

## Phase-field Solver for Eutectic Transformation

Generated by Doxygen 1.9.1



<b>1 Code-guide for solver eutectic</b>	<b>1</b>
1.1 Compiling the solver	1
1.2 Further details	1
<b>2 File Index</b>	<b>3</b>
2.1 File List	3
<b>3 File Documentation</b>	<b>5</b>
3.1 PhaseFieldSolverEutectic/eutectic/createFields.H File Reference	5
3.1.1 Function Documentation	6
3.1.1.1 phi_alpha()	6
3.1.1.2 phi_beta()	6
3.1.1.3 phi_liq()	6
3.1.1.4 T()	7
3.1.1.5 mu()	7
3.2 PhaseFieldSolverEutectic/eutectic/createTol.H File Reference	8
3.2.1 Function Documentation	8
3.2.1.1 Tol()	8
3.2.2 Variable Documentation	8
3.2.2.1 get_Tol_from_this	9
3.3 PhaseFieldSolverEutectic/eutectic/eutectic.C File Reference	9
3.3.1 Function Documentation	9
3.3.1.1 main()	9
3.4 PhaseFieldSolverEutectic/eutectic/info.md File Reference	11
3.5 PhaseFieldSolverEutectic/eutectic/phi_abl_antiT.H File Reference	11
3.5.1 Function Documentation	12
3.5.1.1 phi_alphaEqn()	13
3.5.1.2 phi_betaEqn()	13
3.5.1.3 phi_liqEqn()	13
3.5.1.4 while()	14
3.5.2 Variable Documentation	14
3.5.2.1 initial_residual_alpha	14
3.5.2.2 initial_residual_beta	14
3.5.2.3 initial_residual_liq	14
3.5.2.4 counter	14
3.5.2.5 do	14
3.5.2.6 grad_beta	15
3.5.2.7 grad_liq	15
3.6 PhaseFieldSolverEutectic/eutectic/readTransportProperties.H File Reference	15
3.6.1 Function Documentation	16
3.6.1.1 transportProperties()	16
3.6.1.2 dimt()	16
3.6.1.3 dimx()	17

---

3.6.1.4 ms_alpha()	17
3.6.1.5 ms_beta()	18
3.6.1.6 c_eq_liq()	18
3.6.1.7 c_eq_alpha()	19
3.6.1.8 c_eq_beta()	19
3.6.1.9 G()	20
3.6.1.10 v()	20
3.6.1.11 A()	20
3.6.1.12 D()	21
3.6.1.13 T_eut()	21
3.6.1.14 initial()	21
3.6.1.15 tau()	22
3.6.1.16 gamma()	22
3.6.1.17 epsilon()	22
3.7 PhaseFieldSolverEutectic/eutectic/set_delta_t.H File Reference	22
3.7.1 Function Documentation	22
3.7.1.1 setDeltaT()	23
3.7.2 Variable Documentation	23
3.7.2.1 maxResidual	23
3.7.2.2 maxDeltaT	23
3.7.2.3 maxDeltaTFactor	23
3.7.2.4 deltaTFactor	23
<b>Index</b>	<b>25</b>

# Chapter 1

## Code-guide for solver eutectic

The solver has been successfully tested using OpenFOAM v6.

### 1.1 Compiling the solver

- Following commands should create the executable of the solver

```
cd $FOAM_RUN/PhaseFieldSolverEutectic/eutectic
```

```
wclean
```

```
wmake
```

- The solver can be run by following the instructions in *userGuide*.

### 1.2 Further details

The implementation, client and header files of the solver have been written following OpenFOAM conventions. These are explained next with flow charts generated from the source code using Doxygen. It must be noted that the solver is based on `laplacianFoam` solver within OpenFOAM. Hence, it may be helpful for the user to become familiar with `OpenFOAM Programmer's Guide` and `laplacianFoam` beforehand.



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

PhaseFieldSolverEutectic/eutectic/ <a href="#">createFields.H</a> . . . . .	5
PhaseFieldSolverEutectic/eutectic/ <a href="#">createTol.H</a> . . . . .	8
PhaseFieldSolverEutectic/eutectic/ <a href="#">eutectic.C</a> . . . . .	9
PhaseFieldSolverEutectic/eutectic/ <a href="#">phi_abl_antiT.H</a> . . . . .	11
PhaseFieldSolverEutectic/eutectic/ <a href="#">readTransportProperties.H</a> . . . . .	15
PhaseFieldSolverEutectic/eutectic/ <a href="#">set_delta_t.H</a> . . . . .	22





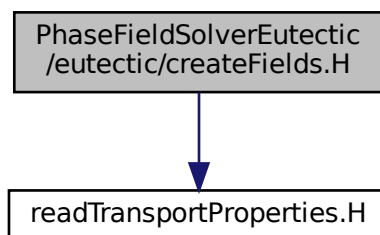
## Chapter 3

# File Documentation

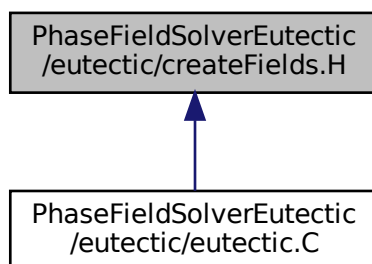
### 3.1 PhaseFieldSolverEutectic/eutectic/createFields.H File Reference

```
#include "readTransportProperties.H"
```

Include dependency graph for createFields.H:



This graph shows which files directly or indirectly include this file:



## Functions

- `volScalarField phi_alpha` (`IObject("phi_alpha", runTime.timeName(), mesh, IObject::MUST_READ, IObject::AUTO_WRITE), mesh`)

*Creating phase-fields with the option to write.*

- `volScalarField phi_beta` (`IObject("phi_beta", runTime.timeName(), mesh, IObject::MUST_READ, IObject::AUTO_WRITE), mesh`)
- `volScalarField phi_liq` (`IObject("phi_liq", runTime.timeName(), mesh, IObject::MUST_READ, IObject::AUTO_WRITE), mesh`)
- `volScalarField T` (`IObject("T", runTime.timeName(), mesh, IObject::MUST_READ, IObject::AUTO_WRITE), mesh`)

*Creating temperature field with the option to write.*

- `volScalarField mu` (`IObject("mu", runTime.timeName(), mesh, IObject::MUST_READ, IObject::AUTO_WRITE), mesh`)

*Creating chemical potential field with the option to write.*

### 3.1.1 Function Documentation

#### 3.1.1.1 phi\_alpha()

```
volScalarField phi_alpha (
    IObject("phi_alpha", runTime.timeName(), mesh, IObject::MUST_READ, IObject::
AUTO_WRITE) ,
    mesh )
```

Creating phase-fields with the option to write.

#### 3.1.1.2 phi\_beta()

```
volScalarField phi_beta (
    IObject("phi_beta", runTime.timeName(), mesh, IObject::MUST_READ, IObject::
AUTO_WRITE) ,
    mesh )
```

#### 3.1.1.3 phi\_liq()

```
volScalarField phi_liq (
    IObject("phi_liq", runTime.timeName(), mesh, IObject::MUST_READ, IObject::
AUTO_WRITE) ,
    mesh )
```

#### 3.1.1.4 T()

```
volScalarField T (
    IOobject("T", runTime.timeName(), mesh, IOobject::MUST_READ, IOobject::AUTO_↵
WRITE) ,
    mesh )
```

Creating temperature field with the option to write.

Referenced by main().

Here is the caller graph for this function:



#### 3.1.1.5 mu()

```
volScalarField mu (
    IOobject("mu", runTime.timeName(), mesh, IOobject::MUST_READ, IOobject::AUTO_↵
WRITE) ,
    mesh )
```

Creating chemical potential field with the option to write.

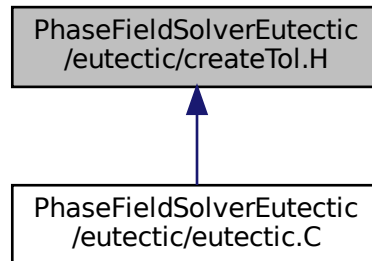
Referenced by main().

Here is the caller graph for this function:



## 3.2 PhaseFieldSolverEutectic/eutectic/createTol.H File Reference

This graph shows which files directly or indirectly include this file:



### Functions

- scalar [Tol](#) (readScalar(get\_Tol\_from\_this.lookup("Tol")))
   
Initial tolerance "Tol" within subDict "Tol\_is\_defined\_here" in fvSolution.

### Variables

- const dictionary & [get\\_Tol\\_from\\_this](#) = mesh.solutionDict().subDict("Tol\_is\_defined\_here")

### 3.2.1 Function Documentation

#### 3.2.1.1 Tol()

```

scalar Tol (
    readScalar(get_Tol_from_this.lookup("Tol")) )
  
```

Initial tolerance "Tol" within subDict "Tol\_is\_defined\_here" in fvSolution.

### 3.2.2 Variable Documentation

### 3.2.2.1 get\_Tol\_from\_this

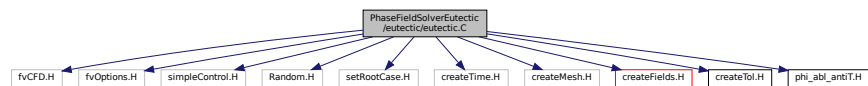
```
const dictionary& get_Tol_from_this = mesh.solutionDict().subDict("Tol_is_defined_here")
```

Definition at line 1 of file createTol.H.

## 3.3 PhaseFieldSolverEutectic/eutectic/eutectic.C File Reference

```
#include "fvCFD.H"
#include "fvOptions.H"
#include "simpleControl.H"
#include "Random.H"
#include "setRootCase.H"
#include "createTime.H"
#include "createMesh.H"
#include "createFields.H"
#include "createTol.H"
#include "phi_abl_antiT.H"
```

Include dependency graph for eutectic.C:



## Functions

- int [main](#) (int argc, char \*argv[ ])

### 3.3.1 Function Documentation

#### 3.3.1.1 main()

```
int main (
    int argc,
    char * argv[ ] )
```

The imposed temperature field as a function of thermal gradient in the x direction, G

Initial tolerance to check convergence

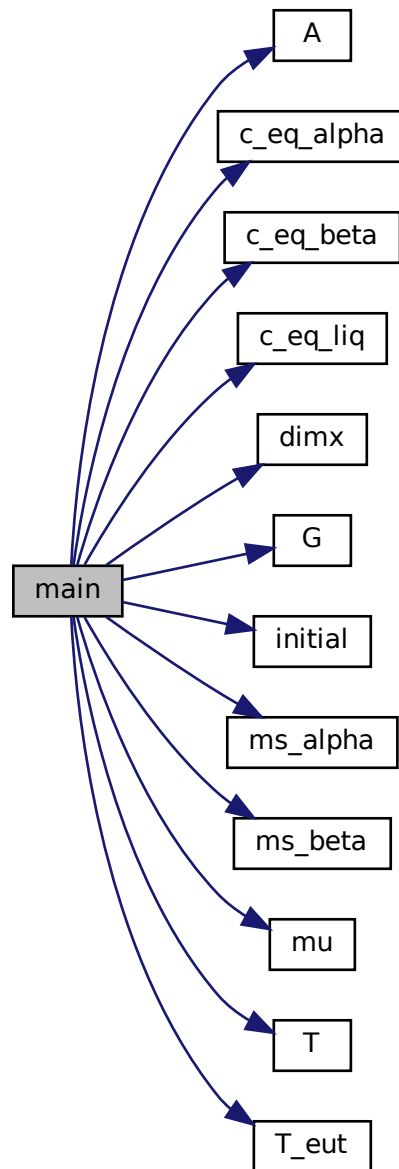
Solving the phase-field and chemical potential equations

Writing the results according to keywords in controlDict

Definition at line 39 of file eutectic.C.

References `A()`, `c_eq_alpha()`, `c_eq_beta()`, `c_eq_liq()`, `dimx()`, `G()`, `initial()`, `ms_alpha()`, `ms_beta()`, `mu()`, `T()`, and `T_eut()`.

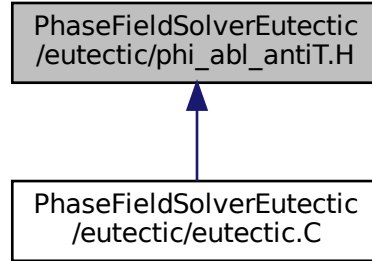
Here is the call graph for this function:



### 3.4 PhaseFieldSolverEutectic/eutectic/info.md File Reference

### 3.5 PhaseFieldSolverEutectic/eutectic/phi\_abl\_antiT.H File Reference

This graph shows which files directly or indirectly include this file:



### Functions

- fvScalarMatrix `phi_alphaEqn` (`tau * epsilon * dimt * fvm::ddt(phi_alpha) - v * grad_alpha.component(vector<::X>)=(-2.0/3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvm::laplacian(phi_alpha)+9.0 * gamma/epsilon * 2.0 * (phi_alpha) * (1.0-phi_alpha) * (1.0-2.0 * phi_alpha)+(-1.0/4.0 * (mu - B_alpha) * (mu - B_alpha)/A+D_alpha) * 0.25 * 15 * (3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha - 4.0 * phi_alpha * phi_alpha * phi_alpha - phi_alpha * phi_alpha + 2.0 * phi_alpha - (phi_beta - phi_liq) * (phi_beta - phi_liq) * (2.0 * phi_alpha - 3.0 * phi_alpha * phi_alpha)) + (-1.0/4.0 * (mu - B_beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - phi_liq)) + (-1.0/4.0 * (mu - B_liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)) + (1.0/3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_liq)+9.0 * gamma/epsilon * 2.0 * (phi_liq) * (1.0-phi_liq) * (1.0-2.0 * phi_liq) + (-1.0/4.0 * (mu - B_alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_liq - phi_beta)) + (-1.0/4.0 * (mu - B_beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi_alpha)) + (-1.0/4.0 * (mu - B_liq) * (mu - B_liq)/A+D_liq) * 0.25 * 15 * (3.0 * phi_liq * phi_liq * phi_liq * phi_liq - 4.0 * phi_liq * phi_liq * phi_liq - phi_liq * phi_liq + 2.0 * phi_liq - (phi_alpha - phi_beta) * (phi_alpha - phi_beta) * (2.0 * phi_liq - 3.0 * phi_liq * phi_liq)) + (1.0/3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_beta)+9.0 * gamma/epsilon * 2.0 * (phi_beta) * (1.0-phi_beta) * (1.0-2.0 * phi_beta) + (-1.0/4.0 * (mu - B_alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_beta - phi_liq)) + (-1.0/4.0 * (mu - B_beta) * (mu - B_beta)/A+D_beta) * 0.25 * 15 * (3.0 * phi_beta * phi_beta * phi_beta * phi_beta - 4.0 * phi_beta * phi_beta * phi_beta - phi_beta * phi_beta + 2.0 * phi_beta - (phi_liq - phi_alpha) * (phi_liq - phi_alpha) * (2.0 * phi_beta - 3.0 * phi_beta * phi_beta)) + (-1.0/4.0 * (mu - B_liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha))`)
- fvScalarMatrix `phi_betaEqn` (`tau * epsilon * dimt * fvm::ddt(phi_beta) - v * grad_beta.component(vector<::X>)=(-2.0/3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvm::laplacian(phi_beta)+9.0 * gamma/epsilon * 2.0 * (phi_beta) * (1.0-phi_beta) * (1.0-2.0 * phi_beta) + (-1.0/4.0 * (mu - B_alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_beta - phi_liq)) + (-1.0/4.0 * (mu - B_beta) * (mu - B_beta)/A+D_beta) * 0.25 * 15 * (3.0 * phi_beta * phi_beta * phi_beta * phi_beta - 4.0 * phi_beta * phi_beta * phi_beta - phi_beta * phi_beta + 2.0 * phi_beta - (phi_liq - phi_alpha) * (phi_liq - phi_alpha) * (2.0 * phi_beta - 3.0 * phi_beta * phi_beta)) + (-1.0/4.0 * (mu - B_liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)) + (1.0/3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvc::laplacian(phi_liq)+9.0 * gamma/epsilon * 2.0 * (phi_liq) * (1.0-phi_liq) * (1.0-2.0 * phi_liq) + (-1.0/4.0 * (mu -`

```

B_alpha)*(mu - B_alpha)/(A+D_alpha)*-0.5*15*((phi_alpha*phi_alpha - phi_alpha*phi_alpha*phi_alpha)
*(phi_liq - phi_beta))+(-1.0/4.0*(mu - B_beta)*(mu - B_beta)/(A+D_beta)*-0.5*15*((phi_beta*phi_beta
- phi_beta*phi_beta*phi_beta)*(phi_liq - phi_alpha))+(-1.0/4.0*(mu - B_liq)*(mu - B_liq)/(A+D_liq)*0.25
*15*(3.0*phi_liq*phi_liq*phi_liq*phi_liq - 4.0*phi_liq*phi_liq*phi_liq - phi_liq*phi_liq+2.0*phi_liq -
(phi_alpha - phi_beta)*(phi_alpha - phi_beta)*(2.0*phi_liq - 3.0*phi_liq*phi_liq)))+(1.0/3.0)*(-2.0*gamma
*epsilon*dimx*dimx*fvc::laplacian(phi_alpha)+9.0*gamma/epsilon*2.0*(phi_alpha)*(1.0-phi_alpha)
*(1.0-2.0*phi_alpha))+(-1.0/4.0*(mu - B_alpha)*(mu - B_alpha)/(A+D_alpha)*0.25*15*(3.0*phi_alpha
*phi_alpha*phi_alpha*phi_alpha - 4.0*phi_alpha*phi_alpha*phi_alpha - phi_alpha*phi_alpha+2.0
*phi_alpha -(phi_beta - phi_liq)*(phi_beta - phi_liq)*(2.0*phi_alpha - 3.0*phi_alpha*phi_alpha))+(-
1.0/4.0*(mu - B_beta)*(mu - B_beta)/(A+D_beta)*-0.5*15*((phi_beta*phi_beta - phi_beta*phi_beta
*phi_beta)*(phi_alpha - phi_liq))+(-1.0/4.0*(mu - B_liq)*(mu - B_liq)/(A+D_liq)*-0.5*15*((phi_liq*phi_liq
- phi_liq*phi_liq*phi_liq)*(phi_alpha - phi_beta))))
• fvScalarMatrix phi_liqEqn (tau*epsilon*dimt*fvm::ddt(phi_liq) - v*grad_liq.component(vector::X)==(-2.0/
0/3.0)*(-2.0*gamma*epsilon*dimx*dimx*fvm::laplacian(phi_liq)+9.0*gamma/epsilon*2.0*(phi_liq)
*(1.0-phi_liq)*(1.0-2.0*phi_liq))+(-1.0/4.0*(mu - B_alpha)*(mu - B_alpha)/(A+D_alpha)*-0.5*15
*((phi_alpha*phi_alpha - phi_alpha*phi_alpha*phi_alpha)*(phi_liq - phi_beta))+(-1.0/4.0*(mu - B_
beta)*(mu - B_beta)/(A+D_beta)*-0.5*15*((phi_beta*phi_beta - phi_beta*phi_beta*phi_beta)*(phi_liq
- phi_alpha))+(-1.0/4.0*(mu - B_liq)*(mu - B_liq)/(A+D_liq)*0.25*15*(3.0*phi_liq*phi_liq*phi_liq
*phi_liq - 4.0*phi_liq*phi_liq*phi_liq - phi_liq*phi_liq+2.0*phi_liq -(phi_alpha - phi_beta)*(phi_alpha
- phi_beta)*(2.0*phi_liq - 3.0*phi_liq*phi_liq)))+(1.0/3.0)*(-2.0*gamma*epsilon*dimx*dimx*fvc::
laplacian(phi_alpha)+9.0*gamma/epsilon*2.0*(phi_alpha)*(1.0-phi_alpha)*(1.0-2.0*phi_alpha))+(-1.0/
0/4.0*(mu - B_alpha)*(mu - B_alpha)/(A+D_alpha)*0.25*15*(3.0*phi_alpha*phi_alpha*phi_alpha
*phi_alpha - 4.0*phi_alpha*phi_alpha*phi_alpha - phi_alpha*phi_alpha+2.0*phi_alpha -(phi_beta
- phi_liq)*(phi_beta - phi_liq)*(2.0*phi_alpha - 3.0*phi_alpha*phi_alpha))+(-1.0/4.0*(mu - B_beta)
*(mu - B_beta)/(A+D_beta)*-0.5*15*((phi_beta*phi_beta - phi_beta*phi_beta*phi_beta)*(phi_alpha -
phi_liq))+(-1.0/4.0*(mu - B_liq)*(mu - B_liq)/(A+D_liq)*-0.5*15*((phi_liq*phi_liq - phi_liq*phi_liq*phi_liq)
*(phi_alpha - phi_beta)))+(1.0/3.0)*(-2.0*gamma*epsilon*dimx*dimx*fvc::laplacian(phi_beta)+9.0
*gamma/epsilon*2.0*(phi_beta)*(1.0-phi_beta)*(1.0-2.0*phi_beta))+(-1.0/4.0*(mu - B_alpha)*(mu -
B_alpha)/(A+D_alpha)*-0.5*15*((phi_alpha*phi_alpha - phi_alpha*phi_alpha*phi_alpha)*(phi_beta -
phi_liq))+(-1.0/4.0*(mu - B_beta)*(mu - B_beta)/(A+D_beta)*0.25*15*(3.0*phi_beta*phi_beta*phi_beta
*phi_beta - 4.0*phi_beta*phi_beta*phi_beta - phi_beta*phi_beta+2.0*phi_beta -(phi_liq - phi_alpha)
*(phi_liq - phi_alpha)*(2.0*phi_beta - 3.0*phi_beta*phi_beta))+(-1.0/4.0*(mu - B_liq)*(mu - B_liq)/(A+D_
liq)*-0.5*15*((phi_liq*phi_liq - phi_liq*phi_liq*phi_liq)*(phi_beta - phi_alpha))))
• while ((initial_residual_alpha.value() > Tol||initial_residual_beta.value() > Tol||initial_residual_liq.value() >
Tol) &&counter < 100)

```

## Variables

- dimensionedScalar `initial_residual_alpha` = `phi_alphaEqn.solve().max().initialResidual()`
- dimensionedScalar `initial_residual_beta` = `phi_betaEqn.solve().max().initialResidual()`
- dimensionedScalar `initial_residual_liq` = `phi_liqEqn.solve().max().initialResidual()`
- scalar `counter` = 0
- do
- volVectorField `grad_beta` = `dimx*fvc::grad(phi_beta)`
- volVectorField `grad_liq` = `dimx*fvc::grad(phi_liq)`

## 3.5.1 Function Documentation



### 3.5.1.1 phi\_alphaEqn()

```
fvScalarMatrix phi_alphaEqn (
    tau * epsilon * dimt * fvm::ddtphi_alpha) - v * grad_alpha.component(vector::X =
    =(-2.0/3.0) * (-2.0 * gamma * epsilon * dimx * dimx * fvm::laplacian(phi_alpha) + 9.0 * gamma/epsilon * 2.0 * (phi_alpha)
    _alpha) * (mu - B_alpha)/A+D_alpha) * 0.25 * 15 * (3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha - 4.0 * phi_alpha
    _beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - phi
    _liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
    ::laplacian(phi_liq) + 9.0 * gamma/epsilon * 2.0 * (phi_liq) * (1.0-phi_liq) * (1.0-2.0 * phi_liq) + (-1.0/4.0 * (mu - B
    _alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_l
    _beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi
    _liq) * (mu - B_liq)/A+D_liq) * 0.25 * 15 * (3.0 * phi_liq * phi_liq * phi_liq * phi_liq - 4.0 * phi_liq * phi_liq * phi_l
    ::laplacian(phi_beta) + 9.0 * gamma/epsilon * 2.0 * (phi_beta) * (1.0-phi_beta) * (1.0-2.0 * phi_beta) + (-1.0/4.0 * (mu -
    _alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_b
    _beta) * (mu - B_beta)/A+D_beta) * 0.25 * 15 * (3.0 * phi_beta * phi_beta * phi_beta * phi_beta - 4.0 * phi_beta * phi_be
    _liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
    )
```

### 3.5.1.2 phi\_betaEqn()

```
fvScalarMatrix phi_betaEqn (
    tau * epsilon * dimt * fvm::ddtphi_beta) - v * grad_beta.component(vector::X = (-2.0/3.0) * (-2.0 * gamma
    ::laplacian(phi_beta) + 9.0 * gamma/epsilon * 2.0 * (phi_beta) * (1.0-phi_beta) * (1.0-2.0 * phi_beta) + (-1.0/4.0 * (mu -
    _alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_b
    _beta) * (mu - B_beta)/A+D_beta) * 0.25 * 15 * (3.0 * phi_beta * phi_beta * phi_beta * phi_beta - 4.0 * phi_beta * phi_be
    _liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
    ::laplacian(phi_liq) + 9.0 * gamma/epsilon * 2.0 * (phi_liq) * (1.0-phi_liq) * (1.0-2.0 * phi_liq) + (-1.0/4.0 * (mu - B
    _alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_l
    _beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi
    _liq) * (mu - B_liq)/A+D_liq) * 0.25 * 15 * (3.0 * phi_liq * phi_liq * phi_liq * phi_liq - 4.0 * phi_liq * phi_liq * phi_l
    ::laplacian(phi_alpha) + 9.0 * gamma/epsilon * 2.0 * (phi_alpha) * (1.0-phi_alpha) * (1.0-2.0 * phi_alpha) + (-1.0/4.0 *
    _alpha) * (mu - B_alpha)/A+D_alpha) * 0.25 * 15 * (3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha - 4.0 * phi_alpha
    _beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - p
    _liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
    )
```

### 3.5.1.3 phi\_liqEqn()

```
fvScalarMatrix phi_liqEqn (
    tau * epsilon * dimt * fvm::ddtphi_liq) - v * grad_liq.component(vector::X = (-2.0/3.0) * (-2.0 * gamma
    ::laplacian(phi_liq) + 9.0 * gamma/epsilon * 2.0 * (phi_liq) * (1.0-phi_liq) * (1.0-2.0 * phi_liq) + (-1.0/4.0 * (mu - B
    _alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_l
    _beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_liq - phi
    _liq) * (mu - B_liq)/A+D_liq) * 0.25 * 15 * (3.0 * phi_liq * phi_liq * phi_liq * phi_liq - 4.0 * phi_liq * phi_liq * phi_l
    ::laplacian(phi_alpha) + 9.0 * gamma/epsilon * 2.0 * (phi_alpha) * (1.0-phi_alpha) * (1.0-2.0 * phi_alpha) + (-1.0/4.0 *
    _alpha) * (mu - B_alpha)/A+D_alpha) * 0.25 * 15 * (3.0 * phi_alpha * phi_alpha * phi_alpha * phi_alpha - 4.0 * phi_alpha
    _beta) * (mu - B_beta)/A+D_beta) * -0.5 * 15 * ((phi_beta * phi_beta - phi_beta * phi_beta * phi_beta) * (phi_alpha - p
    _liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_alpha - phi_beta)
    ::laplacian(phi_beta) + 9.0 * gamma/epsilon * 2.0 * (phi_beta) * (1.0-phi_beta) * (1.0-2.0 * phi_beta) + (-1.0/4.0 * (mu -
    _alpha) * (mu - B_alpha)/A+D_alpha) * -0.5 * 15 * ((phi_alpha * phi_alpha - phi_alpha * phi_alpha * phi_alpha) * (phi_b
    _beta) * (mu - B_beta)/A+D_beta) * 0.25 * 15 * (3.0 * phi_beta * phi_beta * phi_beta * phi_beta - 4.0 * phi_beta * phi_be
    _liq) * (mu - B_liq)/A+D_liq) * -0.5 * 15 * ((phi_liq * phi_liq - phi_liq * phi_liq * phi_liq) * (phi_beta - phi_alpha)
    )
```

#### 3.5.1.4 while()

```
while ( )
```

### 3.5.2 Variable Documentation

#### 3.5.2.1 initial\_residual\_alpha

```
initial_residual_alpha = phi_alphaEqn.solve().max().initialResidual()
```

Definition at line 2 of file phi\_abl\_antiT.H.

#### 3.5.2.2 initial\_residual\_beta

```
initial_residual_beta = phi_betaEqn.solve().max().initialResidual()
```

Definition at line 3 of file phi\_abl\_antiT.H.

#### 3.5.2.3 initial\_residual\_liq

```
initial_residual_liq = phi_liqEqn.solve().max().initialResidual()
```

Definition at line 4 of file phi\_abl\_antiT.H.

#### 3.5.2.4 counter

```
counter = 0
```

Definition at line 5 of file phi\_abl\_antiT.H.

#### 3.5.2.5 do

```
do
```

##### Initial value:

```
{
    volVectorField grad_alpha =dimx*fvc::grad(phi_alpha)
```

Implicit discretization using fvm class for time derivative and laplacian. Explicit discretization using fvc class for gradient and divergence. Phase-field equations with approximate relation between c, mu and T according to the parabolic approximation for free energy

Definition at line 9 of file phi\_abl\_antiT.H.

### 3.5.2.6 grad\_beta

```
volVectorField grad_beta =dimx*fvc::grad(phi_beta)
```

Definition at line 11 of file phi\_abl\_antiT.H.

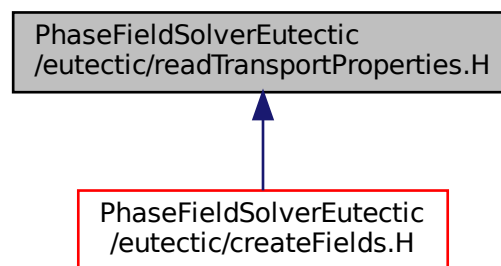
### 3.5.2.7 grad\_liq

```
volVectorField grad_liq =dimx*fvc::grad(phi_liq)
```

Definition at line 12 of file phi\_abl\_antiT.H.

## 3.6 PhaseFieldSolverEutectic/eutectic/readTransportProperties.H File Reference

This graph shows which files directly or indirectly include this file:



## Functions

- IOdictionary [transportProperties](#) (IOobject("transportProperties", runTime.constant(), mesh, IOobject::MUST\_READ, IOobject::NO\_WRITE))  
*The input properties to be exported to createFields.*
- dimensionedScalar [dimt](#) (transportProperties.lookup("dimt"))  
*The input properties are read from constant/transportProperties dictionary.*
- dimensionedScalar [dimx](#) (transportProperties.lookup("dimx"))  
*Dimension of position.*
- dimensionedScalar [ms\\_alpha](#) (transportProperties.lookup("ms\_alpha"))  
*Slope liq-alpha and alpha-liq.*
- dimensionedScalar [ms\\_beta](#) (transportProperties.lookup("ms\_beta"))  
*Slope liq-beta and beta-liq.*
- dimensionedScalar [c\\_eq\\_liq](#) (transportProperties.lookup("c\_eq\_liq"))

- Eutectic composition of liquid phase.*
- dimensionedScalar **c\_eq\_alpha** (transportProperties.lookup("c\_eq\_alpha"))
- Eutectic composition of alpha phase.*
- dimensionedScalar **c\_eq\_beta** (transportProperties.lookup("c\_eq\_beta"))
- Eutectic composition of beta phase.*
- dimensionedScalar **G** (transportProperties.lookup("G"))
- Thermal gradient.*
- dimensionedScalar **v** (transportProperties.lookup("v"))
- Velocity.*
- dimensionedScalar **A** (transportProperties.lookup("A"))
- dimensionedScalar **D** (transportProperties.lookup("D"))
- Diffusivity in liquid.*
- dimensionedScalar **T\_eut** (transportProperties.lookup("T\_eut"))
- Eutectic temperature.*
- dimensionedScalar **initial** (transportProperties.lookup("initial"))
- Constant value from temperature profile.*
- dimensionedScalar **tau** (transportProperties.lookup("tau"))
- Relaxation coefficient.*
- dimensionedScalar **gamma** (transportProperties.lookup("gamma"))
- Interface energy parameter.*
- dimensionedScalar **epsilon** (transportProperties.lookup("epsilon"))
- Interface width parameter.*

### 3.6.1 Function Documentation

#### 3.6.1.1 transportProperties()

```
IOdictionary transportProperties (
    IOobject("transportProperties", runTime.constant(), mesh, IOobject::MUST_READ,
    IOobject::NO_WRITE) )
```

The input properties to be exported to createFields.

#### 3.6.1.2 dimt()

```
dimensionedScalar dimt (
    transportProperties. lookup "dimt " )
```

The input properties are read from constant/transportProperties dictionary.

Dimension of time

### 3.6.1.3 dimx()

```
dimensionedScalar dimx (
    transportProperties.  lookup<"dimx"> )
```

Dimension of position.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.4 ms\_alpha()

```
dimensionedScalar ms_alpha (
    transportProperties.  lookup<"ms_alpha"> )
```

Slope liq-alpha and alpha-liq.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.5 ms\_beta()

```
dimensionedScalar ms_beta (
    transportProperties.  lookup"ms_beta" )
```

Slope liq-beta and beta-liq.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.6 c\_eq\_liq()

```
dimensionedScalar c_eq_liq (
    transportProperties.  lookup"c_eq_liq" )
```

Eutectic composition of liquid phase.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.7 c\_eq\_alpha()

```
dimensionedScalar c_eq_alpha (
    transportProperties. lookup<"c_eq_alpha"> )
```

Eutectic composition of alpha phase.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.8 c\_eq\_beta()

```
dimensionedScalar c_eq_beta (
    transportProperties. lookup<"c_eq_beta"> )
```

Eutectic composition of beta phase.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.9 G()

```
dimensionedScalar G (  
    transportProperties. lookup"G" )
```

Thermal gradient.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.10 v()

```
dimensionedScalar v (  
    transportProperties. lookup"v" )
```

Velocity.

### 3.6.1.11 A()

```
dimensionedScalar A (  
    transportProperties. lookup"A" )
```

Referenced by main().

Here is the caller graph for this function:





#### 3.6.1.12 D()

```
dimensionedScalar D (
    transportProperties.  lookup"D" )
```

Diffusivity in liquid.

#### 3.6.1.13 T\_eut()

```
dimensionedScalar T_eut (
    transportProperties.  lookup"T_eut" )
```

Eutectic temperature.

Referenced by main().

Here is the caller graph for this function:



#### 3.6.1.14 initial()

```
dimensionedScalar initial (
    transportProperties.  lookup"initial" )
```

Constant value from temperature profile.

Referenced by main().

Here is the caller graph for this function:



### 3.6.1.15 tau()

```
dimensionedScalar tau (
    transportProperties. lookup"tau" )
```

Relaxation coefficient.

### 3.6.1.16 gamma()

```
dimensionedScalar gamma (
    transportProperties. lookup"gamma" )
```

Interface energy parameter.

### 3.6.1.17 epsilon()

```
dimensionedScalar epsilon (
    transportProperties. lookup"epsilon" )
```

Interface width parameter.

## 3.7 PhaseFieldSolverEutectic/eutectic/set\_delta\_t.H File Reference

### Functions

- runTime [setDeltaT](#) (min([deltaTFactor](#) \*runTime.deltaTValue(), [maxDeltaT](#)))

### Variables

- dimensionedScalar [maxResidual](#) = 1e-7  
*Timestep reduction for improving convergence.*
- dimensionedScalar [maxDeltaT](#) = 1
- dimensionedScalar [maxDeltaTFactor](#) = [maxResidual](#)/( [initial\\_residual\\_liq](#))
- dimensionedScalar [deltaTFactor](#) = min(min([maxDeltaTFactor](#), 1.0 + 0.1\*[maxDeltaTFactor](#)), 1.↵  
2\*([maxDeltaTFactor](#)/[maxDeltaTFactor](#)) )

### 3.7.1 Function Documentation

### 3.7.1.1 setDeltaT()

```
runTime.setDeltaT (
    min(deltaTFactor *runTime.deltaTValue(), maxDeltaT) )
```

## 3.7.2 Variable Documentation

### 3.7.2.1 maxResidual

```
dimensionedScalar maxResidual = 1e-7
```

Timestep reduction for improving convergence.

Definition at line 36 of file set\_delta\_t.H.

### 3.7.2.2 maxDeltaT

```
dimensionedScalar maxDeltaT = 1
```

Definition at line 37 of file set\_delta\_t.H.

### 3.7.2.3 maxDeltaTFactor

```
dimensionedScalar maxDeltaTFactor = maxResidual/( initial_residual_liq)
```

Definition at line 38 of file set\_delta\_t.H.

### 3.7.2.4 deltaTFactor

```
dimensionedScalar deltaTFactor = min(min(maxDeltaTFactor, 1.0 + 0.1*maxDeltaTFactor), 1.↵
2*(maxDeltaTFactor/maxDeltaTFactor) )
```

Definition at line 39 of file set\_delta\_t.H.



# Index

## A

readTransportProperties.H, [20](#)

c\_eq\_alpha

readTransportProperties.H, [18](#)

c\_eq\_beta

readTransportProperties.H, [19](#)

c\_eq\_liq

readTransportProperties.H, [18](#)

counter

phi\_abl\_antiT.H, [14](#)

createFields.H

mu, [7](#)

phi\_alpha, [6](#)

phi\_beta, [6](#)

phi\_liq, [6](#)

T, [6](#)

createTol.H

get\_Tol\_from\_this, [8](#)

Tol, [8](#)

## D

readTransportProperties.H, [20](#)

deltaTFactor

set\_delta\_t.H, [23](#)

dimt

readTransportProperties.H, [16](#)

dimx

readTransportProperties.H, [16](#)

do

phi\_abl\_antiT.H, [14](#)

epsilon

readTransportProperties.H, [22](#)

eutectic.C

main, [9](#)

## G

readTransportProperties.H, [19](#)

gamma

readTransportProperties.H, [22](#)

get\_Tol\_from\_this

createTol.H, [8](#)

grad\_beta

phi\_abl\_antiT.H, [14](#)

grad\_liq

phi\_abl\_antiT.H, [15](#)

initial

readTransportProperties.H, [21](#)

initial\_residual\_alpha

phi\_abl\_antiT.H, [14](#)

initial\_residual\_beta

phi\_abl\_antiT.H, [14](#)

initial\_residual\_liq

phi\_abl\_antiT.H, [14](#)

main

eutectic.C, [9](#)

maxDeltaT

set\_delta\_t.H, [23](#)

maxDeltaTFactor

set\_delta\_t.H, [23](#)

maxResidual

set\_delta\_t.H, [23](#)

ms\_alpha

readTransportProperties.H, [17](#)

ms\_beta

readTransportProperties.H, [17](#)

mu

createFields.H, [7](#)

PhaseFieldSolverEutectic/eutectic/createFields.H, [5](#)

PhaseFieldSolverEutectic/eutectic/createTol.H, [8](#)

PhaseFieldSolverEutectic/eutectic/eutectic.C, [9](#)

PhaseFieldSolverEutectic/eutectic/info.md, [11](#)

PhaseFieldSolverEutectic/eutectic/phi\_abl\_antiT.H, [11](#)

PhaseFieldSolverEutectic/eutectic/readTransportProperties.H,  
[15](#)

PhaseFieldSolverEutectic/eutectic/set\_delta\_t.H, [22](#)

phi\_abl\_antiT.H

counter, [14](#)

do, [14](#)

grad\_beta, [14](#)

grad\_liq, [15](#)

initial\_residual\_alpha, [14](#)

initial\_residual\_beta, [14](#)

initial\_residual\_liq, [14](#)

phi\_alphaEqn, [12](#)

phi\_betaEqn, [13](#)

phi\_liqEqn, [13](#)

while, [13](#)

phi\_alpha

createFields.H, [6](#)

phi\_alphaEqn

phi\_abl\_antiT.H, [12](#)

phi\_beta

createFields.H, [6](#)

phi\_betaEqn

phi\_abl\_antiT.H, [13](#)

phi\_liq

- createFields.H, 6
- phi\_liqEqn
  - phi\_abl\_antiT.H, 13
- readTransportProperties.H
  - A, 20
  - c\_eq\_alpha, 18
  - c\_eq\_beta, 19
  - c\_eq\_liq, 18
  - D, 20
  - dimt, 16
  - dimx, 16
  - epsilon, 22
  - G, 19
  - gamma, 22
  - initial, 21
  - ms\_alpha, 17
  - ms\_beta, 17
  - T\_eut, 21
  - tau, 21
  - transportProperties, 16
  - v, 20
- set\_delta\_t.H
  - deltaTFactor, 23
  - maxDeltaT, 23
  - maxDeltaTFactor, 23
  - maxResidual, 23
  - setDeltaT, 22
- setDeltaT
  - set\_delta\_t.H, 22
- T
  - createFields.H, 6
- T\_eut
  - readTransportProperties.H, 21
- tau
  - readTransportProperties.H, 21
- Tol
  - createTol.H, 8
- transportProperties
  - readTransportProperties.H, 16
- v
  - readTransportProperties.H, 20
- while
  - phi\_abl\_antiT.H, 13