```
/*---------------------------------------------------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     |
    \\  /    A nd           | www.openfoam.com
     \\/     M anipulation  |
-------------------------------------------------------------------------------
    Copyright (C) 2017-2020 OpenCFD Ltd.
-------------------------------------------------------------------------------
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/>.

Class
    Foam::meltingEvaporationModels::LeeCNT

Description
    Mass transfer LeeCNT model. Simple model driven by field value difference as:

    \f[
        \dot{m} = C \rho \alpha (T - T_{activate})/T_{activate}
    \f]

    where C is a model constant.

    if C > 0:
    \f[
        \dot{m} = C \rho \alpha (T - T_{activate})/T_{activate}
    \f]
        for \f[ T > T_{activate} \f]

    and

        \f[ mDot = 0.0 \f]  for \f[  T < T_{activate} \f]


    if C < 0:
    \f[
        \dot{m} = -C \rho \alpha (T_{activate} - T)/T_{activate}
    \f]
        for \f[ T < T_{activate} \f]

    and
```

```
        \f[ \dot{m} = 0.0 \f] for \f[ T > T_{activate} \f]

    Based on the reference:
    -# W. H. LeeCNT. "A Pressure Iteration Scheme for Two-Phase Modeling".
    Technical Report LA-UR 79-975. Los Alamos Scientific Laboratory,
    Los Alamos, New Mexico. 1979.

Usage
    Example usage:
    \verbatim
        massTransferModel
        (
            (solid to liquid)
            {
                type            LeeCNT;
                C               40;
                Tactivate       302.78;
            }
        );
    \endverbatim

    Where:

    \table
        Property    | Description            | Required    | Default value
        Tactivate   | Activation temperature | yes
        C           | Model constant         | yes
        includeVolChange    | Volumen change  | no          | yes
        species     | Specie name on the other phase | no   | none
    \endtable

SourceFiles
    LeeCNT.C

\*---------------------------------------------------------------------------*/

#ifndef meltingEvaporationModels_LeeCNT_H
#define meltingEvaporationModels_LeeCNT_H

#include "InterfaceCompositionModel.H"

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

namespace Foam
{
namespace meltingEvaporationModels
{

/*---------------------------------------------------------------------------*\
                          Class LeeCNT Declaration
\*---------------------------------------------------------------------------*/

template<class Thermo, class OtherThermo>
class LeeCNT
:
```

```
    public InterfaceCompositionModel<Thermo, OtherThermo>
{
    // Private Data

        //- Condensation coefficient [1/s]
        dimensionedScalar C_;

        volScalarField interfaceVolume_;
        //- Phase transition temperature
        const dimensionedScalar Tactivate_;

        //- Phase minimum value for activation
        scalar alphaMin_;

        //- Planck constant [J.s]
        const dimensionedScalar planck_;

        //- Boltzmann constant [J/K]
        const dimensionedScalar boltzmann_;

        //- Activation energy of water molecules passing through water-ice interface [J]
        const dimensionedScalar deltag_;

        //- Number of water molecule in a water volume [m3]
        const dimensionedScalar nL_;

        //- Superficial free energy of the water-ice interface [J/m2]
        const dimensionedScalar gammaYW_;

        //- Latent heat per volume [J/m3]
        const dimensionedScalar hLV_;

        //- Shape coefficient of nucleation
        const dimensionedScalar alphaEY_;


public:

    //- Runtime type information
    TypeName("LeeCNT");


    // Constructors

        //- Construct from components
        LeeCNT
        (
            const dictionary& dict,
            const phasePair& pair
        );


    //- Destructor
    virtual ~LeeCNT() = default;
```

```
    // Member Functions

        //- Explicit total mass transfer coefficient
        virtual tmp<volScalarField> Kexp
        (
            const volScalarField& field
        );

        //- Implicit mass transfer coefficient
        virtual tmp<volScalarField> KSp
        (
            label modelVariable,
            const volScalarField& field
        );

        //- Explicit mass transfer coefficient
        virtual tmp<volScalarField> KSu
        (
            label modelVariable,
            const volScalarField& field
        );

        //- Return T transition between phases
        virtual const dimensionedScalar& Tactivate() const;

        //- Add/subtract alpha*div(U) as a source term
        //- for alpha, substituting div(U) = mDot(1/rho1 - 1/rho2)
        virtual bool includeDivU();

        virtual const dimensionedScalar& planck() const;

        virtual const dimensionedScalar& boltzmann() const;

        virtual const dimensionedScalar& deltag() const;

        virtual const dimensionedScalar& nL() const;

        virtual const dimensionedScalar& gammaYW() const;

        virtual const dimensionedScalar& hLV() const;

        virtual const dimensionedScalar& alphaEY() const;


};


// ************************************************************************* //

} // End namespace meltingEvaporationModels
} // End namespace Foam

// ************************************************************************* //
```

```
#ifdef NoRepository
#   include "LeeCNT.C"
#endif

// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //

#endif

// ************************************************************************* //
```