

1 General information

This software is a script compilation for FreeFem++ to solve interface cell problems and arrive with coefficients in boundary condition (BC) between porous medium and free fluid. This module gives the possibility to compute the coefficients for BC in two-dimensions (2D) for porous case. If you find this software useful, please cite the corresponding publication <arXiv:1604.02880> (arXiv preprint).

The contents of this module are listed below.

src/2D_MSE_bc_Ksolver.edp	- script to solve for K tensor
src/2D_MSE_bc_Lsolver.edp	- script to solve for L tensor
src/2D_MSE_solver.edp	- script to solve a Cavity problem with developed boundary conditions
doc/2D_MSE_bc_velocity.pdf	- this document

In this document we explain the key elements of the software, parameters of the simulation, and implemented variational formulation.

2 Solver for interface tensors K and L

2.1 Structure of the simulation scripts

Each simulation script take following input:

1. geometry of the interface cell (in-line cylinders);
2. interface location (*intfShft*);
3. extent of the cell (*ymin, ymax*);
4. solid volume fraction (*thetas*);
5. mesh;
6. problem indices (*indk* for permeability and *indk, indl* for receptivity);

The geometry and mesh (resolution controlled by parameters n and $nIntf$) currently are generated within the software, but could be generated using some other tool and read in. The geometry generation of interface begins with code shown below.

```

90 // Define geometry, top
91 border sqr1t(t=intfShft,ymax) {x= 0.5; y=t; label=1; }
92 border sqr2t(t=0.5,-0.5) {x=t; y=ymax; label=2; }
93 border sqr3t(t=ymax,intfShft) {x=-0.5; y=t; label=3; }
94 border sqr4t(t=-0.5,0.5) {x=t; y=intfShft; label=4; }

```

Assertion is carried out to ensure that interface is not slicing the upper cylinder (line 59). Geometry defining routines must be modified in order to allow interface locations slicing the solid structure. Additionally, the classical permeability problem of the interior is defined and solved in lines 65–86 for getting appropriate boundary conditions for the interface cell (see main paper).

The result of the simulation script is tensor fields at the full interface cell. The output is organized in text files, one for plane-average at the interface (resulting coefficients for velocity boundary condition), one for plane-average for sample coordinates over the whole domain (for understanding the details of flow near the interface).

2.2 Variational formulation

The developed software solve coupled systems of Stoke's equations. By fixing indices k, l , we set $u_i^{K\pm} = K_{ik}^\pm$, $u_i^{L\pm} = L_{ikl}^\pm$, $p^{K\pm} = A_k^\pm$ and $p^{L\pm} = B_{kl}^\pm$. Then the governing equations take form (see paper <arXiv:1604.02880> for all details)

$$-p_{,i}^{K+} + u_{i,jj}^{K+} = 0, \quad -p_{,i}^{L+} + u_{i,jj}^{L+} = 0, \quad (1)$$

$$u_{i,i}^{K+} = 0, \quad u_{i,i}^{L+} = 0, \quad (2)$$

$$u_i^{K+} \Big|_{\Gamma_C} = 0, \quad u_i^{L+} \Big|_{\Gamma_C} = 0, \quad (3)$$

$$u_i^{K-} \Big|_{\Gamma_I} = u_i^{K+} \Big|_{\Gamma_I}, \quad u_i^{L-} \Big|_{\Gamma_I} = u_i^{L+} \Big|_{\Gamma_I}, \quad (4)$$

$$[-p^{K+} \delta_{ij} + 2 \varepsilon_{ij} (u^{K+})] \Big|_{\Gamma_I} n_j = [-p^{L+} \delta_{ij} + 2 \varepsilon_{ij} (u^{L+})] \Big|_{\Gamma_I} n_j = \quad (5)$$

$$= [-p^{K-} \delta_{ij} + 2 \varepsilon_{ij} (u^{K-})] \Big|_{\Gamma_I} n_j, \quad = [-p^{L-} \delta_{ij} + 2 \varepsilon_{ij} (u^{L-})] \Big|_{\Gamma_I} n_j - \delta_{ik} n_l, \quad (6)$$

$$-p_{,i}^{K-} + u_{i,jj}^{K-} = -\delta_{ik}, \quad -p_{,i}^{L-} + u_{i,jj}^{L-} = 0, \quad (7)$$

$$u_{i,i}^{K-} = 0, \quad u_{i,i}^{L-} = 0, \quad (8)$$

$$u_i^{K-} \Big|_{\Gamma_C} = 0, \quad u_i^{L-} \Big|_{\Gamma_C} = 0. \quad (9)$$

Structure of both set of equation systems is exactly the same, the only difference lies in volume forcing and stress boundary condition. Without losing generality, we proceed by showing how to solve problems for K . In order to simplify notation, we skip the K in equations. In order to solve these problems, we define test functions \hat{u}_i^\pm , \hat{p}^\pm and $\hat{\lambda}$ for velocity, pressure and Lagrange multiplier (defined only on the common interface) fields. To arrive with weak formulation, we multiply equations with test function and employ integration by parts for fluid stresses, dis-

nated by $\Sigma_{ij}^\pm = -p^\pm \delta_{ij} + 2\varepsilon_{ij}(u^\pm)$, to obtain

$$-\int_{\Omega} \Sigma_{ij}^+ \hat{u}_{i,j}^+ d\Omega + \int_{\partial\Omega} \Sigma_{ij}^+ n_j \hat{u}_i^+ dS = 0, \quad (10)$$

$$\int_{\Omega} u_{i,i}^+ \hat{p}^+ d\Omega = 0, \quad (11)$$

$$-\int_{\Omega} \Sigma_{ij}^- \hat{u}_{i,j}^- d\Omega + \int_{\partial\Omega} \Sigma_{ij}^- n_j \hat{u}_i^- dS = -\int_{\Omega} \delta_{ik} \hat{u}_i^- d\Omega, \quad (12)$$

$$\int_{\Omega} u_{i,i}^- \hat{p}^- d\Omega = 0, \quad (13)$$

for governing equations, and

$$\int_{\Gamma_I} u_i^+ \hat{\lambda} dS - \int_{\Gamma_I} u_i^- \hat{\lambda} dS = 0, \quad (14)$$

$$\int_{\Gamma_I} \Sigma_{ij}^+ n_j \hat{\lambda} dS - \int_{\Gamma_I} \Sigma_{ij}^- n_j \hat{\lambda} dS = 0, \quad (15)$$

for boundary conditions. A scalar field has been selected for Lagrange multiplier space in order to have the ability to operate with boundary conditions on component basis, i.e., put separate boundary conditions for different velocity or stress components. Additionally, we must modify the weak form of the governing equations and add the forcing from Lagrange multipliers to force the required conditions. The governing equations then become

$$-\int_{\Omega} \Sigma_{ij}^+ \hat{u}_{i,j}^+ d\Omega + \int_{\partial\Omega} \Sigma_{ij}^+ n_j \hat{u}_i^+ dS + \int_{\Gamma_I} m_i^+(\lambda) \hat{u}_i^+ dS = 0, \quad (16)$$

$$\int_{\Omega} u_{i,i}^+ \hat{p}^+ d\Omega + \int_{\Gamma_I} d^+(\lambda) \hat{p}^+ dS = 0, \quad (17)$$

$$-\int_{\Omega} \Sigma_{ij}^- \hat{u}_{i,j}^- d\Omega + \int_{\partial\Omega} \Sigma_{ij}^- n_j \hat{u}_i^- dS + \int_{\Gamma_I} m_i^-(\lambda) \hat{u}_i^- dS = -\int_{\Omega} \delta_{ik} \hat{u}_i^- d\Omega, \quad (18)$$

$$\int_{\Omega} u_{i,i}^- \hat{p}^- d\Omega + \int_{\Gamma_I} d^-(\lambda) \hat{p}^- dS = 0, \quad (19)$$

where functions m_i^+ , m_i^- , d^+ and d^- are corresponding transpose functions of boundary conditions on the interface. This weak formulation results in a linear matrix to solve, as shown below.

```

161 // Assemble the main matrix, add transpose matrices for feedback
162 // from Lagrange multipliers
163 CoupSYS = [ [ L1, 0, B1', B2', B3', B4', B5', B6' ],
164             [ 0, L2, B1n', B2n', B3n', B4n', 0, 0 ],
165             [ B1, B1n, BPT, 0, 0, 0, 0, 0 ],
166             [ B2, B2n, 0, BPT, 0, 0, 0, 0 ],
167             [ B3, B3n, 0, 0, BPT, 0, 0, 0 ],
168             [ B4, B4n, 0, 0, 0, BPT, 0, 0 ],
169             [ B5, 0, 0, 0, 0, 0, BPB, 0 ],
170             [ B6, 0, 0, 0, 0, 0, 0, BPB ] ];

```

The matrices $L1$ and $L2$ are bilinear forms of fluid stress in upper and lower domains, $B1$, $B2$, $B1n$ and $B2n$ are bilinear forms of velocity boundary condition two components, $B3$, $B4$, $B3n$ and $B4n$ are bilinear forms of stress boundary condition two components at the interface, and $B5$ and $B6$ are bilinear forms of stress boundary condition two components at the top of the upper domain. The prime denotes a transpose of matrices and those are added in the system as feedback from Lagrange multipliers. Due to the fact, that FreeFem++ doesn't have the capability to work with surface-only meshes, we have to add penalty for unused parts of Lagrange multiplier space, which are BPT and BPB . In addition, we have through testing noticed that Lagrangian multiplier approach works the best using double the resolution and one order smaller finite element space (in our case $P1$).

3 Solver for led driven cavity flow over porous medium

The simulation script take following input:

1. geometry of the flow problem, both free fluid and porous medium, number of grid points n per unit length;
2. scale separation parameter $epsP$ and volume fraction $thetas$;
3. interface location (yif);
4. results from interface cell (tensors K and L);
5. Upper wall velocity $Udrv$.

The domain consists of free fluid domain Ω_f and porous medium domain Ω_p , encompassed by boundaries at interface Γ_{if} , upper wall Γ_{f2} and side walls Γ_{f13} of free fluid, and lower wall Γ_{p4} and side walls Γ_{p13} of porous domain. Within these two domains, we solve equation system

$$-p_{,i} + \epsilon u_{i,jj} = 0 \quad \text{in } \Omega_f, \quad u_i|_{\Gamma_{f13}} = 0, \quad u_1|_{\Gamma_{f2}} = Udrv, \quad u_2|_{\Gamma_{f2}} = 0, \quad (20)$$

$$u_{i,i} = 0 \quad \text{in } \Omega_f, \quad u_1|_{\Gamma_{if}} = -K_{1j} \frac{\tilde{p}_{,j}}{\epsilon} + L_{1ij} (u_{i,j} + u_{j,i}), \quad (21)$$

$$u_2|_{\Gamma_{if}} = -K_{2j} \frac{\tilde{p}_{,j}}{\epsilon} + L_{2ij} (u_{i,j} + u_{j,i}), \quad (22)$$

$$\tilde{p}_{,jj} = 0 \quad \text{in } \Omega_p, \quad \tilde{p}|_{\Gamma_{if}} = p, \quad \tilde{p}_{,1}|_{\Gamma_{p13}} = 0, \quad \tilde{p}_{,2}|_{\Gamma_{p4}} = 0. \quad (23)$$

The Darcy flow in the interior is expressed using the Darcy law with constant isotropic permeability K_{drc}

$$\tilde{u}_i = -K_{drc} \frac{\tilde{p}_{,i}}{\epsilon}. \quad (24)$$

For the final remark, we have observed that construction of Lagrange multipliers is different. Before we saw that transpose operation was always sufficient (and required) for accurate solution. However, in this case, one has to explicitly define that forcing from velocity boundary condition acts only on velocities (and not on derivatives of velocity or pore pressure), while pressure

boundary condition for pore pressure equation acts only as forcing in the pore pressure equation. Otherwise, the weak form and solver is constructed exactly as described in the previous section.

4 Issue: fluid stress tensor in weak form

The authors think that it is their responsibility to warn potential users of this software that the FreeFem++ might be in some cases sensitive to exact form in which the weak form of the fluid stress tensor is written. It seems that it could be a bug in the software.

To illustrate the issue, we consider fluid stress tensor written in two forms, as shown below.

```
// Fluid viscous and pressure part
varf lapT([u,v,p],[uu,vv,pp]) = int2d(ThT) ( - 2.0*dx(u)*dx(uu) - (dy(u)+dx(v))*dy(uu)
- 2.0*dy(v)*dy(vv) - (dx(v)+dy(u))*dx(vv)
+ p*dx(uu)
+ p*dy(vv)
+ (dx(u)+dy(v))*pp );

// Fluid viscous and pressure part
varf lapT([u,v,p],[uu,vv,pp]) = int2d(ThT) ( - (-p + 2.0*dx(u))*dx(uu) - (dy(u)+dx(v))*dy(uu)
- (-p + 2.0*dy(v))*dy(vv) - (dx(v)+dy(u))*dx(vv)
+ (dx(u)+dy(v))*pp );
```

Running the solver for \mathcal{L}_{112} coefficient using parameters $\phi = 0.02$ and $y_s = 0.1$ yields results of

$$\mathcal{L}_{112}^{(1)} = 0.166932, \quad (25)$$

$$\mathcal{L}_{112}^{(2)} = 0.178310, \quad (26)$$

for the first and second form, respectively. We have observed that this difference decreases, as interface coordinate y_s is increased, or the interface is moved up. We have also observed that only L solver is affected by this bug. Results from the K solver are identical using both stress forms. This behaviour has been confirmed to exist in FreeFem++ versions 3.46 and 3.37. In order to determine, which version is correct, we compare results with literature, paper by Carraro, Thomas, et al. “Pressure jump interface law for the Stokes–Darcy coupling: confirmation by direct numerical simulations.” J. Fluid Mech. 732 (2013): 510-536. Based on results shown in Tab. 1, we have concluded that second stress form yields accurate results.

Literature	$-C_{1,h}^{bl} = 0.303821942379 \pm 2 \cdot 10^{-12}$
Stress form 1	$\mathcal{L}_{112}^{(1)} = 0.297088$
Stress form 2	$\mathcal{L}_{112}^{(2)} = 0.303857$

Table 1: Comparison of interface cell results to literature.