

```

/*-----*|
=====
|| / F i e l d      | OpenFOAM: The Open Source CFD Toolbox
|| / O p e r a t i o n |
|| / A n d           | www.openfoam.com
|| V M a n i p u l a t i o n |

```

Copyright (C) 2014-2015 OpenFOAM Foundation

License

This file is part of OpenFOAM.

OpenFOAM is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

OpenFOAM is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with OpenFOAM. If not, see <<http://www.gnu.org/licenses/>>.

```

|*-----*/

```

```

#include "fvMatrices.H"
#include "fvcDdt.H"
#include "fvcDiv.H"
#include "basicThermo.H"

```

```

// ***** Private Member Functions ***** //

```

```

template<class RhoFieldType>
void Foam::fv::mySolidificationMeltingSource::apply
(
    const RhoFieldType& rho,
    fvMatrix<scalar>& eqn
)
{
    if (debug)
    {
        Info<< type() << ": applying source to " << eqn.psi().name() << endl;
    }

    update();
    const auto& CpVoF = mesh_.lookupObject<volScalarField>(CpName_);
    const auto& rhoCpPhiVoF = mesh_.lookupObject<surfaceScalarField>(rhoCpPhiName_);
    dimensionedScalar L("L", dimEnergy/dimMass, L_);

    // contributions added to rhs of solver equation
    if (eqn.psi().dimensions() == dimTemperature)
    {

```

```

    eqn -= L/CpVoF*(fvc::ddt(rho, alpha1_) + fvc::div(rhoCpPhiVoF, alpha1_));
}
else
{
    //This option is not activated since fvOptions in TEqn does not enable this condition
    eqn -= L*(fvc::ddt(rho, alpha1_));
}
}

// ***** //

```