrac

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

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

3.1.4.4 double eos::dYMassFrac

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

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

3.1.4.5 double eos::dLogRhoMin

Minimum density of the table in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.6 double eos::dLogRhoDelta

Increment of the density between table entries in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.7 double eos::dLogTMin

Minimum temperature of the table in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.8 double eos::dLogTDelta

Increment of the temperature between table entries in log10.

Referenced by dDRhoDP(), dGetEnergy(), dGetOpacity(), dGetPressure(), dSoundSpeed(), eos(), gamma1DelAdC_v(), getEAndDTDE(), getEKappa(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), and writeBin().

3.1.4.9 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(), getPAndDRhoDP(), getPEKappa(), getPEKappaGamma(), getPEKappaGammaCp(), getPKappaGamma(), operator=(), readAscii(), readBin(), readBobsAscii(), writeAscii(), writeBin(), and ~eos().

3.1.4.10 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()`, `getEAndDTDE()`, `getEKappa()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBin()`, `readBobsAscii()`, `writeAscii()`, `writeBin()`, and `~eos()`.

3.1.4.11 `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()`, `getEKappa()`, `getPEKappa()`, `getPEKappaGamma()`, `getPEKappaGammaCp()`, `getPKappaGamma()`, `operator=()`, `readAscii()`, `readBin()`, `readBobsAscii()`, `writeAscii()`, `writeBin()`, and `~eos()`.

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

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

~eos
 eos, [6](#)
 /home/cgeroux/SPHERLS/src/eos.cpp, [15](#)
 /home/cgeroux/SPHERLS/src/eos.h, [16](#)

dDRhoDP
 eos, [9](#)

dGetEnergy
 eos, [8](#)

dGetOpacity
 eos, [8](#)

dGetPressure
 eos, [8](#)

dLogE
 eos, [13](#)

dLogKappa
 eos, [14](#)

dLogP
 eos, [13](#)

dLogRhoDelta
 eos, [13](#)

dLogRhoMin
 eos, [12](#)

dLogTDelta
 eos, [13](#)

dLogTMin
 eos, [13](#)

dSoundSpeed
 eos, [9](#)

dXMassFrac
 eos, [12](#)

dYMassFrac
 eos, [12](#)

eos, [5](#)
 ~eos, [6](#)
 dDRhoDP, [9](#)
 dGetEnergy, [8](#)
 dGetOpacity, [8](#)
 dGetPressure, [8](#)
 dLogE, [13](#)
 dLogKappa, [14](#)
 dLogP, [13](#)
 dLogRhoDelta, [13](#)
 dLogRhoMin, [12](#)
 dLogTDelta, [13](#)
 dLogTMin, [13](#)
 dSoundSpeed, [9](#)
 dXMassFrac, [12](#)
 dYMassFrac, [12](#)
 eos, [6](#)
 gamma1DelAdC_v, [11](#)
 getEAndDTDE, [11](#)
 getEKappa, [9](#)
 getPAndDRhoDP, [11](#)
 getPEKappa, [9](#)
 getPEKappaGamma, [10](#)
 getPEKappaGammaCp, [10](#)
 getPKappaGamma, [10](#)
 nNumRho, [12](#)
 nNumT, [12](#)
 operator=, [7](#)
 readAscii, [7](#)
 readBin, [7](#)
 readBobsAscii, [7](#)
 writeAscii, [7](#)
 writeBin, [7](#)

gamma1DelAdC_v
 eos, [11](#)

getEAndDTDE
 eos, [11](#)

getEKappa
 eos, [9](#)

getPAndDRhoDP
 eos, [11](#)

getPEKappa
 eos, [9](#)

getPEKappaGamma
 eos, [10](#)

getPEKappaGammaCp
 eos, [10](#)

getPKappaGamma
 eos, [10](#)

nNumRho
 eos, [12](#)

nNumT
 eos, [12](#)

operator=

eos, [7](#)

readAscii

 eos, [7](#)

readBin

 eos, [7](#)

readBobsAscii

 eos, [7](#)

writeAscii

 eos, [7](#)

writeBin

 eos, [7](#)