Multiphase KKS model in Moose

Johan Hektor

August 4, 2017

1 Introduction

This document contain notes on the implementation of the phase field model presented in [1] in Moose. The following special notation of volume and surface integrals will be used, in accordance with the official Moose documentation;

$$\int a \cdot b \, \mathrm{d}V = (a, b) \, , \qquad \int a \cdot b \, \mathrm{d}S = \langle a, b \rangle \tag{1}$$

The sets of phase field variables and phase compositions are denoted $\boldsymbol{\eta}$ and $\boldsymbol{x}$, respectively. The global concentration field is denoted c and h denotes the switching functions. The phase fields, phase compositions and the global concentration are all dimensionless fields in the range $[0, 1]$.

2 Model

2.1 Free energy

The free energy is given by

$$F = f_{ch}(\boldsymbol{\eta}, \boldsymbol{x}) + f_{int}(\boldsymbol{\eta}) + f_d(\boldsymbol{\eta}, \dots), \tag{2}$$ {{eq:F}}

with f_{ch} and f_{int} representing the chemical and interface energies respectively. f_d represents the contribution from any other physics included in the model e.g. deformation, temperature, electric field etc.

Here we take

$$f_{ch} = \sum_i h_i(\boldsymbol{\eta}) f_i(x_i, \boldsymbol{\eta}), \tag{3}$$ {{eq:fch}}

with $f_i(x_i, \boldsymbol{\eta}) = \frac{G_i(x_i)}{\bar{V}^m}$ where G_i is a representation of the Gibbs energy of phase i and $\bar{V}^m = \sum_j h_j V_j^m$ denotes the molar volume in the current integration point. The interface energy is taken as

$$f_{int} = m\left(\sum_i \left(\frac{\eta_i^4}{4} - \frac{\eta_i^2}{2}\right) + \sum_i \sum_{j\neq i} \frac{\beta}{2}\eta_i^2\eta_j^2 + \frac{1}{4}\right) + \frac{\alpha}{2}\sum_i \left(\boldsymbol{\nabla}\eta_i\right)^2, \qquad (4)$$ {{eq:fint}}

with m, α and β being parameters.

2.2 Governing equations

2.2.1 KKS conditions

KKS models require the chemical potential to be equal in all coexisting phases, i.e.

$$\frac{\partial F}{\partial c} = \frac{\partial f_1}{\partial x_1} = \cdots = \frac{\partial f_n}{\partial x_n} = \mu_1 \cdots = \mu_n. \qquad (5)$$ {{eq:KKS}}

Furthermore, mass conservation is fulfilled by

$$c = \sum_i h_i x_i. \qquad (6)$$ {{eq:mass}}

2.2.2 Cahn-Hilliard equation

The evolution of the concentration field follows the Cahn-Hilliard equation

$$\frac{\partial c}{\partial t} = \boldsymbol{\nabla}\cdot M(\boldsymbol{\eta}, c)\boldsymbol{\nabla}\frac{\delta F}{\delta c} = \boldsymbol{\nabla}\cdot M(\boldsymbol{\eta}, c)\boldsymbol{\nabla}\left(\frac{\partial F}{\partial c} - \boldsymbol{\nabla}\cdot\frac{\partial F}{\partial \boldsymbol{\nabla} c}\right), \qquad (7)$$ {{eq:CH1}}

where M is the mobility. Since there is no dependence on $\boldsymbol{\nabla} c$ in the free energy the divergence term of (7) vanishes.[1]

Using (2), (3) and (4) the derivative become

$$\frac{\partial F}{\partial c} = \frac{\partial f_{ch}}{\partial c} = \sum_i h_i \frac{\partial f_i}{\partial x_i}\frac{\partial x_i}{\partial c} + \cancelto{0}{\frac{\partial h_i}{\partial c}} f_i. \qquad (8)$$ {{eq:dFdc1}}

By taking the derivative of the mass conservation (6) we get

$$\frac{\partial c}{\partial c} = \sum_i h_i \frac{\partial x_i}{\partial c} = 1. \qquad (9)$$

[1] This is only true for KKS models, in other multiphase models a term like $\kappa\left(\boldsymbol{\nabla} c\right)^2$ is included. This term will make the Cahn-Hilliard a fourth order PDE.

Inserting into (8) and taking into account (5) we end up with

$$\frac{\partial F}{\partial c} = \mu, \tag{10}$$

where we can arbitrarily choose $\mu = \mu_i$ based on the KKS condition. With this (7) can be written as

$$\frac{\partial c}{\partial t} = \boldsymbol{\nabla} \cdot M(\boldsymbol{\eta}, c)\boldsymbol{\nabla}\mu_i \tag{11}$$

{{eq:CH}}

2.2.3 Allen-Cahn equation

The evolution of each order parameter field follow the Allen-Cahn equation

$$\begin{aligned} \frac{\partial \eta_i}{\partial t} &= -L(\boldsymbol{\eta})\frac{\delta F}{\delta \eta_i} = -L\left(\frac{\partial F}{\partial \eta_i} - \boldsymbol{\nabla} \cdot \frac{\partial F}{\partial \boldsymbol{\nabla}\eta_i}\right) \\ &= -L\left(\frac{\partial f_{ch}}{\partial \eta_i} + \frac{\partial f_{int}}{\partial \eta_i} + \frac{\partial f_d}{\partial \eta_i} - \boldsymbol{\nabla}\frac{\partial f_{int}}{\partial \boldsymbol{\nabla}\eta_i}\right). \end{aligned} \tag{12}$$

{{eq:AC1}}

The derivatives of the interface energy are easy to calculate from (4)

$$\frac{\partial f_{int}}{\partial \eta_i} - \boldsymbol{\nabla}\frac{\partial f_{int}}{\partial \boldsymbol{\nabla}\eta_i} = m\left(\eta_i^3 - \eta_i + 2\beta\eta_i\sum_{j\neq i}\eta_j^2\right) - \boldsymbol{\nabla}\cdot\alpha\boldsymbol{\nabla}\eta_i. \tag{13}$$

The derivative of the chemical energy is

$$\frac{\partial f_{ch}}{\partial \eta_i} = \sum_j \left(\frac{\partial h_j}{\partial \eta_i} f_j + h_j \frac{\partial f_j}{\partial \eta_i}\right), \tag{14}$$

{{eq:dfchdeta}}

where

$$\begin{aligned} h_j\frac{\partial f_j}{\partial \eta_i} &= h_j\frac{\partial f_j}{\partial \bar{V}^m}\frac{\partial \bar{V}^m}{\partial h_j}\frac{\partial h_j}{\partial \eta_i} + h_j\underbrace{\frac{\partial f_j}{\partial G_j}\frac{\partial G_j}{\partial x_j}}_{\mu_j}\frac{\partial x_j}{\partial h_j}\frac{\partial h_j}{\partial \eta_i} \qquad , \\ &= -h_j\frac{f_j}{\bar{V}^m}\frac{\partial \bar{V}^m}{\partial h_j}\frac{\partial h_j}{\partial \eta_i} + \mu_j h_j\frac{\partial x_j}{\partial h_j}\frac{\partial h_j}{\partial \eta_i} \end{aligned} \tag{15}$$

{{eq:dfjdetai}}

where it was used that

$$\frac{\partial f_j}{\partial \bar{V}^m} = -\frac{G_j}{\bar{V}^{m^2}} = -\frac{f_j}{\bar{V}^m}. \tag{16}$$

In the case where the molar volume is assumed constant the first term of (15) vanishes. For the second term we can use that $\frac{\partial c}{\partial \eta_i} = 0$, equation (6) then gives

$$\begin{aligned} \frac{\partial c}{\partial \eta_i} = \sum_j \left(\frac{\partial h_j}{\partial \eta_i} x_j + h_j \frac{\partial x_j}{\partial h_j} \frac{\partial h_j}{\partial \eta_i} \right) = \sum_j \frac{\partial h_j}{\partial \eta_i} \left(x_j + h_j \frac{\partial x_j}{\partial h_j} \right) = 0 \\ \iff \sum_j h_j \frac{\partial x_j}{\partial h_j} \frac{\partial h_j}{\partial \eta_i} = - \sum_j \frac{\partial h_j}{\partial \eta_i} x_j. \end{aligned} \tag{17}$$

By making use of the KKS condition (14) can then be written

$$\frac{\partial f_{ch}}{\partial \eta_i} = \sum_j \frac{\partial h_j}{\partial \eta_i} \left(f_j - h_j \frac{f_j}{\bar{V}^m} \frac{\partial \bar{V}^m}{\partial h_j} \frac{\partial h_j}{\partial \eta_i} - \mu_j x_j \right) \tag{18}$$

Equation (12) now becomes

$$\begin{aligned} \frac{\partial \eta_i}{\partial t} = -L \Bigg(\sum_j \frac{\partial h_j}{\partial \eta_i} \left(f_j - h_j \frac{f_j}{\bar{V}^m} \frac{\partial \bar{V}^m}{\partial h_j} \frac{\partial h_j}{\partial \eta_i} - \mu_j x_j \right) \\ + m \left(\eta_i^3 - \eta_i + 2\beta \eta_i \sum_{j \neq i} \eta_j^2 \right) - \boldsymbol{\nabla} \cdot \alpha \boldsymbol{\nabla} \eta_i \Bigg). \end{aligned} \tag{19}$$

2.3 Weak form residual equations

The weak form of the residual equation for η_i is constructed from (12) by multiplication with a test function ψ and integrating over the domain:

$$\begin{aligned} \boldsymbol{R}_{\eta_i} &= \left(\frac{\partial \eta_i}{\partial t}, \psi \right) + \left(L \left(\frac{\partial f_{ch}}{\partial \eta_i} + \frac{\partial f_{int}}{\partial \eta_i} + \frac{\partial f_d}{\partial \eta_i} \right), \psi \right) - (L \boldsymbol{\nabla} \cdot \alpha \boldsymbol{\nabla} \eta_i, \psi) \\ &= \left(\frac{\partial \eta_i}{\partial t}, \psi \right) + \left(L \left(\frac{\partial f_{ch}}{\partial \eta_i} + \frac{\partial f_{int}}{\partial \eta_i} + \frac{\partial f_d}{\partial \eta_i} \right), \psi \right) + (\alpha \boldsymbol{\nabla} \eta_i, \boldsymbol{\nabla} L \psi) \\ &- \langle (\alpha \boldsymbol{\nabla} \eta_i) \cdot \boldsymbol{n}, L\psi \rangle, \end{aligned} \tag{20}$$

with the divergence theorem used on the last term. Inserting the derivatives calculated above gives

$$\begin{aligned} \boldsymbol{R}_{\eta_i} = \left(\frac{\partial \eta_i}{\partial t}, \psi \right) + \left(L \sum_j \frac{\partial h_j}{\partial \eta_i} f_j, \psi \right) - \left(L \frac{\partial h_j}{\partial \eta_i} \mu_j x_j, \psi \right) \\ - \left(L \sum_j \frac{\partial h_j}{\partial \eta_i} h_j \frac{f_j}{\bar{V}^m} \frac{\partial \bar{V}^m}{\partial h_j} \frac{\partial h_j}{\partial \eta_i}, \psi \right) + \left(L \frac{\partial f_d}{\partial \eta_i}, \psi \right) \\ + \left(Lm \left(\eta_i^3 - \eta_i + 2\beta \eta_i \sum_{j \neq i} \eta_j^2 \right), \psi \right) + (\alpha \boldsymbol{\nabla} \eta_i, \boldsymbol{\nabla} L \psi) \\ - \langle (\alpha \boldsymbol{\nabla} \eta_i) \cdot \boldsymbol{n}, L\psi \rangle \end{aligned} \tag{21}$$

The Cahn-Hilliard equation (11) can be solved in two ways. Either you can solve it directly using the residual

$$\boldsymbol{R}_c = \left(\frac{\partial c}{\partial t}, \psi\right) - \left(\boldsymbol{\nabla} \cdot M \boldsymbol{\nabla} \mu, \psi\right) = \\ \left(\frac{\partial c}{\partial t}, \psi\right) + \left(M \boldsymbol{\nabla} \mu, \boldsymbol{\nabla} \psi\right) - \left\langle M \boldsymbol{\nabla} \mu \cdot \boldsymbol{n}, \psi \right\rangle . \quad (22)$$

This seems to be fairly straight forward, but for some reason it is “not fully implemented” in Moose. Instead, the equation is split into two and solved for the concentration c as well as for the chemical potential μ. The residuals for this are

$$\boldsymbol{R}_\mu = \left(\frac{\partial c}{\partial t}, \psi\right) + \left(M \boldsymbol{\nabla} \mu, \boldsymbol{\nabla} \psi\right) - \left\langle M \boldsymbol{\nabla} \mu \cdot \boldsymbol{n}, \psi \right\rangle \quad (23)$$

$$\boldsymbol{R}_c = \left(\frac{\partial F}{\partial c} - \mu, \psi\right) . \quad (24)$$

3 Implementation

This section describes the input file used to run the simulation. Moose input files consists of different blocks which together defines the model. All necessary blocks are described below.

3.1 Mesh

The mesh block can be used to create meshes of simple geometries. It can also be used to read the mesh from a file. A rectangular 2D mesh is created by like this:

```
[Mesh]
  type = GeneratedMesh
  dim = 2
  nx = 100
  ny = 10
  xmax = 0.304 # Length
  ymax = 0.0257 # Height
[]
```

3.2 Variables

The `Variables` block declares the variables of the model. A variables is defined like this:

```
[Variables]
  # concentration Sn
  [./c] # Variable name
      order = FIRST
      family = LAGRANGE
  [../]
[]
```

To solve the model the following variables are needed: c,μ,x_i,η_i, i.e. a simulation with three grains of different phases contains 8 variables.

3.3 ICs and BCs

Initial conditions can be set in the `ICs` block. For simple initial conditions the type `FunctionIC` can be used. The following code will initiate the variable η_{imc} as a ring with radius 8 centered at $(20, 20)$.

```
[ICs]
    [./eta2] #Cu6Sn5
        variable = eta_imc
        type = FunctionIC
      function = 'r:=sqrt((x-20)^2+(y-20)^2);if(r>8&r<=16,1,0)'
    [../]
[]
```

There are some more complicated initial conditions already implemented but probably I'll have to make something myself.

Boundary conditions are set in the `BCs` block using a very similar syntax as the ICs.

3.4 Kernels

In Moose each part of the residual equations are implemented in separate kernels. Each kernel is responsible for evaluating the residual and (optionally) calculate the Jacobian. For each of the Allen-Cahn equations the kernels in Table 1 is needed. For the Cahn-Hilliard equation and the KKS conditions the kernels in Table 2 are needed.

[2] This kernel includes a double well that we don't use. Therefore $w = 0$.

Table 1: Kernels needed for the Allen-Cahn equation {tab:AC}

Residual term	Parameters	Kernel name
$\left(\frac{\partial\eta_i}{\partial t},\psi\right)$		`TimeDerivative`
$\left(L\sum_j \frac{\partial h_j}{\partial \eta_i} f_j + w\frac{\partial g}{\partial\eta_i},\psi\right)$	L, $w=0$	`KKSMultiACBulkF`[2]
$-\left(L\frac{\partial h_j}{\partial\eta_i}\mu_j x_j,\psi\right)$	L	`KKSMultiACBulkC`
$-\left(L\sum_j \frac{\partial h_j}{\partial\eta_i} h_j \frac{f_j}{\bar{V}^m}\frac{\partial \bar{V}^m}{\partial h_j}\frac{\partial h_j}{\partial \eta_i},\psi\right)$		
$\left(L\frac{\partial f_d}{\partial\eta_i},\psi\right)$		
$\left(Lm\left(\eta_i^3-\eta_i+2\beta\eta_i\sum_{j\neq i}\eta_j^2\right),\psi\right)$	m, β	`ACGrGrMulti`
$(\alpha\boldsymbol{\nabla}\eta_i,\boldsymbol{\nabla}L\psi)$	L, α	`ACInterface`

Table 2: Kernels needed for the split version of the Cahn-Hilliard equation and the KKS conditions {tab:CH}

Residual term	Parameters	Kernel name
$\left(\frac{\partial c}{\partial t},\psi\right)$		`CoupledTimeDerivative`
$(M\boldsymbol{\nabla}\mu,\boldsymbol{\nabla}\psi)$	M	`SplitCHWRes`
$\left(\frac{\partial F}{\partial c}-\mu,\psi\right)$		`KKSSplitCHCRes`
$\mu=\mu_i=\mu_j$		`KKSPhaseChemicalPotential`
$c=\sum_i h_i x_i$		`KKSMultiPhaseConcentration`

3.5 Materials

The free energies and the switching functions are defined in the `Materials` block of the input file. The chemical free energy of each phase can be described using the type `DerivativeParsedMaterial`. This type uses automatic differentiation to calculate the derivatives.

```
[Materials]
  [./fch_cu] #Chemical energy Cu phase
        type = DerivativeParsedMaterial
        f_name = fch_cu
        args = 'c_cu'
        function = '20*(c_cu-0.1)^2'
  [../]
[]
```

The switching functions are set like

```
  [./h_cu]
        type = SwitchingFunctionMultiPhaseMaterial
        h_name = h_cu
        all_etas = 'eta_cu eta_imc eta_sn'
        phase_etas = eta_cu
  [../]
```

The same switching function can be used to describe all grain of the same phase (I think).

The materials block can also be used to define constants. For example

```
  [./constants]
      type = GenericConstantMaterial
      prop_names  = 'L kappa gamma mu tgrad_corr_mult'
      prop_values = '1. 0.5 0.5 1. 0.'
  [../]
```

3.6 Executioner, Preconditioning and Outputs

The parameters of the solver are set in the `Executioner, Preconditioning and Outputs` blocks, here you can also set options for the behavior of the underlying PETSc commands. The blocks can look something like this

```
[Executioner]
  type = Transient
  solve_type = 'PJFNK'
 petsc_options_iname = '-pc_type -sub_pc_type  -sub_pc_factor_shift_type'
  petsc_options_value = 'asm       ilu            nonzero'
  l_max_its = 30
  nl_max_its = 10
  l_tol = 1.0e-4
  nl_rel_tol = 1.0e-10
  nl_abs_tol = 1.0e-11

  num_steps = 100
  dt = 0.5
[]

[Preconditioning]
  active = 'full'
  [./full]
    type = SMP
    full = true
  [../]
  [./mydebug]
    type = FDP
    full = true
  [../]
[]

[Outputs]
  exodus = true
[]
```

These settings will solve the equations using the Pre-Conditioned Jacobian-Free Newton-Krylow method which is the standard way in Moose. The results will be written to a file that can be viewed in e.g. Peacock or Paraview.

References

[1] Johan Hektor, Matti Ristinmaa, Håkan Hallberg, Stephen A Hall, and Srinivasan Iyengar. Coupled diffusion-deformation multiphase field model for elastoplastic materials applied to the growth of cu 6 sn 5. *Acta Materialia*, 108:98–109, 2016.