PORE-FLOW© User's Manual

(version 1.2)

Hua Tan, Reza Masoodi, and Krishna M Pillai

January 2011

Laboratory for Flow and Transport Studies in Porous Media

University of Wisconsin-Milwaukee

TABLE OF CONTENTS

1. Introduction

PORE-FLOW© is a comprehensive computational fluid dynamics tool that solves flow infiltration/wetting problems encountered in industrial porous media. The finite element/control volume method is implemented in the code to simulate flow behind a moving-boundary. The algorithm is efficient and robust for solving the moving-boundary problems in complex domain geometries. The geometry may be 2D or 3D, and the mesh may be structured or unstructured to give maximum flexibility to the user. The porous-medium flow in the code is governed by either Darcy's law or Brinkman equation, depending on the user's choice. PORE-FLOW© also can solve the fluid flow problems governed by Stokes or Navier-Stokes equations. The heat flow as well as certain types of reactive flows can also be simulated by the code.

1.1 Features and Benefits

- Easy implementation – Can be used with current software (ANSYS preprocessing and Tecplot post-processing)
- Extensive validation - Modeling tools have been extensively validated using controlled mold filling experiments
- Less modeling error - 30% better agreement compared to current alternatives
- Cheaper - Minimizes cost through optimization of mold design, lower design costs, lower prototyping costs, and lessens need for reworking of molds
- More accurate – Better prediction of pressure and temperature in molds
- Better prediction - Flux-corrected transport for filling simulation removes localized wiggle often seen in solutions and better predictions for permeability
- More versatile - More precise estimation with different mat types such as woven/stitched mats by first solving at the microscale

1. Overview of PORE-FLOW$^©$ data file

2.1 General remarks

The main purpose of this document is to explain how to create the command file for PORE-FLOW$^©$. The command file consists of several command blocks. A template for all of blocks is presented in the corresponding section. The command file has the following properties:

- Lines starting with the symbol "$" are skipped
- Symbols "_", "=", ":", and "," are skipped
- Symbol "!" or "$" can be used to give a comment in a command line
- Only the first EIGHT alphabetic characters of commands are read
- It is not case sensitive
- Blanks are not considered
- The commands written between brackets[] are optional

2.2 Structure of the command file

The command file consists of several groups (blocks) of commands. All of them have a starting word that identifies the group and must finish with the same word following the word END_. There are also several sub-groups of commands that must be provided following the same syntax. The order of the commands within the groups is irrelevant.

The complete list of blocks and sub-blocks of command file is listed as following:

```
$************************************************************
$  PORE-FLOW: Title
$************************************************************
PHYSICAL_PROBLEM
      NAVIER_STOKES_EQUATIONS
      BRINKMAN_EQUATIONS
      DARCY_EQUATION
      HEAT_EQUATION
      SPECIES_EQUATION
      FREE_SURFACE
END_PHYSICAL_PROBLEM

PROPERTIES
     NUM_OF_SETS
     SET
        DENSITY
        VISCOSITY
        POROSITY
        PERMEABILITY
        CONDUCTIVITY
        SPECIFIC_HEAT
        HEAT_GENERATION
        REACTION_RATE
        THICKNESS
     ENDSET
END_PROPERTIES

MESH_DATA
     DIMENSIONS
     GEOMETRY
          ELEMENTS
          COORDINATES
END_MESH_DATA

BOUNDARY_CONDITIONS
     INITIAL_CONDITIONS
     DIRICHLET_CONDITIONS
     NEUMANN_BOUNDARY_CONDITIONS
END_BOUNDARY_CONDITIONS

NUMERICAL_TREATMENT
    TIME_DATA
    INTEGRATION_POINTS
    ITERATION
    ERROR
    OUT_OF_CORE
    NAVIER_STOKES_BRINKMAN
END_NUMERICAL_TREATMENT

OUTPUT
    FILE_NAME
    FREQUENCY
    MUMPS_INFORMATION
END_OUTPUT
```

3. Physical problem

This block of data can have six sub-blocks depends on the problem needed to be solved. The structure of these blocks and their sub-blocks are:

```
 PHYSICAL_PROBLEM
    NAVIER_STOKES_EQUATIONS
    BRINKMAN_EQUATIONS
    DARCY_EQUATION
    HEAT_EQUATION
    SPECIES_EQUATION
    FREE_SURFACE
END_PHYSICAL_PROBLEM
```

3.1 Navier-Stokes equations

All the parameters for NS equations are defined in the following block

NAVIER_STOKES_EQUATIONS: ON / OFF, [DOMAIN_NUM=*n*]
  PRESSURE_ELIMINATION: ON, PENALTY=ε / OFF
  TRANSIENT: ON / OFF
  CONVECTION: ON / OFF
  STABLIZATION: SUPG / OFF
  BODY_FORCES: ON, GX: g_x, GY: g_y, GZ: g_z / OFF
END_NAVIER_STOKES_EQUATIONS

The header of the block '**NAVIER_STOKES_EQUATIONS**' has two options

ON: The NS equations are solved

OFF: The NS equations are not solved

DOMAIN_NUM=*n* is an optional parameter. When NS equations combined with Brinkman equations are solved simultaneously for porous-clear fluid problem, **DOMAIN_NUM=***n* must be provided. *n* is an integer assigned to the finite elements which lie in the clear-fluid domain.

PRESSURE_ELIMINATION: This command determines if penalty method is used in solution of NS equations or not.

ON: Penalty method[1] is used and **PENALTY=**ε must be provide. ε must be a very small number, e.g. 1e-8.

OFF: Penalty method is not used.

[1] In penalty method, the pressure DOFs do not appear in the final discrete algebraic equations. The continuity equation is incorporated into the momentum equations through a penalty term. Details can be found in FIDAP theory manual.

TRANSIENT: This command determines if the transient effect is considered or not.

ON: Transient term in NS equations is taken into account

OFF: No transient term considered

CONVECTION: This command determines if the inertia term is considered or not.

ON: Inertia term in NS equations is taken into account

OFF: No Inertia term considered. It's in fact Stokes equation

STABLIZATION: This command determines stabilized technique. When the convection is strong, element Reynolds number may be too large to cause solution oscillation. The stabilized technique can stabilize the solution without refining the mesh. When CONVECTION is turned 'ON', it is suggested that the stabilized technique be used.

SUPG: Streamline-Upwinding Petrov Galerkin method

OFF: No stabilized technique is used

BODY_FORCES: the command defines body force

ON: body force is considered. **GX**: *x* component of ***g***, **GY**: *y* component of ***g***, **GZ**: *z* component of ***g***.

OFF: no body force is considered.

3.2 Brinkman equations

All the parameters for Brinkman equations are defined in the following block

```
 BRINKMAN_EQUATIONS:  ON / OFF, [DOMAIN_NUM=n]
    PRESSURE_ELIMINATION:  ON, PENALTY=ε / OFF
    BODY_FORCES: ON, GX: g_x, GY: g_y, GZ: g_z  / OFF
END_BRINKMAN_EQUATION
```

The header of the block '**BRINKMAN_EQUATIONS**' has two options

ON: The Brinkman equations are solved

OFF: The Brinkman equations are not solved

DOMAIN_NUM=*n* is an optional parameter. When Brinkman equations combined with NS equations are solved simultaneously for porous-clear fluid problem, **DOMAIN_NUM=***n* must be provided. *n* is an integer assigned to the finite elements which lie in the porous domain.

PRESSURE_ELIMINATION: This command determines if penalty method is used in solution of NS equations or not.

ON: Penalty method is used and **PENALTY**=ε must be provide. ε must be a very small number, e.g. 1e-8.

OFF: Penalty method is not used.

BODY_FORCES: the command defines body force

ON: body force is considered. **GX**: x component of $\boldsymbol{g}$, **GY**: y component of $\boldsymbol{g}$, **GZ**: z component of $\boldsymbol{g}$.

OFF: no body force is considered.

3.3 Darcy equation

The block of commands describing the Darcy equation to be solved is the following:

```
DARCY_EQUATION:ON / OFF
    BODY_FORCES: ON, GX: gx, GY: gy, GZ: gz  / OFF
END_DARCY_EQUATION
```

The header of the block '**DARCY_EQUATION**' has two options

ON: The Darcy equation is solved

OFF: The Darcy equation is not solved

BODY_FORCES: the command defines body force

ON: body force is considered. **GX**: x component of $\boldsymbol{g}$, **GY**: y component of $\boldsymbol{g}$, GZ: z component of $\boldsymbol{g}$.

OFF: no body force is considered.

3.4 Heat equation

The block of commands describing the heat equation to be solved is the following:

```
HEAT_EQUATION: ON / OFF
     TRANSIENT: ON / OFF
      CONVECTION: ON / OFF
      SHELL_MODEL: ON, WALL_TEMPERATURE=Twall / OFF
      STABLIZATION: SUPG / FCT / OFF
END_HEAT_EQUATION
```

The header of the block '**HEAT_EQUATION**' has two options

ON: The heat equation is solved

OFF: The heat equation is not solved

TRANSIENT: This command determines if the transient effect is considered or not.

ON: Transient term is taken into account

OFF: No transient term considered

CONVECTION: This command determines if the convective heat transfer is considered or not.

ON: heat convection is taken into account

OFF: No convective term considered. It's in fact pure heat conduction equation

SHELL_MODEL: determines if the thin-wall model used in 3D heat transfer problem.

ON: thin-wall model is considered. In this case, only heat conduction in thickness direction is considered along with heat transfer in-plane. **WALL_TEMPERATURE** represent wall temperature T_{wall}.

OFF: no thin-wall model is used

STABLIZATION: This command determines stabilized techniques. When the heat convection is strong, element Peclet number may be too large to cause solution oscillation (so called convection-dominated convection-diffusion equation). The stabilized techniques can stabilize the solution without refining the mesh. When CONVECTION is turned 'ON', it is suggested that the stabilized technique be used.

SUPG: Streamline-Upwinding Petrov Galerkin method

FCT: Flux-Corrected Transport method

OFF: No stabilized technique is used

3.5 Species equation

The block of commands describing the species equation to be solved is the following:

```
SPECIES_EQUATION: ON / OFF
END_SPECIES_EQUATION
```

The header of the block '**SPECIES_EQUATION**' has two options

ON: The heat equation is solved

OFF: The heat equation is not solved

3.6 Free surface flow problem

PORE-FLOW© uses control volume method to track the flow front. The block of commands describing the free surface-flow problem to be solved is the following:

```
FREE_SURFACE: ON / OFF
   WICKING: ON, CAPILLARY_PRESSURE=pa  /  OFF
   DUAL_SCALE: ON /OFF
END_FREE_SURFACE
```

The header of the block '**FREE_SURFACE**' has two options

ON: The free surface is considered

OFF: The free surface is not considered

WICKING: This command determines the wicking flow.

ON: the wicking flow is considered. **CAPILLARY_PRESSURE** must be provided; p_a should be a negative number.

OFF: it's not wicking flow

DUAL_SCALE: The command determines if dual-scale flow is considered

ON: dual-scale flow

OFF: single-scale flow

SINK_MODEL: This command determines the algorithm for solving flow through the dual scale fibrous media. (When **DUAL_SCALE** is turned **ON**, This card must be provided)

MULTISCALE: the algorithm needs two meshes, one is global mesh, and the other one is unit cell mesh.

LUMPED: the algorithm needs only one mesh. The parameters for sink function *must be* defined under '**PHYSICAL PROPERTIES**'.

4. Physical properties

This block of commands provides the physical properties used in simulation.

```
PROPERTIES
        NUM_OF_SETS=n
        SET n
             DENSITY
             VISCOSITY
             POROSITY
             PERMEABILITY
             CONDUCTIVITY
             SPECIFIC_HEAT
             HEAT_GENERATION
             REACTION_RATE
             THICKNESS
             SINK_FUNCTION
          ENDSET n
 END_PROPERTIES
```

NUM_OF_SETS: defines the total number of property sets. For example, if there are two different permeabilities in a porous medium, then *n*=2.

SET: determines n^{th} sets of properties

DENSITY: defines the density ρ . If density is a constant, then a float type number ρ is followed by **DENSITY**, e.g. DENSITY 1000.0. Density can be defined as a function of variables, for example

DENSITY VARIABLE=TIME, FILLTIME, TEMP, CURE, X, Y, Z

FUN: expression of the function

VISCOSITY: defines the viscosity μ. Similar to Density, viscosity can be defined as either a constant or a function.

POROSITY: defines the porosity ϕ. Similar to Density, porosity can be defined as either a constant or a function.

PERMEABILITY: defines the permeability tensor. If permeability tensor is a constant, **KX**=k_x x-component of permeability tensor, **KY**=k_y y-component of permeability tensor, **KZ**=k_z z-component of permeability tensor, **ROTATION**=θ, rotation angle of principle direction to x, y, z coordinates. If permeability is homogenous, it can be defined as a function similar to density.

CONDUCTIVITY: defines the thermal conductivity tensor. If conductivity tensor is a constant, **CX=**k_x x-component of conductivity tensor, **CY=**k_y y-component of conductivity tensor, **CZ=**k_z z-component of conductivity tensor, **ROTATION=**θ, rotation angle of principle direction to x, y, z coordinates.

SPECIFIC_HEAT: defines the specific heat c_p. Similar to Density, specific heat can be defined as either a constant or a function.

HEAT_GENERATION: defines the heat generation. Similar to Density, heat generation can be defined as either a constant or a function.

REACTION_RATE: defines the reaction rate. Similar to Density, reaction rate can be defined as either a constant or a function.

THICKNESS: defines the thickness of 2D mesh

SINK_FUNCTION: This determines parameters for sink function

$$\frac{dS_{\text{tow}}}{dt} = \frac{A_1 P_{gap}}{a\mu}\left\{ e^{\left[A_2\left(1-S_{\text{tow}}\right)^{A_3}\right]} - 1 \right\}$$

A1=A_1, **A2=** A_2, **A3=** A_3, **B=**a

5. Mesh data

This section contains the definition of the computational domain where the problem has to be solved and its discretization.

```
MESH_DATA
       DIMENSIONS
       GEOMETRY
            ELEMENTS
            COORDINATES
END_MESH_DATA
```

5.1 Dimensions

This sub-block contains the following commands

```
DIMENSIONS
     NUMBER_NODES:     ng
     NUMBER_ELEMENTS:   ne
     SPACE:   2D /3D
     ELEMENT_TYPE: TRIA 3 / QUAD 4 / HEXA 8 /TETR 4
     AXISYMMETRY:   YES / NO
END_DIMENSIONS
```

NUMBER_NODES: total number of nodes in FE mesh.

NUMBER_ELEMENTS: total number of elements in FE mesh.

SPACE: defines space dimensions.

ELEMENT_TYPE: defines the element type. ELEMENT_TYPE has the following value depending on the mesh type.

Value	Meaning
TRIA 3	Triangular element with 3 nodes
QUAD 4	Quadrilateral element with 4 nodes
HEXA 8	Hexahedral element with 8 nodes
TETR 4	Tetrahedral element with 4 nodes

AXISYMMETRY: YES-> axi-symmetrical problem, NO. This option is necessary for 2D problems. Note: for axi-symmetrical problem, *y* direction is always viewed as axis of symmetry.

5.2 Geometry

This block of commands defines the FE mesh.

```
GEOMETRY
        ELEMENTS
            ie,  IDdomain, IDset, node1, …, node n
        END_ELEMENTS
        COORDINATES
            in,  LOCX,  LOCY,  LOCZ
        END_COORDINATES
END_GEOMETRY
```

ELEMENTS: defines the connectivity of each element.

ie: element number

IDdomain: domain number corresponding to DOMAIN_NUM=*n* in Section 3.1 and 3.2

IDset: property set number corresponding to SET *n* in Section 4

node*1*, …, node *n*: list of nodes belonging to element *ie*

COORDINATES: defines the coordinates of nodes.

in: node number

LOCX: *x*-coordinate of node *in*

LOCY: *y*-coordinate of node *in*

LOCZ: *z*-coordinate of node *in*

NOTE: To make the command file compact and neat, the sub-blocks of ELEMENTS and COORDINATES can be included in another file. The path and file name need to be provided in GEOMETRY, e.g.,

```
GEOMETRY
   INCLUDE C:\*****\***\XXX.XX
END_GEOMETRY
```

6. Boundary conditions

This block contains the definition of the boundary conditions for the problem to be solved.

```
BOUNDARY_CONDITIONS
    INITIAL_CONDITIONS
    DIRICHLET_CONDITIONS
    NEUMANN_CONDITIONS
    FREE_SURFACE_CONDITIONS
END_BOUNDARY_CONDITIONS
```

6.1 Initial conditions

The block defines the initial conditions for transient analysis.

INITIAL_CONDITIONS
    VX=V_x , VY=V_y, VZ=V_z, TEMPERATURE=T_0, CURE=α_0
END_INITIAL_CONDITIONS

VX: initial velocity in *x*-direction V_x

VY: initial velocity in *y*-direction V_y

VZ: initial velocity in *z*-direction V_z. For 2D problems, do not include this term.

TEMPERATURE: initial temperature. If HEAT_EQUATION is turned '**OFF**', do not include this term.

CURE: initial degree of cure. If SPECIES_EQUATION is turned '**OFF**', do not include this term.

NOTE: for pure heat transfer problems, the INITIAL_CONDITIONS can be used to give the velocity field.

6.2 Dirichlet condition

The block defines the Dirichlet boundary conditions. Note: for NS equations, Brinkman equations, Dirichlet condition specifies the velocity on the boundary, while for Darcy equation, Dirichlet condition specifies the pressure value on the boundary.

DIRICHLET_CONDITIONS
    in, IDvx, IDvy, IDvz, IDT, IDcure, V_x, V_y, V_z, T, α (NS or Brinkman)
    in, IDP, IDT, IDcure, P_a, T, α (Darcy equation)
END_DIRICHLET_CONDITIONS

in: node number.

IDvx: code for boundary condition of *x*-velocity. IDvx is either 0 or 1. 0 means *x*-velocity is free. 1 means the *x*-velocity is prescribed.

IDvy: code for boundary condition of *y*-velocity. The possibilities of IDvy are same as IDvx.

IDvz: code for boundary condition of *z*-velocity. The possibilities of IDvz are same as IDvx. For 2D problem, do not include this term.

IDP: code for boundary condition of pressure for Darcy equation. The possibilities of IDP are same as IDvx.

IDT: code for boundary condition of temperature. The possibilities of IDT are same as IDvx. If HEAT_EQUATION is turned '**OFF**', do not include this term.

IDcure: code for boundary condition of cure. The possibilities of IDcure are same as IDvx. If SPECIES_EQUATION is turned '**OFF**', do not include this term.

V_x: *x*-velocity of node *in*.

V_y: *y*-velocity of node *in*.

V_z: *z*-velocity of node *in*. For 2D problem, do not include this term.

P_a: pressure of node *in*.

T: temperature of node *in*. If HEAT_EQUATION is turned '**OFF**', do not include this term.

α : degree of cure of node *in*. If SPECIES_EQUATION is turned '**OFF**', do not include this term.

6.3 Neumann conditions

The block defines the Neumann boundary conditions. Note: for NS equations, Brinkman equations, Neumann condition specifies the pressure on the boundary, while for Darcy equation, Neumann condition specifies flow rate on the boundary.

NEUMANN_CONDITIONS
in, IDP, IDFlux, P_a, *Flux* (for NS or Brinkman)
in, IDF, IDFlux, *Q*, *Flux* (for Darcy equation)
END_NEUMANN_CONDITIONS

in: node number.

IDP: code for boundary condition of pressure. IDP is either 0 or 1. 0 means pressure is free. 1 means that pressure is prescribed.

IDF: code for boundary condition of flow rate. IDF is either 0 or 1. 0 means pressure is free. 1 means that flow rate is prescribed.

***P*_a**: pressure applied on node *in*.

Q: flow rate applied on node *in*.

Flux: heat flow applied on node *in*. If HEAT_EQUATION is turned '**OFF**', do not include this term.

6.4 Free Surface Conditions

This block defines the inlet and outlet boundaries. The block is necessary only when FREE_SURFACE is turned "**ON**".

```
FREE_SURFACE_CONDITIONS
   in, ID
END_FREE_SURFACE_CONDITIONS
```

in: node number.

ID: code for inlet or outlet. ID is either 0 or 1. 0 means node *in* belongs to inlet. 1 means node *in* belongs to outlet.

NOTE: To make the command file compact and neat, the sub-blocks of DIRICHLET_CONDITIONS, NEUMANN_CONDITIONS, and FREE_SURFACE_CONDITIONS can be included in a file. The path and file name need to be provided in BOUNDARY_CONDITIONS, e.g.

```
BOUNDARY_CONDITIONS
     INITIAL_CONDITIONS
          VX=0, VY=0, VZ=0
     END_INITIAL_CONDITIONS
     INCLUDE C:\*****\***\XXX.XX
END_BOUNDARY_CONDITIONS
```

7. Numerical Treatment

The block determines the parameters in numerical solution.

```
NUMERICAL_TREATMENT
    TIME_DATA
    INTEGRATION_POINTS
    ITERATION
    ERROR
    OUT_OF_CORE
    NAVIER_STOKES_BRINKMAN
        BUBBLE_FORMULATION
        PRESSURE_INTERPOLATION
    END_NAVIER_STOKES
END_NUMERICAL_TREATMENT
```

TIME_DATA: determines the time integration algorithm for transient analysis, which has following keywords:

INITIAL_TIME: initial time

FINAL_TIME: end of time

TIME_STEP_SIZE: time increment in each step

THETA: an adjustable parameter varying between 0 and 1. THETA can be 0 (forward), 0.5 (Crank-Nicolson), 1 (backward).

Note: all the keywords including TIME_DATA should be in the same line, for example,

TIME_DATA: INITIAL_TIME=0.0, FINAL_TIME=100, TIME_STEP_SIZE=1.e-1, THETA=0.5

INTEGRATION_POINTS: the number of integral points for each element. It is an integer number which is determined by the element type of Section 5.1.

Element type	# Integration points
TRIA 3	4
QUAD 4	4 or 9
HEXA 8	8 or 27
TETR 4	1, 4, 5, 10, or 11

ITERATION: iteration method. It can be following value

PICARD: fixed point iteration method

NEWTON: Newton iteration method

PICNEW *n*: the combination of Picard and Newton method. Here *n* means the number of iteration using Picard method. In this method, the first *n* steps use Picard method followed by Newton method. When PICNEW is used, *n* must be provided.

ERROR: criterion for iteration method. When the difference between the solutions of successive iterations is less than the number, the convergence is reached. For example

$$\frac{\left\| \mathbf{u}_{i+1} - \mathbf{u}_i \right\|}{\mathbf{u}_i} \le \varepsilon$$

where **u** is solution, subscript *i* is iteration number, *ε* is error of iteration.

OUT_OF_CORE: '**ON**' or '**OFF**'. When the problem is very large, the in-core memory requirement may exceed the capacity of the computer. Turning on OUT_OF_CORE can use hard drive to store the matrix.

7.1 Numerical treatment for NS and Brinkman equations

This block determines the parameters just for NS and Brinkman equation. If NAVIER_STOKES_EQUATIONS or BRINKMAN_EQUATIONS is turned '**ON**', the block must be included. Otherwise, it is not necessary to include the whole block.

```
NAVIER_STOKES_BRINKMAN
        BUBBLE_FORMULATION:  ON / OFF
        PRESSURE_INTERPOLATION: n
END_NAVIER_STOKES
```

BUBBLE_FORMULATION: when tetrahedral elements with 4 nodes are used to solve NS and Brinkman equation, BUBBLE_FORMULATION should be turned '**ON**'. For other types of elements, it should be turn '**OFF**'.

PRESSURE_INTERPOLATION: This defines the interpolation used for the pressure. An integer must be provided. Now it can be only 1.

8. Output

PORE-FLOW$^{©}$ uses TECPLOT as post processor.

```
OUTPUT
    FILE_NAME
    FREQUENCY: STEP= n
    MUMPS_INFORMATION: ON /OFF
END_OUTPUT
```

FILE_NAME: file name for output file.

FREQUENCY: for transient analysis or filling analysis, this parameter determines how often the solution is written into the solution file.

MUMPS_INFORMATION: PORE-FLOW© uses MUMPS to solve the algebraic equation. When MUMPS_INFORMATION is turn ‘**ON**’, the solution information coming from MUMPS will be printed on the screen. Otherwise, no solution information is printed.

9. Preprocessing

In order to run this software, one needs three following files:

1. Command file
2. Mesh file
3. Boundary condition file

The command file was discussed in the previous sections. We discuss how to make the mesh and boundary condition files in this section:

9.1 Mesh file

Mesh file has two parts; the first part is in the following format:

```
ELEMENTS
element number, Domain ID, Property ID, local node 1, ...., local node 8
END_ELEMENTS
```

The first three columns are the element #, domain ID and property ID. The next columns are nodes associated with the element in the first column. Therefore, we have 4 nodes for 2D and 8 nodes for 3D simulation. The second part of mesh file is coordinate, which gives the coordinate information for nodes. Here is the format:

```
COORDINATES
 node number, x coordinate, y coordinate, z coordinate
END_COORDINATES
```

In case of working in 2D, then one of these columns is zero or there are just two columns. In case, there is one column, the other column is considered to be zero. If the mesh was made in ANSYS, then above data can be achieved through following command:

Preprocessor>>Archive Model>>Write>>Data to Archive>>pick GEOM; then in “Archive file” give a name for output file.

We may need to delete some rows or columns from GEOM file. For this purpose, UltraEdit is a powerful editing software. Following is a sample 2D mesh file:

```
ELEMENTS
$ element number, Domain ID, Property ID, local node 1, ...., local node 4
   1  1  1   1    3    81    80
   2  1  1   3    4   100    81
   3  1  1   4    5   119   100
   4  1  1   5    6   138   119
   5  1  1   6    7   157   138
   6  1  1   7    8   176   157
   7  1  1   8    9   195   176
   8  1  1   9   10   214   195
   9  1  1  10   11   233   214
  10  1  1  11   12   252   233
  .
  .
  .
 393  1  1  308  327   49   50
 394  1  1  327  346   48   49
 395  1  1  346  365   47   48
 396  1  1  365  384   46   47
 397  1  1  384  403   45   46
 398  1  1  403  422   44   45
 399  1  1  422  441   43   44
 400  1  1  441   41   22   43
END_ELEMENTS

COORDINATES
$ node number, x coordinate, y coordinate, z coordinate
   1  0.00000000
   2  0.177800000
   3  8.890000000E-03
   4  1.778000000E-02
   5  2.667000000E-02
   6  3.556000000E-02
   7  4.445000000E-02
   8  5.334000000E-02
   9  6.223000000E-02
  10  7.112000000E-02
   .
   .
   .
   435  0.168910000   0.115570000
   436  0.168910000   0.124460000
```

```
437   0.168910000    0.133350000
438   0.168910000    0.142240000
439   0.168910000    0.151130000
440   0.168910000    0.160020000
441   0.168910000    0.168910000
 END_COORDINATES
```

9.2 Boundary condition file

Boundary condition has two or three of following parts:

DIRICHLET_CONDITIONS
in, IDvx, IDvy, IDvz, IDT, IDcure, V_x, V_y, V_z, T, α (NS or Brinkman)
in, IDP, IDT, IDcure, P_a, T, α (Darcy equation)
END_DIRICHLET_CONDITIONS

Or

NUMANN_CONDITIONS
in, IDP, IDFlux, P_a, *Flux* (for NS or Brinkman)
in, IDF, IDFlux, Q, *Flux* (for Darcy equation)
END_NEUMANN

Or

FREE_SURFACE_CONDITIONS
in, ID
END_FREE_SURFACE_CONDITIONS

The details of above characters are explained in sections 6.2, 6.3, and 6.4. If the mesh was made in ANSYS, then above described node numbers can be obtained through following command:

1) Selecting the boundary of the object. In a 2D mesh, one can use following commands to select a boundary line:

   Select>>Entities>>Lines>>By Num/Pick>>click on boundary line(s)>>OK

2) Select the nodes connected to previous selected boundary:

   Select>>Entities>>Nodes>>Attached to>>Lines, all>>OK

3) Output the node information:

   List>>Nodes>>OK

Following is a sample boundary condition file for mold filling simulation under constant pressure injection:

```
DIRICHLET_CONDITIONS
  1 1 20000
 42 1 20000
 62 1 20000
 63 1 20000
 .
.
.
 77 1 20000
 78 1 20000
 79 1 20000
 80 1 20000
END_DIRICHLET_CONDITIONS

FREE_SURFACE_CONDITIONS
 1  0
42  0
62  0
63  0
.
.
.
37  1
38  1
39  1
40  1
41  1
END_FREE_SURFACE_CONDITIONS
```

10. Numerical examples

10.1 Wicking Flow through swelling medium

The problem is a wicking flow through a swelling medium involving a moving boundary. The gravity effect is taken into consideration. The permeability is changing with time as follows

$$P_c = -96271.21679 \text{ Pa (capillary pressure)}$$

$$K/\varepsilon_0 * 1E14 = 0.04508552357\, t - 1.673195381\, t^{1/2} + 23.62425661 \quad m^2$$

The viscosity and density of the liquid are μ=0.000911 Pa.s and 1000 kg/m^3. Porosity ε_0 is 0.5. In this case, 2D analysis is carried out. Quadrilateral elements are used to descritize the computational domain. Total numbers of nodes and elements are 1111 and 1000, respectively. The mesh model is shown in figure 1.

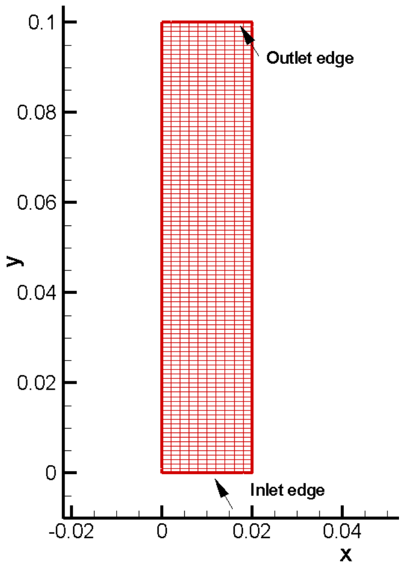


Figure 1 : FE mesh model.

The command file is listed here

```
$**************************************************************************
PORE-FLOW 2D swelling media flow problem
$**************************************************************************
PHYSICAL_PROBLEM
$-----------------------------------------------------------------------
 DARCY_EQUATION: ON
     BODY_FORCES: ON,GX:0., GY:-9.8, GZ:0.    ←— Consider gravity
 END_DARCY
$-----------------------------------------------------------------------
 FREE_SURFACE: ON   ←—— FREE_SURFACE must be turned on, since it's a moving boundary problem
   WICKING: ON, CAPILLARY_PRESSURE= -96271.21679
   DUAL_SCALE:OFF                          ↖  Wicking flow and capillary pressure
 END_FREE_SURFACE
$-----------------------------------------------------------------------
```

```
END_PHYSICAL_PROBLEM
$**************************************************************************
PROPERTIES
 NUM_OF_SETS=1        Since the material is homogenous, Only one set of property
                      is used in the simulation.
$-----------------------------------------------------------------------
 SET 1
     DENSITY: 1000
     VISCOSITY: .000911                                Permeability is function
     POROSITY: .5                                      of  filling time.
     PERMEABILITY:  VARIABLE=FILLTIME
          FUN: (0.04508552357*FILLTIME-1.673195381*FILLTIME^(.5)+23.62425661)*1e-14*0.5
     THICKNESS: 1.  ! valid for 2D filling analysis
 ENDSET 1                                                            Function expression
END_PROPERTIES      '!' can be used to give comment in a command line
$**************************************************************************
MESH_DATA
$-----------------------------------------------------------------------
   DIMENSIONS
       NUMBER_NODES:      1111
       NUMBER_ELEMENTS:   1000
       SPACE:   2D
       ELEMENT_TYPE: QUAD 4
       AXISYMMETRY:   NO
   END_DIMENSIONS
$-----------------------------------------------------------------------
   GEOMETRY
     INCLUDE C:\hua tan\code validation\example\swell.qu    Mesh data is included in swell.qu.
   END_GEOMETRY                                             When running the example, note
$-----------------------------------------------------------------------   the path of swell.qu.
END_MESH_DATA
$**************************************************************************
BOUNDARY_CONDITIONS
$-----------------------------------------------------------------------
    INITIAL_CONDITIONS
        VX=0. VY=0. VZ=0.     INITIAL_CONDITIONS does not work in this case.
    END_INITIAL                                                B.C. data is included in swell.ini.
    INCLUDE C:\hua tan\code validation\example\swell.ini       When running the example, note
$-----------------------------------------------------------------------   the path of swell.ini.
END_BOUNDARY_CONDITIONS
$**************************************************************************
NUMERICAL_TREATMENT
    TIME_DATA:  INITIAL_TIME=0.0 FINAL_TIME=5 TIME_STEP_SIZE=1. THETA=0.5
    INTEGRATION_POINTS: 4
    ITERATION: PICARD          TIME_DATA, ITERATION, and
    ERROR: 5.D-5               ERROR are not necessary in this case.
    OUT_OF_CORE: OFF
END_NUMERICAL_TREATMENT
$**************************************************************************
OUTPUT
    FILE_NAME: global.out
    FREQUENCY:  STEP=5       The solution is output every 5 steps
    MUMPS_INFORMATION:Off
END_OUTPUT
$**************************************************************************
```

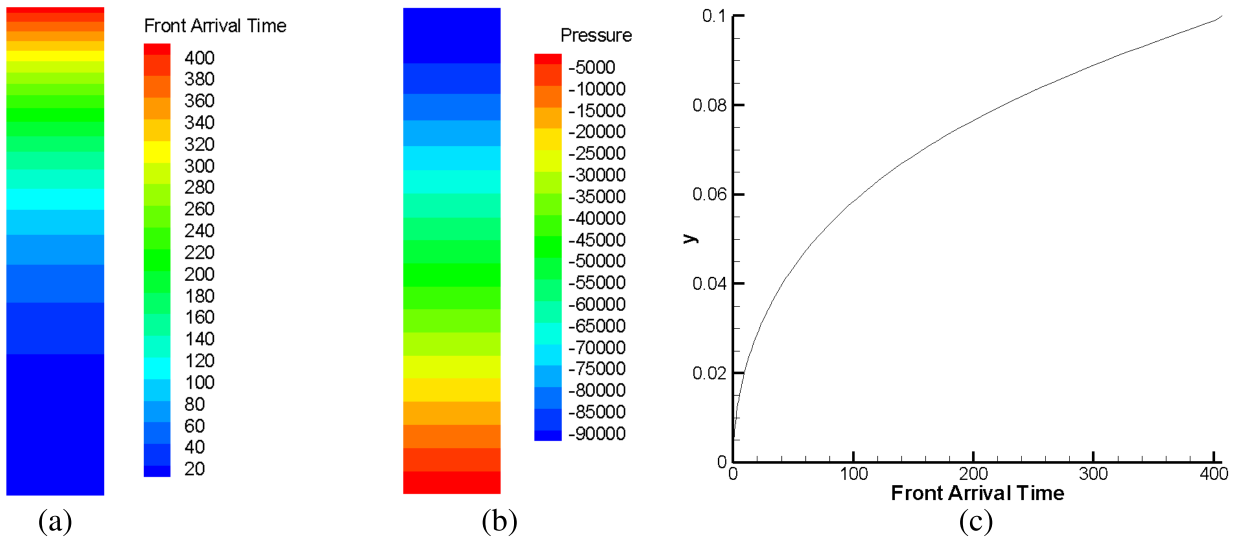


Figure 2. (a) the flow-front evolution, (b) the pressure distribution at the end of filling, (c) flow-front position vs time

10.2 Permeability prediction

A unit cell model of bi-axial fabrics is constructed to predict the permeability. The problem is steady-state. The longitudinal permeability $K_{\parallel}$ and transverse permeability $K_{\perp}$ of fiber tow are assumed to be 5e-10 m^2 and 1e-10 m^2, respectively. The liquid viscosity is 1 Pa s. The flow in inter-tow gap region is modeled using Stokes equations, while the flow in the intra-tow region is modeled using Brinkman equation. Therefore, there are two different computational domain (one for Stokes equation, the other one for Brinkman equation), and three sets of properties (one for gap region, one for tow parallel to z-direction, one for tow parallel to x-direction). The FE model of the unit cell is shown in Figure 3. It has 17572 nodes and 15320 hexahedral elements. The pressure boundary conditions of 100 Pa and 0 Pa are applied on the opposite surfaces of the unit cell along the flow direction to simulate the z-direction flow. Symmetric boundary conditions are imposed on the remaining surfaces of the unit cell.

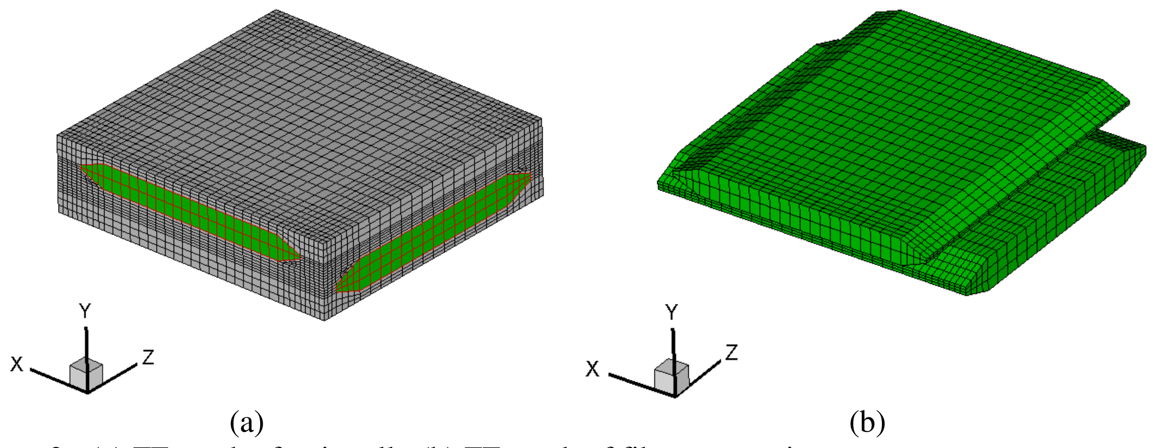


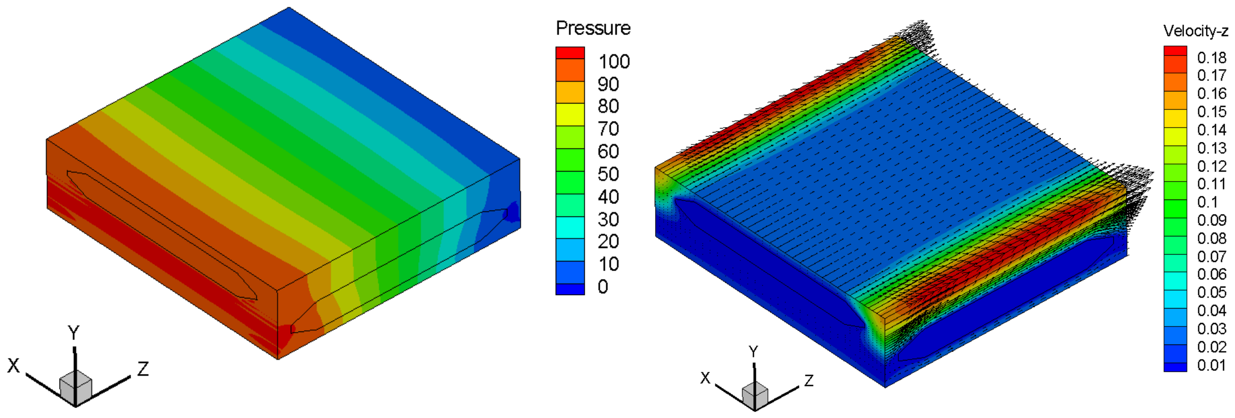
Figure 3 : (a) FE mesh of unit-cell, (b) FE mesh of fiber tow region.

The command file is listed as

```
$**************************************************************************
PORE-FLOW: PERMEABILITY PREDICTION PROBLEM
$**************************************************************************
PHYSICAL_PROBLEM
$-------------------------------------------------------------------------
 NAVIER_STOKES_EQUATIONS: ON, DOMAIN_NUM=1  ——> Domain 1 is Stokes equation
  PRESSURE_ELIMINATION: ON  PENALTY=1D-8
  TRANSIENT:  OFF
  CONVECTION: OFF  <—  Convection is turned off, so the inertial term is not considered.
  STABLIZATION: OFF  <—  Convection is turned off, hence there is no need to stabilize
  BODY_FORCES: OFF       oscillatory solution caused by the convection term.
 END_NAVIER_STOKES_EQUATIONS
```

```
$--------------------------------------------------------------------------
  BRINKMAN_EQUATIONS: ON, DOMAIN_NUM=2  ----> Domain 2 is Brinkman equation
  END_BRINKMAN_EQUATION
$--------------------------------------------------------------------------
END_PHYSICAL_PROBLEM
$**************************************************************************
PROPERTIES
 NUM_OF_SETS= 3  ----> There are 3 sets of properties in this example.
$--------------------------------------------------------------------------
 SET 1
   VISCOSITY: 1.      ----> Property set 1 for the inter-tow gap region.
 ENDSET 1
$--------------------------------------------------------------------------
 SET 2
   VISCOSITY: 1.          ----> Property set 2 for the tow parallel to z-direction
   POROSITY: .5
   PERMEABILITY: KX=1D-9, KY=1D-9, KZ=5D-9,ROTATION=0
 ENDSET 2
$--------------------------------------------------------------------------
 SET 3
   VISCOSITY: 1.          ----> Property set 3 for the tow parallel to x-direction
   POROSITY: .5
   PERMEABILITY: KX=5D-9, KY=1D-9, KZ=1D-9,ROTATION=0
 ENDSET 3
END_PROPERTIES
$**************************************************************************
MESH_DATA
$--------------------------------------------------------------------------
   DIMENSIONS
      NUMBER_NODES:      17572
      NUMBER_ELEMENTS:   15320
      SPACE:   3D
      ELEMENT_TYPE: HEXA 8
      AXISYMMETRY:   NO
   END_DIMENSIONS
$--------------------------------------------------------------------------
   GEOMETRY                                              Mesh data is included in UNITCELL.HEX.
     INCLUDE C:\hua tan\incompressiveflow1\UNITCELL.HEX  When running the example, note the path of
   END_GEOMETRY                                          UNITCELL.HEX.
$--------------------------------------------------------------------------
END_MESH_DATA
$**************************************************************************
BOUNDARY_CONDITIONS
$--------------------------------------------------------------------------
  INITIAL_CONDITIONS }                                   BC data is included in UNITCELL.INI. When
     VX=0 VY=0 VZ=0  }  It's not work in this case       running the example, note the path of
  END_INITIAL        }                                   UNITCELL.INI.
  INCLUDE C:\hua tan\incompressiveflow1\UNITCELL.INI
$--------------------------------------------------------------------------
END_BOUNDARY_CONDITIONS
$**************************************************************************
NUMERICAL_TREATMENT
    TIME_DATA:  INITIAL_TIME=0.0 FINAL_TIME=100. TIME_STEP_SIZE=1.d-1 THETA=0.5
    INTEGRATION_POINTS: 8  Eight integration points used
    ITERATION:  NEWTON
    ERROR: 5.D-5              <----  TIME_DATA, ITERATION, and
                              <----  ERROR are not necessary in this case.
    OUT_OF_CORE: OFF
$--------------------------------------------------------------------------
  NAVIER_STOKES_BRINKMAN
       BUBBLE_FORMULATION: OFF $/ON
       PRESSURE_INTERPOLATION: 1
  END_NAVIER_STOKES
$--------------------------------------------------------------------------
END_NUMERICAL_TREATMENT
$****************************************************************************
OUTPUT
     FILE_NAME: global.out
     FREQUENCY:  STEP=8
     MUMPS_INFORMATION: OFF
END_OUTPUT
$****************************************************************************
```

Since it is a steady-state problem, FREQUENCY does not work in this case.



(a) (b)

Figure 4 : (a) pressure distribution across the unit-cell, (b) z-velocity and velocity vector.

10.3 Flow in an elbow pipe

The flow through an elbow pipe is analyzed in this study. The problem is three dimensional Navier-Stokes flow. The liquid density and viscosity are 1000 kg/m^3 and 0.3 Pa s, respectively. The FE model is shown in figure 5. There are 2223 nodes and 1824 hexahedral elements in the model. The pressure boundary conditions of 1000 Pa and 0 Pa are applied on the inlet and outlet surfaces, respectively. The other surfaces are non-slip boundary condition.

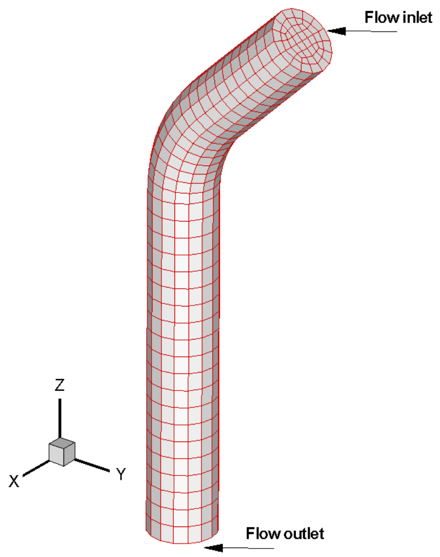
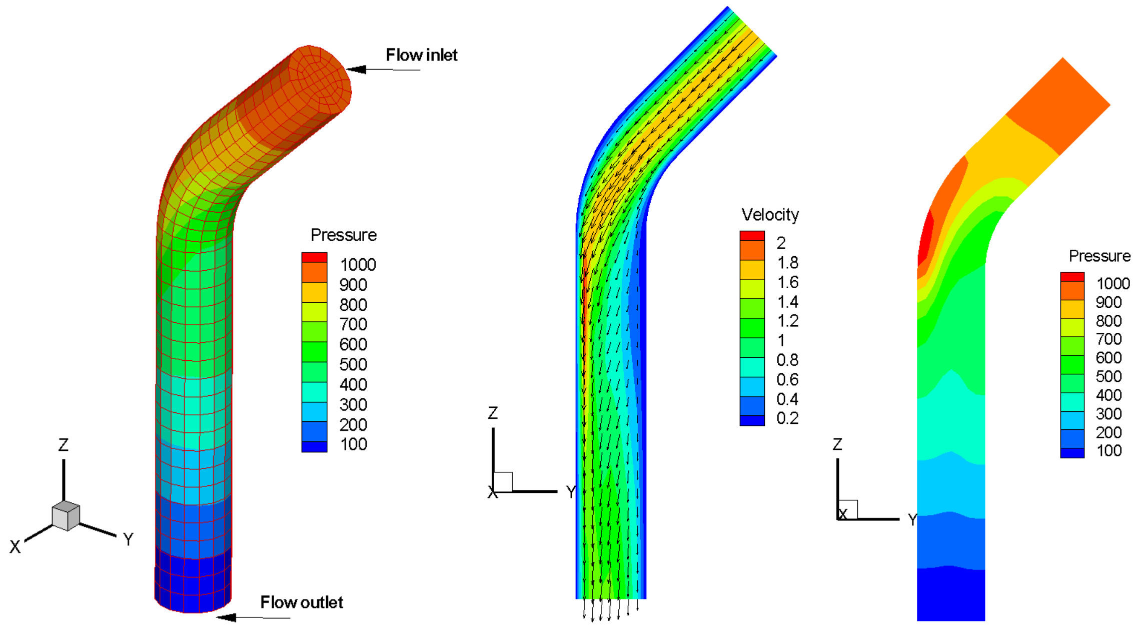


Figure 5: FE mesh of elbow pipe.

The command file is listed as

```
$**************************************************************************
PORE-FLOW: ELBOW 3D PROBLEM
$**************************************************************************
PHYSICAL_PROBLEM
$--------------------------------------------------------------------------
 NAVIER_STOKES_EQUATIONS: ON
    PRESSURE_ELIMINATION: on  PENALTY=1D-8
    TRANSIENT:  OFF
    CONVECTION: ON
    STABLIZATION: SUPG ←─── SUPG is used to stabilize the numerical solution
  BODY_FORCES: OFF,
 END_NAVIER_STOKES_EQUATIONS
$--------------------------------------------------------------------------
END_PHYSICAL_PROBLEM
$**************************************************************************
PROPERTIES
  NUM_OF_SETS=1
$--------------------------------------------------------------------------
  SET 1
    DENSITY:1000
    VISCOSITY: .3
    ENDSET 1
$---------------------------------------------------------------------------
END_PROPERTIES
$*****************************************************************************
MESH_DATA
$---------------------------------------------------------------------------
   DIMENSIONS
        NUMBER_NODES:      2223
        NUMBER_ELEMENTS:   1824
        SPACE:   3D
         ELEMENT_TYPE: HEXA 8
         AXISYMMETRY:   NO
   END_DIMENSIONS
$---------------------------------------------------------------------------
   GEOMETRY
       INCLUDE C:\hua tan\incompressiveflow1\ELBOW.INI
   END_GEOMETRY
$---------------------------------------------------------------------------
END_MESH_DATA
$*****************************************************************************
BOUNDARY_CONDITIONS
$---------------------------------------------------------------------------
  INITIAL_CONDITIONS
    VX=0 VY=0 VZ=0
  END_INITIAL
  INCLUDE C:\hua tan\incompressiveflow1\ELBOW.INI
$---------------------------------------------------------------------------
END_BOUNDARY_CONDITIONS
$*****************************************************************************
NUMERICAL_TREATMENT
    TIME_DATA:  INITIAL_TIME=0.0 FINAL_TIME=100. TIME_STEP_SIZE=1.d-1 THETA=0.5
    INTEGRATION_POINTS: 8
    ITERATION:  PICNEW
    ERROR: 5.D-5
$---------------------------------------------------------------------------
  NAVIER_STOKES_BRINKMAN
      BUBBLE_FORMULATION: OFF
      PRESSURE_INTERPOLATION: 1
  END_NAVIER_STOKES
$---------------------------------------------------------------------------
END_NUMERICAL_TREATMENT
$*****************************************************************************
OUTPUT
    FILE_NAME: global.out
    FREQUENCY:  STEP=8
END_OUTPUT
$*****************************************************************************
```



(a) (b) (c)

Figure 6 : (a) pressure distribution of the elbow, (b) velocity distribution and vector on the middle-plane of the elbow, (c) pressure distribution on the middle-plane.

10.4 Heat flow between two parallel plates

The problem is a cold liquid of 20^{0}C flowing between two infinitely large plates with a higher constant temperature of 75^{0}C. The liquid density, viscosity, conductivity and specific heat are 10 kg/m^3, 1 Pa s, 1 W/m ^{o}K, and 1000 J/kg ^{o}K, respectively. The inlet flow velocity is 1 m/s. Since the domain is symmetrical, only half of the domain is discretized using FE mesh as shown in figure 7. For this problem, the NS equations are first solved to obtain the velocity field, then the heat transfer equation is solved to obtain the temperature field.

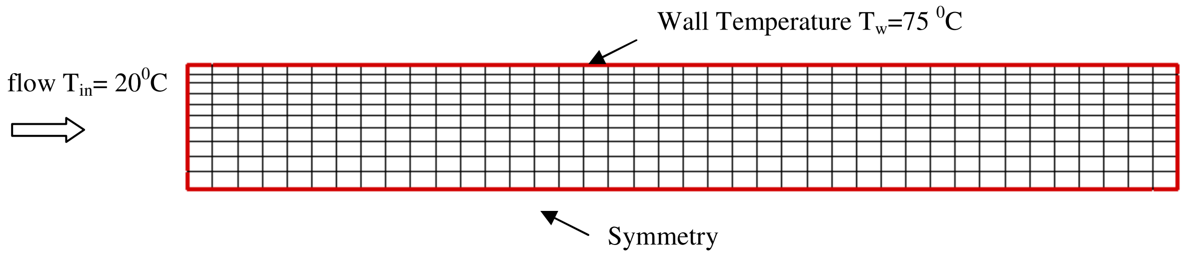


Figure 7: FE mesh

The command file is listed as

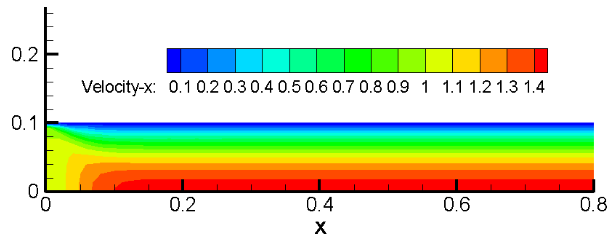
```
$**********************************************************************
$PORE-FLOW 2D heat transfer problem
$**********************************************************************
PHYSICAL_PROBLEM
$---------------------------------------------------------------------------
  NAVIER_STOKES_EQUATIONS: ON
        PRESSURE_ELIMINATION: ON,  PENALTY=1D-8
        TRANSIENT:  OFF
        CONVECTION: ON
        STABLIZATION: SUPG
        BODY_FORCES: OFF
  END_NAVIER_STOKES_EQUATIONS
```

Defines the NS equation

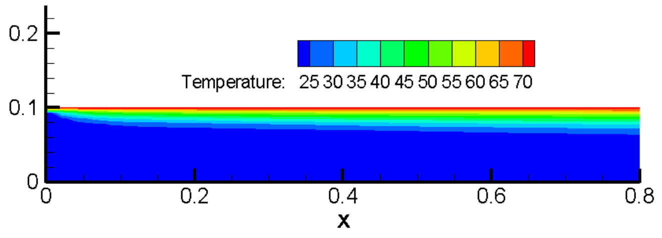
```
$---------------------------------------------------------------------------
  HEAT_EQUATION: ON
      TRANSIENT: OFF
      CONVECTION: ON
      SHELL_MODEL: OFF
      STABLIZATION: SUPG
  END_HEAT_EQUATION
```

Defines the heat transfer equation

```
$---------------------------------------------------------------------------
END_PHYSICAL_PROBLEM
$**********************************************************************
PROPERTIES
 NUM_OF_SETS=1
$---------------------------------------------------------------------------
 SET 1
   DENSITY: 10
   VISCOSITY: 1.
   CONDUCTIVITY: CX=1, CY=1, CZ=1,ROTATION=0
   SPECIFIC_HEAT:1000
   HEAT_GENERATION: 0.
   THICKNESS: 1d-2
 ENDSET 1
END_PROPERTIES
$*****************************************************************************
MESH_DATA
$-------------------------------------------------------------------------
   DIMENSIONS
      NUMBER_NODES:      451
      NUMBER_ELEMENTS:   400
      SPACE:   2D
      ELEMENT_TYPE: QUAD 4
      AXISYMMETRY:   NO
   END_DIMENSIONS
$-------------------------------------------------------------------------
   GEOMETRY
       INCLUDE C:\hua tan\code validation\example\plateflow2D.qu
   END_GEOMETRY
$-------------------------------------------------------------------------
END_MESH_DATA
$*****************************************************************************
BOUNDARY_CONDITIONS
$-------------------------------------------------------------------------
  INITIAL_CONDITIONS
      VX=0. VY=0. VZ=0. TEMPERATURE=293
  END_INITIAL
  INCLUDE C:\hua tan\code validation\example\plateflow2D.ini
$-------------------------------------------------------------------------
END_BOUNDARY_CONDITIONS
$*****************************************************************************
NUMERICAL_TREATMENT
    TIME_DATA:  INITIAL_TIME=0.0 FINAL_TIME=100. TIME_STEP_SIZE=1D-2 THETA=0.5
     INTEGRATION_POINTS: 4
     ITERATION: NEWTON
     ERROR: 6.D-5
     OUT_OF_CORE: OFF
$-------------------------------------------------------------------------
     NAVIER_STOKES_BRINKMAN
        BUBBLE_FORMULATION: OFF
        PRESSURE_INTERPOLATION: 1
      END_NAVIER_STOKES
$-------------------------------------------------------------------------
END_NUMERICAL_TREATMENT
$*****************************************************************************
OUTPUT
     FILE_NAME: global.out
     FREQUENCY:  STEP=5
     MUMPS_INFORMATION: OFF
END_OUTPUT
$*****************************************************************************
```



(a)



(b)

Figure 8: (a) velocity distribution, (b) temperature distribution.

10.5 Mold filling simulation

The problem is a mold of 7”×7” (0.1778×0.1778m) mold that is filling with a resin or a test liquid. The porosity is 0.5 and the density and viscosity of liquid are 860 kg/m^3 and 0.244 Pa.s, respectively. The permeability of fibermats is 1e-7 m^2 and the injection pressure is 20 kPa. We want to study the flow of resin in such a simple 2-d mold. The domain is discretized using FE mesh as shown in Figure 9. For this problem, the Darcy equation and continuity equations are solved to find the pressure distribution. Then the liquid front location and velocity will be found using Darcy’s law.

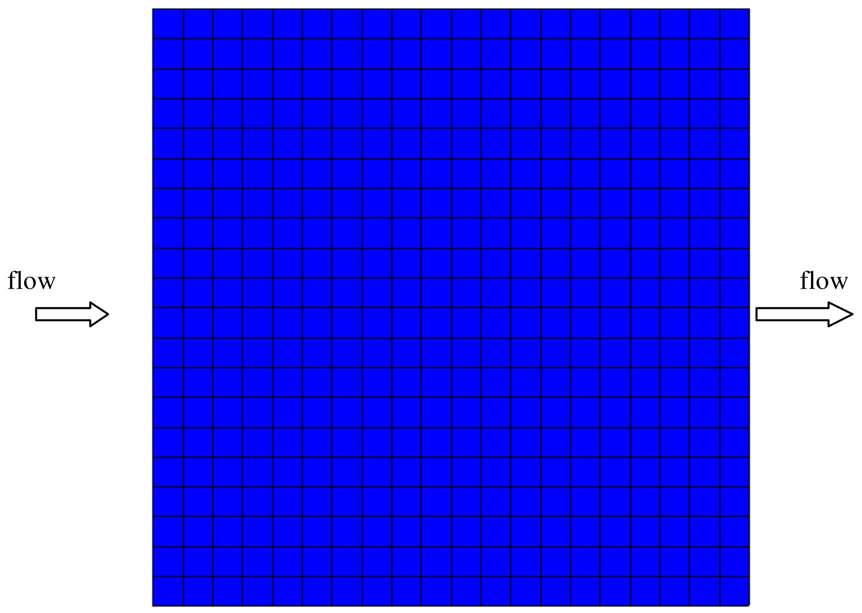


Figure 9: FE mesh

The command file is listed as

```
$*****************************************************************************
PORE-FLOW 2D mold filling flow problem
$*****************************************************************************
PHYSICAL_PROBLEM
$--------------------------------------------------------------------------
  DARCY_EQUATION: ON
     BODY_FORCES: ON, GX:0., GY:0., GZ:0.
  END_DARCY
$--------------------------------------------------------------------------
  FREE_SURFACE: ON
    WICKING: OFF
    DUAL_SCALE: OFF
  END_FREE_SURFACE
$--------------------------------------------------------------------------
END_PHYSICAL_PROBLEM
```

```
$*****************************************************************************
PROPERTIES
 NUM_OF_SETS=1
$---------------------------------------------------------------------------
 SET 1
   DENSITY: 860
   VISCOSITY: .244
   POROSITY: .5
   PERMEABILITY:  KX=1e-7, KY=1e-7, KZ=1e-7, ROTATION=0
   THICKNESS: 1.  ! valid for 2D filling analysis
 ENDSET 1
END_PROPERTIES
$*****************************************************************************
MESH_DATA
$---------------------------------------------------------------------------
   DIMENSIONS
     NUMBER_NODES:      441
     NUMBER_ELEMENTS:   400
     SPACE:   2D
     ELEMENT_TYPE: QUAD 4
     AXISYMMETRY:   NO
   END_DIMENSIONS
$---------------------------------------------------------------------------
   GEOMETRY
     INCLUDE C:\POREFLOW\example\Mold_Geom_2D.dat
   END_GEOMETRY
$---------------------------------------------------------------------------
END_MESH_DATA
$*****************************************************************************
BOUNDARY_CONDITIONS
$---------------------------------------------------------------------------
  INITIAL_CONDITIONS
    VX=0. VY=0. VZ=0.
  END_INITIAL
  INCLUDE C:\POREFLOW\example\Mold_BC_2D.dat
$---------------------------------------------------------------------------
END_BOUNDARY_CONDITIONS
$*****************************************************************************
NUMERICAL_TREATMENT
  TIME_DATA:  INITIAL_TIME=0.0 FINAL_TIME=5 TIME_STEP_SIZE=1. THETA=0.5
  INTEGRATION_POINTS: 4
  ITERATION: PICARD
  ERROR: 5.D-5
  OUT_OF_CORE: OFF
END_NUMERICAL_TREATMENT
$*****************************************************************************
OUTPUT
  FILE_NAME: Mold2D_RESULT.dat
  FREQUENCY:  STEP=5
  MUMPS_INFORMATION:Off
END_OUTPUT
$*****************************************************************************
```

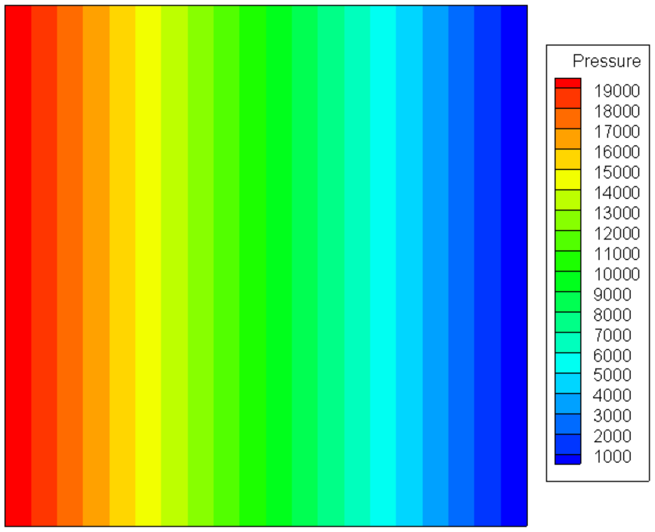


Figure 10: Pressure distribution

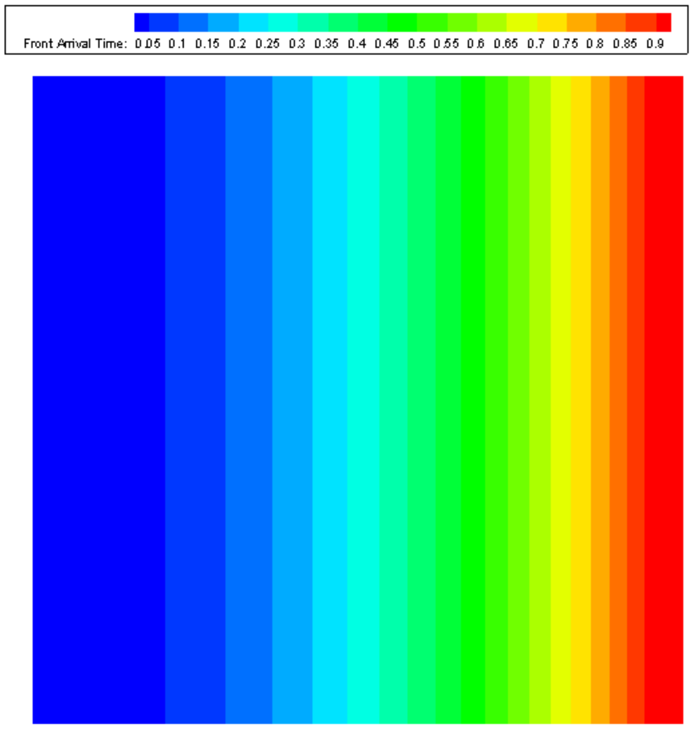


Figure 11: Liquid front location versus time

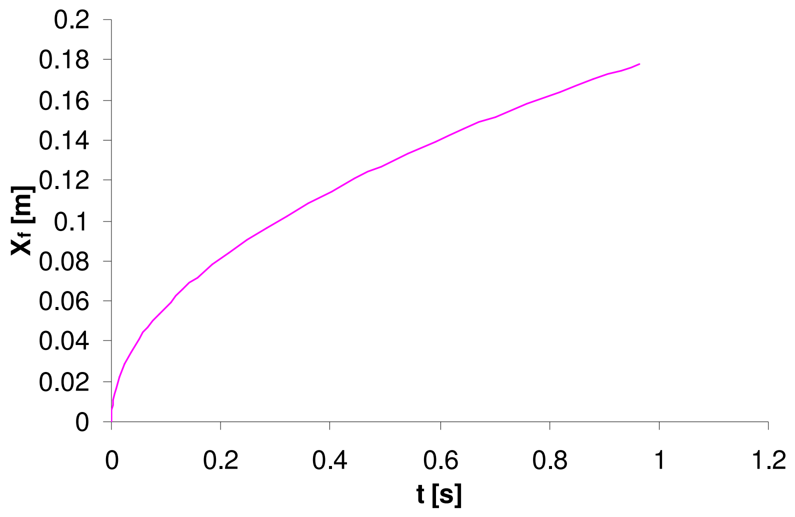


Figure 12: Liquid-front location versus time

More simulation results for mold filling

Geometry, Grid and Boundary Conditions: Initial pressure P=20000

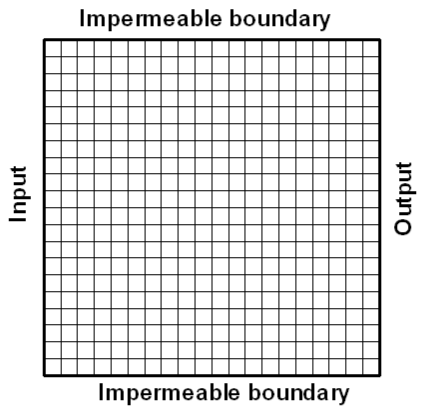


Fig 13 Boundary and Grid

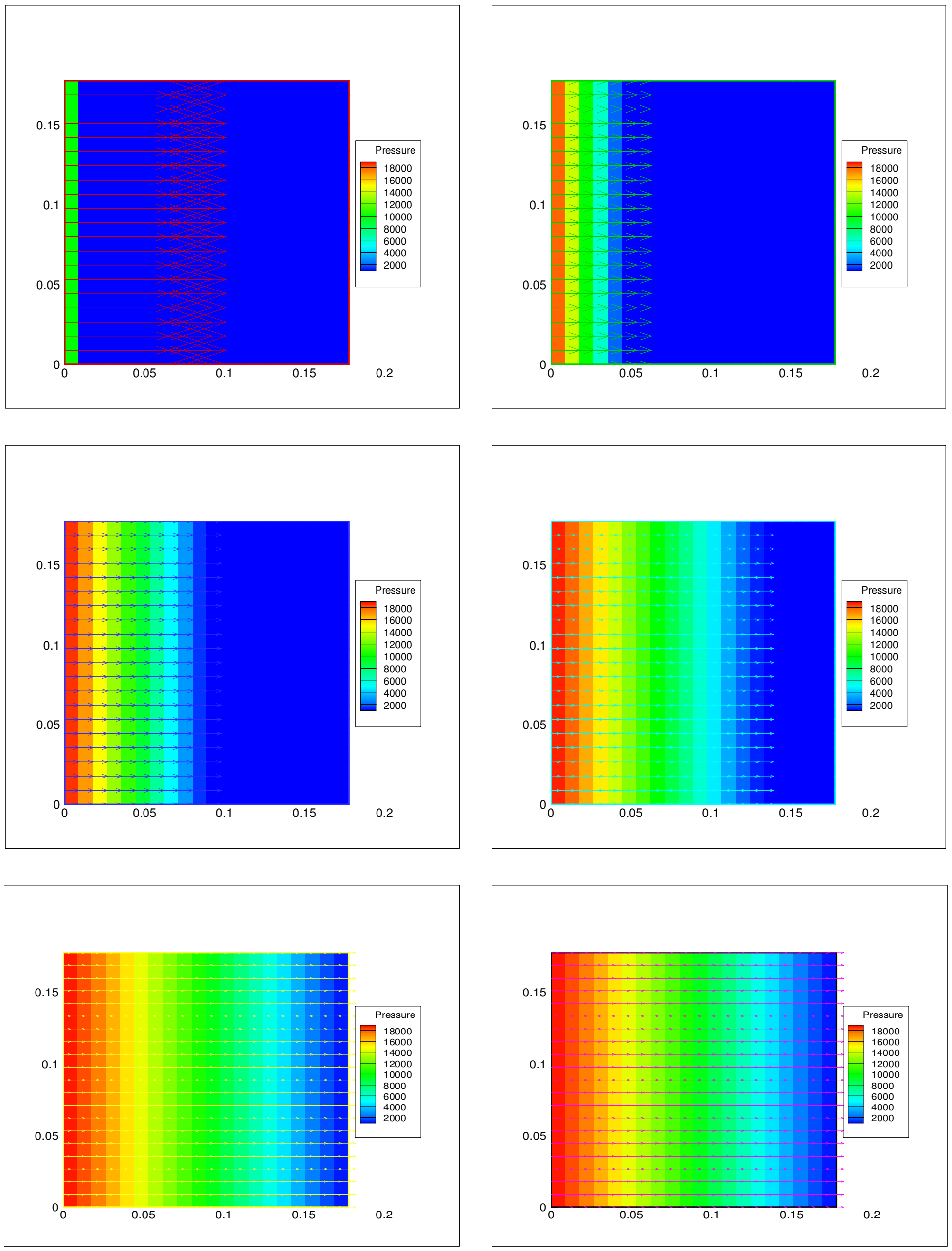


Fig 14 Distribution of Velocity and Pressure in a porous media

Rectangular Geometry Results:

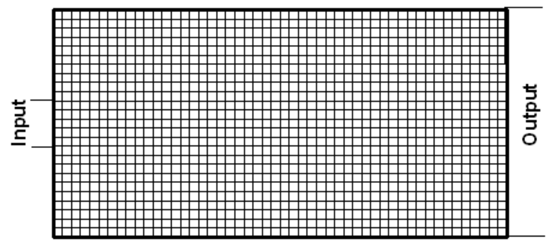


Fig 15 Boundary and Grid

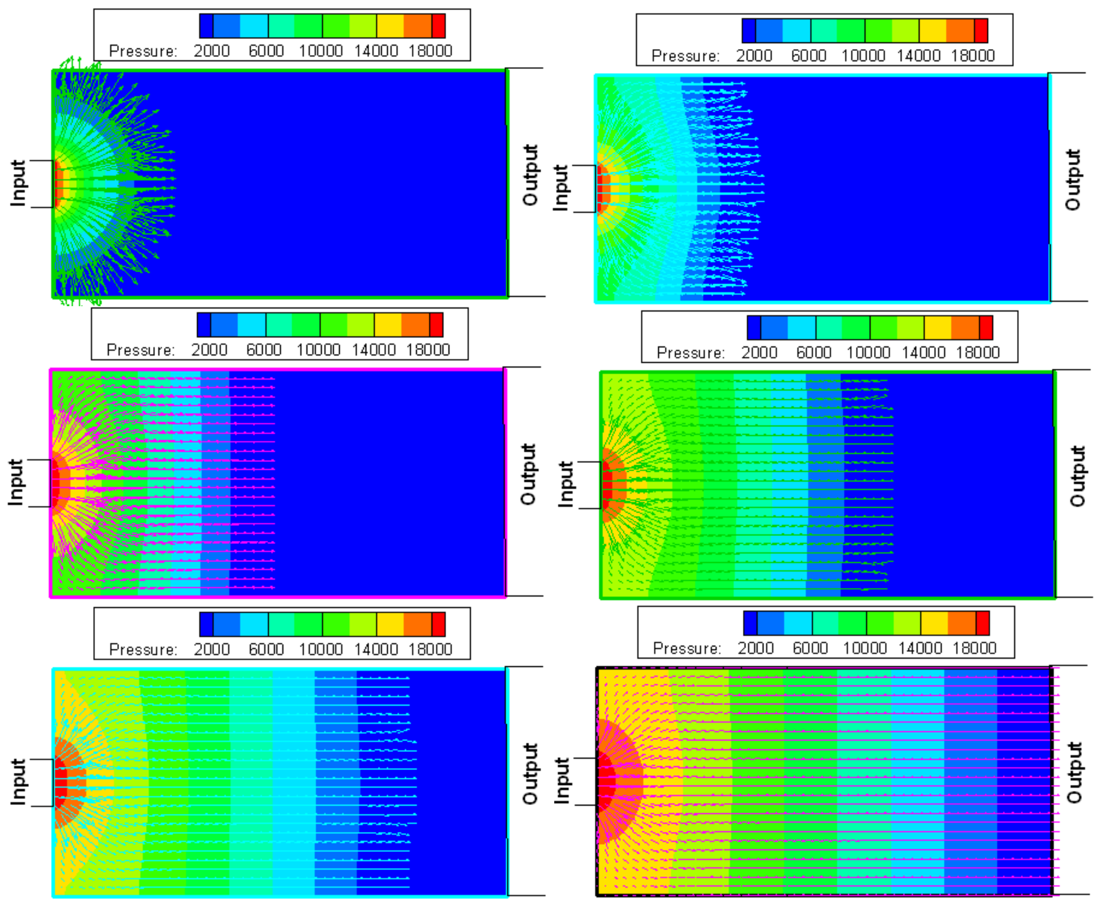


Fig 16 Distribution of Velocity and Pressure in a porous media

3-D Square Geometry Results:

Geometry, Grid and Boundary Conditions: Initial pressure P=20000

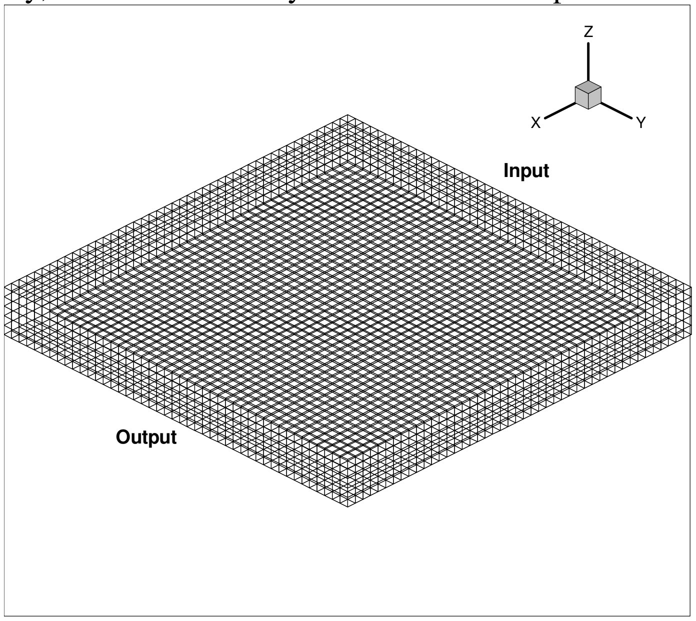
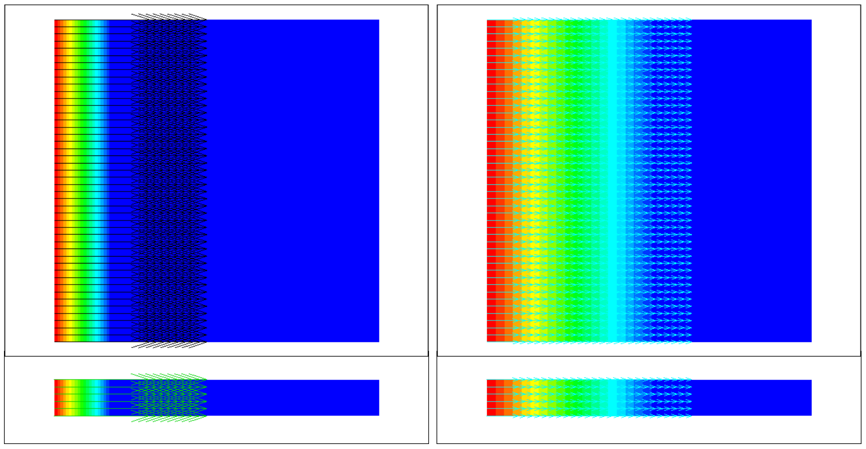


Fig 17 Boundary and Grid



upper map is x-y plane
lower map is y-z plane

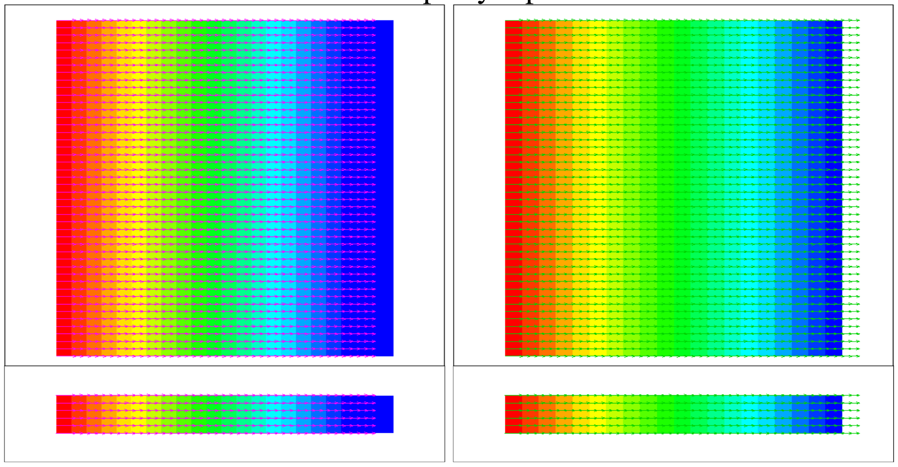


Fig 18 Distribution of Velocity and Pressure in a porous media

3-D Rectangular Geometry Results:
Geometry, Grid and Boundary Conditions: Initial pressure P=20000

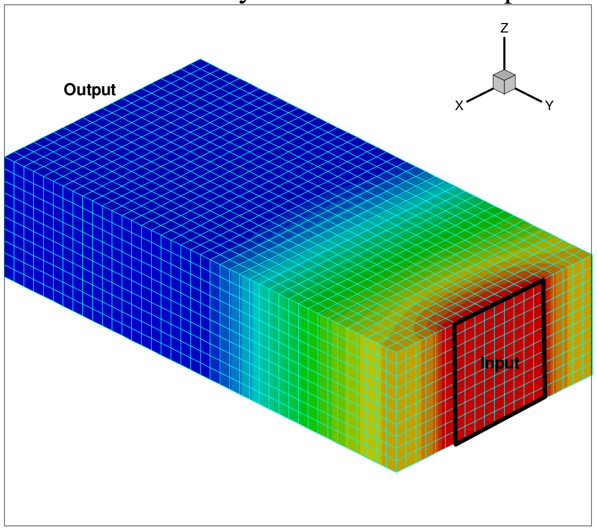


Fig 19 Boundary and Grid

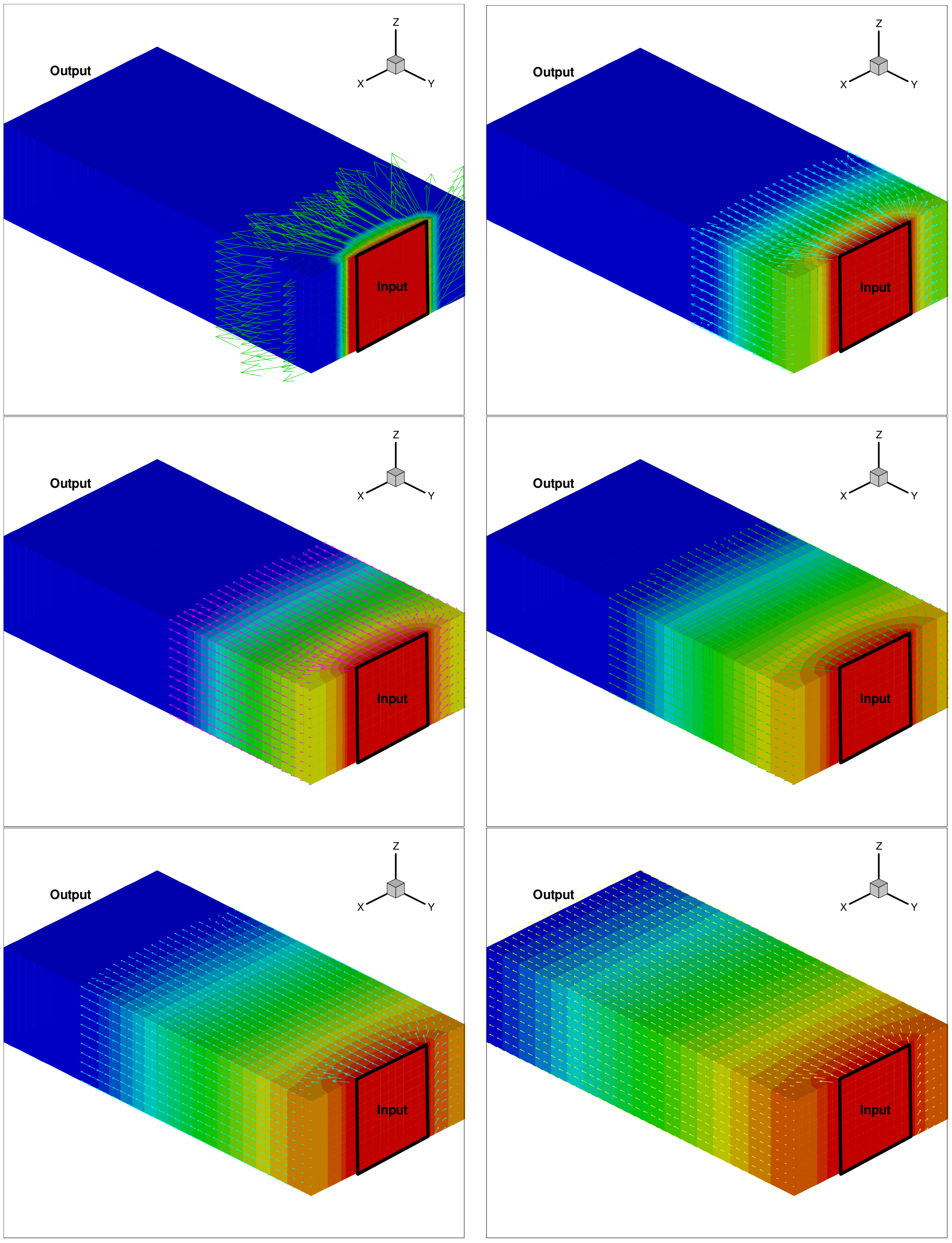


Fig 20 Distribution of Velocity and Pressure in a porous media

10.6 Fast LCM simulation of unsaturated flow in duel-scale fibermats

The problem is a mold of 7”×7” (0.1778×0.1778m) mold that is filling with a resin or a test liquid. The porosity is 0.4742 and the density and viscosity of the liquid are 860 kg/m^3 and 0.1875 Pa.s, respectively. The permeabilities of fibermats are 7.303e-10 m^2 and 6.634e-10 m^2 along x and y directions and the injection pressure is 20 kPa. We want to study flow of resin in such a duel scale mold. The domain is discretized using FE mesh as shown in Figure 9. For this problem, the Darcy equation and continuity equations are solved to find the pressure distribution using fast LCM mold filling simulation[2].

The command file is listed as

```
$**************************************************************************
PORE-FLOW 2D dual-scale flow problem using fast algorithm
$**************************************************************************
PHYSICAL_PROBLEM
$--------------------------------------------------------------------------
  DARCY_EQUATION: ON
     BODY_FORCES: ON,GX:0., GY:-9.8, GZ:0.
  END_DARCY
$--------------------------------------------------------------------------
  FREE_SURFACE: ON
    WICKING: OFF
    DUAL_SCALE:ON
    SINK_MODEL: LUMPED
  END_FREE_SURFACE
$--------------------------------------------------------------------------
END_PHYSICAL_PROBLEM
$**************************************************************************
PROPERTIES
 NUM_OF_SETS=1
$--------------------------------------------------------------------------
 SET 1
   DENSITY: 860
   VISCOSITY: .1875
   POROSITY: .4742
   PERMEABILITY:  KX=7.303e-10, KY=6.634e-10, KZ=1e-7, ROTATION=0
   THICKNESS: 1.  ! valid for 2D filling analysis
   SINK_FUNCTION: A1=4.25773, A2=1.02833, A3=1.28814, B=19593.45
 ENDSET 1
END_PROPERTIES
$**************************************************************************
MESH_DATA
$--------------------------------------------------------------------------
   DIMENSIONS
```

[2] For detail about this method, see following paper:
Hua Tan, Pillai, K.M., Fast LCM Simulation of Unsaturated Flow in Dual-Scale Fiber Mats Using the Imbibition Characteristics of a Fabric-Based Unit Cell, Polymer Composites, Polymer Composites, 31(10), 1790–1807, 2010.

```
    NUMBER_NODES:      441
    NUMBER_ELEMENTS:   400
    SPACE:   2D
    ELEMENT_TYPE: QUAD 4
    AXISYMMETRY:   NO
  END_DIMENSIONS
$--------------------------------------------------------------------------
  GEOMETRY
    INCLUDE C:\POREFLOW\sourcecode_ver1_01\Mold_Geom_2D.dat
  END_GEOMETRY
$--------------------------------------------------------------------------
END_MESH_DATA
$******************************************************************************
BOUNDARY_CONDITIONS
$--------------------------------------------------------------------------
  INITIAL_CONDITIONS
    VX=0. VY=0. VZ=0.
  END_INITIAL
  INCLUDE C:\POREFLOW\sourcecode_ver1_01\Mold_BC_2D.dat
$--------------------------------------------------------------------------
END_BOUNDARY_CONDITIONS
$******************************************************************************
NUMERICAL_TREATMENT
  TIME_DATA:  INITIAL_TIME=0.0 FINAL_TIME=5 TIME_STEP_SIZE=1. THETA=0.5
  INTEGRATION_POINTS: 4
  ITERATION: PICARD
  ERROR: 5.D-5
  OUT_OF_CORE: OFF
END_NUMERICAL_TREATMENT
$******************************************************************************
OUTPUT
  FILE_NAME: Duel_fast_results.out
  FREQUENCY:  STEP=20
  MUMPS_INFORMATION:Off
END_OUTPUT
$******************************************************************************
```

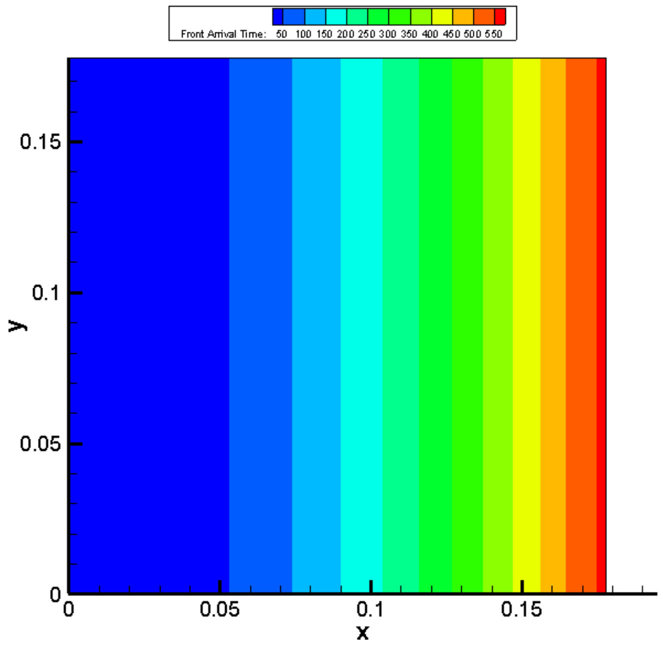


Figure 21: Liquid front movement

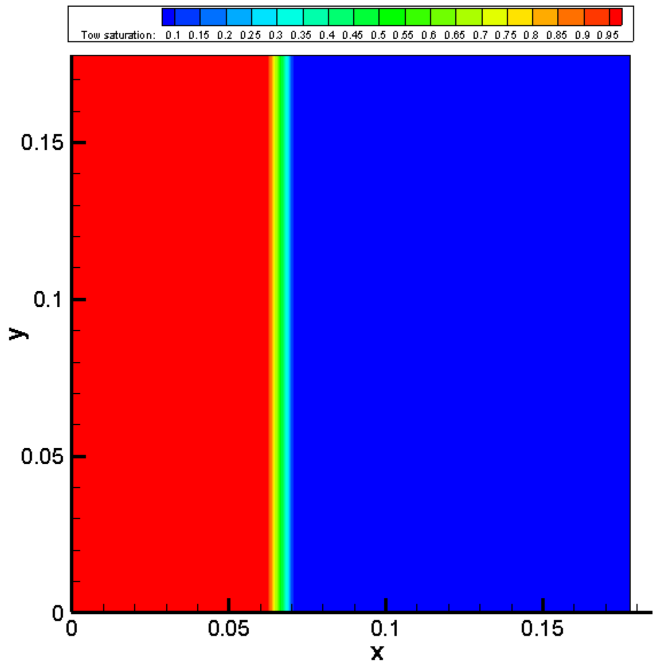
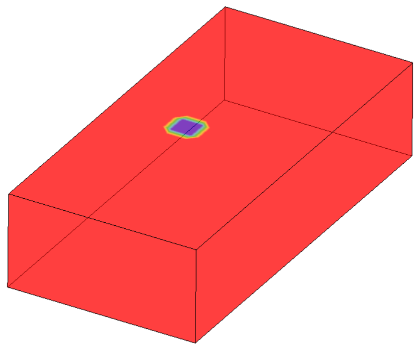
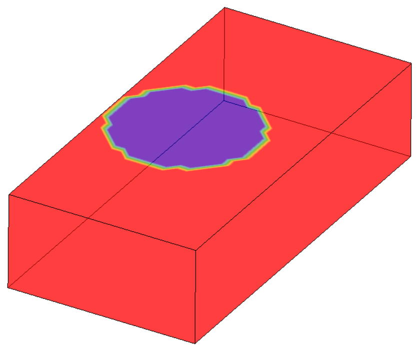


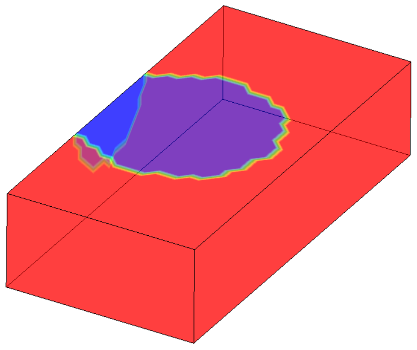
Figure 22: Tow saturation as liquid front moves

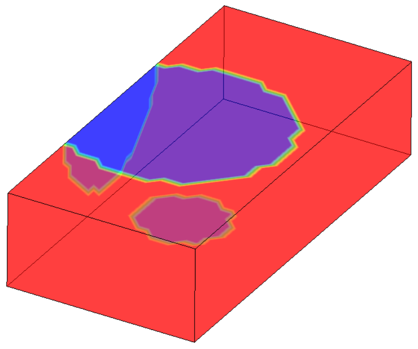
10.7 Flow Modeling in a Diaper/Paper

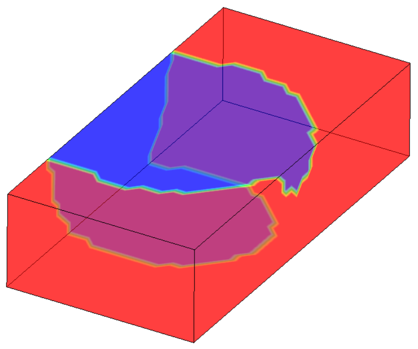
POREFLOW can also model wicking and absorption of liquids into porous substrates. Diapers are porous substances that absorb and retain liquid at a fast rate. The driving force that makes liquid get absorbed by diapers is the capillary pressure. Here we considered a simple block (representing a thick sheet of paper or a section of diaper) of size 20×10×5 mm with a permeability of 2.34e-9 m2 in all directions and a porosity of 0.75. The capillary pressure, which acts on the liquid front, is taken to be 2000 Pa. Here are a series of pictures showing the filling pattern (Blue-wet region, red-dry region):

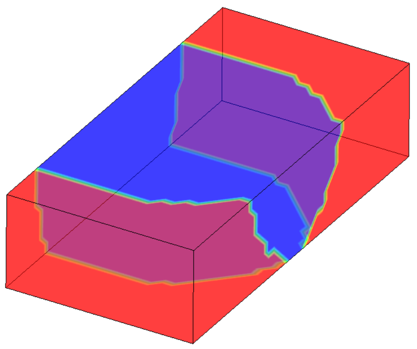

t = 0 s

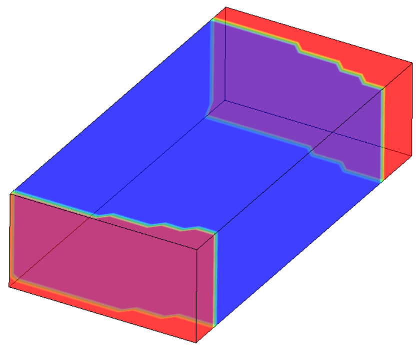

t = 5 s

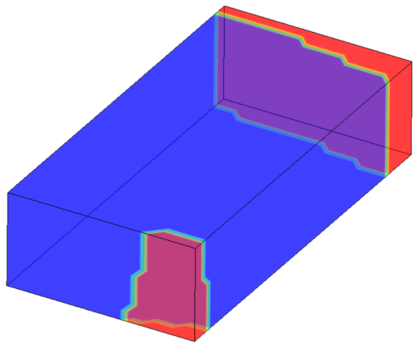

t = 14 s

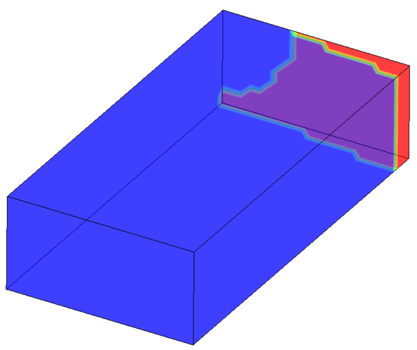

t = 25 s


t = 42 s


t = 58 s


t = 93 s


t = 107 s


t = 121 s