## 1. Problem description

For 1D solid-liquid problem, specie M transport between two domains (left domain 1 and right domain 2).

1) Diffusion in both domains can be described by Fick's law, Specie M has different diffusivities in different materials.
2) Dirichlet boundary conditions are applied at most left and most right.
3) At the interface consider the following
    a) Fluxes are matched from both domains
    b) The reaction is

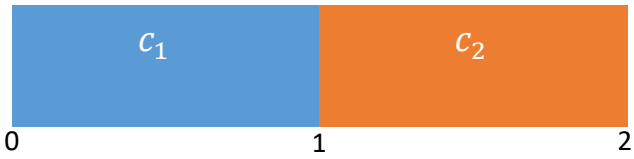
$$M(1) \underset{k_b}{\overset{k_f}{\rightleftharpoons}} M(2)$$

With the first order reaction rate assuming a quasi-steady-state

$$ReactionRate = \frac{dC_1}{dt} = k_f C_1 - k_b C_2 \approx 0$$

## 2. Model in 1-D

### 2.1. Governing equation




In domain 1 ($0 \leq x_1 < 1$)

$$D_1 \frac{\partial^2 C_1}{\partial {x_1}^2} = 0 \quad (1)$$

In domain 2 ($1 < x_2 \leq 2$)

$$D_2 \frac{\partial^2 C_2}{\partial {x_2}^2} = 0 \quad (2)$$

Where $D_1$ =4, $D_2$=2.

### 2.2. Boundary condition

$$C_1|_{x=0} = 1 \quad (3)$$

$$C_2|_{x=2} = 0 \quad (4)$$

### 2.3. At the interface

$$J_1 = J_2$$

or

$$-D_1 \frac{dC_1}{dx_1}\Big|_{x=1} = -D_2 \frac{dC_2}{dx_2}\Big|_{x=1} \tag{5}$$

Also due to the reactions, it should satisfy

$$J_{interface} = R = k_f C_1 - k_b C_2$$

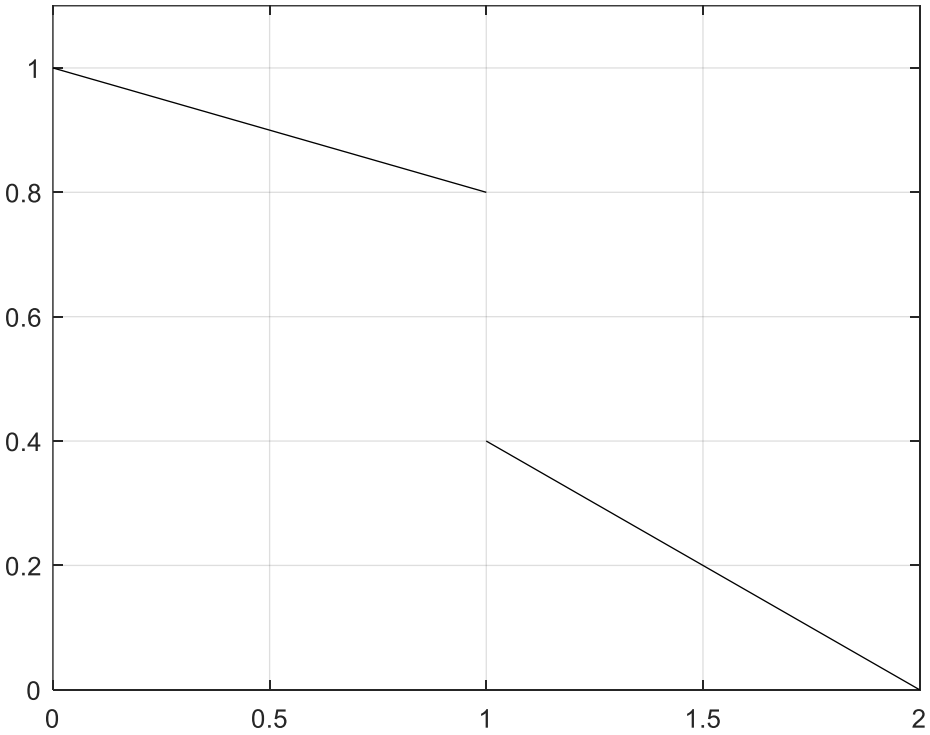
Or

$$k_f C_1 - k_b C_2 = 0 \tag{6}$$

Where $k_f = 1, k_b = 2$.

**2.4. Analytic solution:**

$$\begin{cases} C_1 = -0.2x_1 + 1 & 0 \le x_1 < 1 \\ C_2 = -0.4x_2 + 0.8 & 1 < x_2 \le 2 \end{cases} \tag{7}$$




**3. MOOSE way**

**3.1. Overview**

MOOSE works on "kernel" which represent a piece of physics (one or more operators or terms in a PDE, like $\nabla \cdot D\nabla u$)

| Equation | Kernel | Remark |
|---|---|---|
| (1) | MatDiffusion | MOOSE provide |
| (2) | MatDiffusion | MOOSE provide |
| (3) | DirichletBC | MOOSE provide |

| | | |
|---|---|---|
| (4) | DirichletBC | MOOSE provide |
| (5) | InterfaceDiffusion | MOOSE provide |
| (6) | InterfaceReaction | User defined |

As we can notice, for this simple question MOOSE can provide most of the kernels except for InterfaceReaction Kernel ( Eq.(6)).

### 3.2. Details for InterfaceReaction Kernel

Similar to InterfaceDiffusion kernel [1], in order to write a kernel to operate the variable ($u_1$ for master, $u_2$ for neighbor) for InterfaceReaction, we need to get the Residual [2] (`computeQpResidual`) and Jacobian [2][3] (`computeQpJacobian`) for master domain and neighbor domain respectively.

Residual (`computeQpResidual`)

| | |
|---|---|
| Residual for element (master domain) | $\psi_1 \cdot (k_f u_1 - k_b u_2)$ |
| Residual for neighbor | $-\psi_2 \cdot (k_f u_1 - k_b u_2)$ |

Where $\psi_1$ is the test function of master domain, $\psi_2$ is the test function of neighbor domain. The negative sign in element (master domain), think of a NeumannBC.

Jacobian (`computeQpJacobian`)

| Kernel code | | Understanding and explanation | |
|---|---|---|---|
| computeQpJacobian() case Moose:: | Jac= | Domain | Get derivative w.r.t. |
| ElementElement | $\psi_1 \cdot k_f \phi_1$ | master | $u_1$ |
| NeighborNeighbor | $-\psi_2 \cdot (-k_b \phi_2)$ | neighbor | $u_2$ |
| NeighborElement | $-\psi_2 \cdot (k_f \phi_1)$ | neighbor | $u_1$ |
| ElementNeighbor | $\psi_1 \cdot (-k_b \phi_2)$ | master | $u_2$ |

Where $\phi_1$ is the shape function of master domain, $\phi_2$ is the shape function of neighbor domain.

### 3.3. Result and discussion

The code [4] has been implemented to 1-D geometry giving the results agree with the analytic solution (ElementL2Error is at the order of E-17).

Jacobian is analyzed by the method provided by MOOSE [5], the result shows the No errors detected. :-).

[1] http://mooseframework.org/wiki/MooseSystems/InterfaceKernels/
[2] http://mooseframework.org/wiki/MooseTraining/FEM/NumericalImplementation/
[3] http://mooseframework.org/moose/application_development/jacobian_definition.html
[4] https://github.com/wesleyzzz/InterfaceReaction1.git

[5] http://mooseframework.org/wiki/JacobianDebugging/