

# OCTA

Integrated simulation system for soft materials

Multi-Phase Dynamics Program

Muffin

version 4.1

User's Manual

- Volume IV -

Micro Fluidics Chip Simulator

MEMFluid

OCTA User's Group

March 03 2005

## **Authors of the Manual**

Tatsuya Yamaue

## **Programers**

Tatsuya Yamaue and Makoto Sasaki

## **Version 4.1 release**

Programer, Authors of the Manual    Tatsuya Yamaue

## **Acknowledgment**

This work is supported by the national project, which has been entrusted to the Japan Chemical Innovation Institute (JCII) by the New Energy and Industrial Technology Development Organization (NEDO) under METI's Program for the Scientific Technology Development for Industries that Creates New Industries.

This work is also partially supported by CREST-JST (Japan Science and Technology Agency) from 2003FY.

Copyright ©2000-2005 OCTA Licensing Committee    All rights reserved.

# Contents

<b>1</b>	<b>Theoretical Background of MEMFluid</b>	<b>1</b>
1.1	What's MEMFluid ?	1
1.2	Basic equations of MEMFluid	2
1.2.1	Notation of parameters and definition of variables	2
1.2.2	Equations of time evolution	2
1.2.3	Dimensionless expression for parameters and variables	3
1.2.4	Dimensionless expression for time evolution equations	4
1.3	Formularization by the Finite Element Method (FEM) of MEMFluid	4
1.4	Boundary Conditions of MEMFluid	5
1.4.1	Boundary conditions of Concentration $C_\alpha$	5
1.4.2	Boundary conditions of Flux $\mathbf{j}_\alpha$ and Volume Force $\mathbf{K}$	6
1.4.3	Boundary conditions of Velocity $\mathbf{v}$	6
1.4.4	Boundary conditions of Pressure $p$	8
1.4.5	Boundary conditions of Electric Potential $\Phi$	9
1.4.6	Boundary conditions of Charge Density $\rho$ and $\sigma$	10
1.4.7	Boundary conditions of Dielectric Constant $\epsilon$	10
1.4.8	Boundary conditions of Viscosity $\eta$	11
<b>2</b>	<b>Sample Problems of MEMFluid</b>	<b>13</b>
2.1	Sample Problems for Micro-Reactor Chips of MEMFluid	13
2.1.1	Application 1: Detection efficiency of the flow injection microchip by pressure gradient	13
2.1.2	Application 2: Detection efficiency of the flow injection microchip by electro-osmosis	17
<b>3</b>	<b>Operation Guide of MEMFluid</b>	<b>23</b>
3.1	Input Parameters of MEMFluid	23
3.1.1	Solver control parameters of MEMFluid	23
3.1.2	Physical parameters of MEMFluid	23
3.2	Fields and Commands for Fields of MEMFluid	24
3.2.1	List of fields for MEMFluid	24
3.2.2	List of commands for MEMFluid	24



# List of Figures

2.1	Application 1 of MEMFluid: The flow injection by a pressure gradient and the chemical reaction	13
2.2	Application 1 of MEMFluid: The concentration distribution of ions in the flow injection by a pressure gradient . . . . .	16
2.3	Application 1 of MEMFluid: Temporal change of the reaction generation rate in the flow injection by a pressure gradient . . . . .	17
2.4	Application 2 of MEMFluid: The concentration distribution of ions by the flow injection under the electric field . . . . .	20
2.5	Application 2 of MEMFluid: Temporal change of the reaction generation rate in the flow injection under the electric field . . . . .	21

# Chapter 1

## Theoretical Background of MEMFluid

### 1.1 What's MEMFluid ?

The Micro Electro Chemical Fluidics Chip Simulator "MEMFluid" is a simulator which treats diffusion and flow accompanied by the chemical reaction of the electrolyte fluid in the scale between meso and macro ( $1\mu m \sim 1mm$ ). Analysis of the flow action inside micro fluidity chips, such as MEMS (micro electro mechanical system) and micro-reactors and micro-TAS (total analysis system), is the target of this simulator.

The big difference from the electrolyte simulator Electrolyte is the scale of the target. The Electrolyte is the simulator of the comparatively micro scale in order to treat formation of the electric double layer near the electrolytic interface, electric osmosis, generation of flow potential, in the scale of Bjerrum length ( $\sim 10nm$ ). MEMFluid treats flow and diffusion with the scale between meso and macro ( $1\mu m \sim 1mm$ ) as mentioned above. On this scale, the electric charge neutrality is always satisfied at all bulk points and the important phenomena in interfaces, such as a formation of the electric double layer, are treated as boundary conditions. The main features are listed below.

1. The diffusion and flow by the imposed electric field or pressure field to a electrolyte fluid.  
However, about the electro osmosis by an external electric field, the velocity at a boundary is given as boundary conditions. Following two are incorporated as methods of calculating the velocity field at a boundary.
  - Helmholtz-Smoluchowski equation  
Following two are incorporated to calculate the zeta potential at the boundary which is needed in Helmholtz-Smoluchowski equation.
    - Measured value, such as an experimental value, is given with a parameter.
    - A surface electric charge is given and calculated from the strict solution of a one-dimensional Poisson-Boltzmann equation.
  - An electrolyte fluid simulator Electrolyte simulator can be incorporated. (Zooming)  
Although it takes much calculation time, it is highly precise.
2. Formation of laminar flow in the small space of the size of  $1 \sim 100\mu m$ .  
Since Reynolds number becomes small for the fluid in a micro scale, a turbulent flow is not generated but laminar flow forms. Muffin solves the flow of small Reynolds number by Stokes approximation, and is suitable for the simulation of the flow action in such a micro scale.
3. The chemical reaction in a laminar flow, and a diffusion flow.  
The following two kinds of the chemical reaction are incorporated.
  - Component A changes with reactants (component which is not treated in the calculation) to Component B. :  $A \rightarrow B$
  - Component A and Component B react and Component C generates. :  $A + B \rightarrow C$
4. Separation and extraction of the reaction product by the difference of the chemical potentials between substances.

The  $\chi$  parameter between each component is introduced into the free energy, and the phase separation and extraction can be treated.

Based on the above functions, it is possible to make applicable to the analysis of the micro fluidity chip in which research and development are done actively in recent years.

## 1.2 Basic equations of MEMFluid

### 1.2.1 Notation of parameters and definitioin of variables

Notation of symbols for MEMFluid simulator are as follows:

Notation of parameters	Meaning of parameters
$e$	Unit of charge ( $= 1.602 \times 10^{-19}$ C(oulomb))
$\epsilon_o$	Dielectric constant of vacuum ( $= 8.854 \times 10^{-12}$ C <sup>2</sup> N <sup>-1</sup> m <sup>-2</sup> )
$\epsilon_r$	Relative dielectric constant of the pure water ( $\epsilon_r = 78.2$ )
$\eta_w$	Viscosity of water ( $= 0.89 \times 10^{-3}$ Pa · sec $= 0.89 \times 10^{-2}$ Poise )
$w_0$	Volume of a water molecule
$N_c$	Total Number of Component of ions in the electrolyte solution
$Z_\alpha$	Valency of $\alpha$ -ion
$D_\alpha$	Diffusion Constant of $\alpha$ - ion.
$w_\alpha$	Volume of the $\alpha$ - ion.
$\chi_{\alpha\alpha'}$	$\chi$ -parameter ( $\chi_{\alpha\alpha} \equiv 0$ )
$k_B T$	Thermal energy per 1 particle. ( $1k_B T = 4.12 \times 10^{-21}$ J at $T = 298\text{K}(25^\circ\text{C})$ )
$RT$	Thermal energy per 1 mol particles. ( $1RT = 1k_B N_B T = 2477.7$ J/mol at $T = 298\text{K}(25^\circ\text{C})$ )
Notation of variables	Meaning of variables
$C_\alpha(\mathbf{x})$	concentration field of $\alpha$ ion
$\mathbf{v}(\mathbf{x})$	velocity field
$p(\mathbf{x})$	pressure field
$\Phi(\mathbf{x})$	scalar electric potential
$\mathbf{E}_0$	external field
$\rho_e(\mathbf{x})$	charge density
$\rho(\mathbf{x})$	mass density
$\mathbf{J}_\alpha(\mathbf{x})$	flux of $\alpha$ ion
$\mathbf{K}(\mathbf{x})$	source term in the Stokes equation
$\epsilon(\mathbf{x})$	relative dielectric constants for that of pure water

Here the subscript  $\alpha$  denotes a component of ions, and runs from 0 to  $N_c - 1$ ,  $N_c$  being the total number of component of ions in the solution.

### 1.2.2 Equtions of time evolution

Although the equation which describes MEMFluid is fundamentally similar to the equation of the system of Electrolyte, there are the following three differences.

- The term which describes the chemical reaction is included in the time development equation of ion concentration  $C_\alpha$ .
- The term of the  $\chi$ -parameter for the separation and extraction of reaction products is included.
- The local electric neutrality condition in bulk,  $\sum_\alpha e Z_\alpha C_\alpha = 0$ , is employed.

The equation of concentration  $C_\alpha$  of  $\alpha$ 'th kind of ion (number of ions per unit volume) is described as

$$\frac{\partial C_\alpha}{\partial t} = -\nabla \cdot (\mathbf{v} C_\alpha) - \nabla \cdot \mathbf{j}_\alpha + \sum_\beta R1_{\alpha\beta} C_\beta + \sum_{\beta, \gamma} R2_{\alpha\beta\gamma} C_\beta C_\gamma \quad (1.1)$$

Here, the first term of the right-hand side comes from a hydrodynamic flow, the second term comes from a diffusion, and the third term and the fourth term mean the following chemical reactions, respectively.

- 3rd term :  $\beta \rightarrow \alpha$
- 4th term :  $\beta + \gamma \rightarrow \alpha$

In the above equation,  $\mathbf{j}_\alpha$  is the flux density of each ion,

$$\mathbf{j}_\alpha = C_\alpha(\mathbf{v}_\alpha - \mathbf{v}) = -L_\alpha \left[ k_B T \{ \nabla C_\alpha + \sum_\beta \chi_{\alpha\beta} \frac{w_\beta w_\alpha}{w_0} C_\alpha \nabla C_\beta \} + e Z_\alpha C_\alpha (\nabla \Phi - \mathbf{E}_0) \right] \quad (1.2)$$

where,  $\mathbf{v}_\alpha$  is the velocity of  $\alpha$  ion. Here  $L_\alpha \equiv 1/\xi_\alpha$  is the Onsager transport coefficient.  $\xi_\alpha$  is a friction constant between an ion  $\alpha$  and the solvent.<sup>1</sup>

The Stokes equation of the velocity field is written as follows.

$$\rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \nabla (\eta_w \{ \nabla \mathbf{v} + (\nabla \mathbf{v})^t \}) + \mathbf{K} \quad (1.3)$$

Usually, dynamic balance is realized in an instant in systems with small Reynolds number, such as a system with an enough large viscosity, and a system of a micro fluidity chip. Therefore, in calculating a flow of a time scale longer than the time scale of this balance, it uses the following Oseen equation is used.

$$-\nabla p + \nabla (\eta_w \{ \nabla \mathbf{v} + (\nabla \mathbf{v})^t \}) + \mathbf{K} = 0 \quad (1.4)$$

The MEMFluid solves a velocity field under the incompressible condition  $\nabla \cdot \mathbf{v} = 0$ . Here,  $\mathbf{K}$  is the source term for Stokes equation and calculated as the body force caused by friction between ion fluxes.

$$\mathbf{K} = - \sum_\alpha \left[ k_B T \{ \nabla C_\alpha + \sum_\beta \chi_{\alpha\beta} \frac{w_\beta w_\alpha}{w_0} C_\alpha \nabla C_\beta \} + e Z_\alpha C_\alpha (\nabla \Phi - \mathbf{E}_0) \right] \quad (1.5)$$

$$= - \sum_\alpha k_B T \{ \nabla C_\alpha + \sum_\beta \chi_{\alpha\beta} \frac{w_\beta w_\alpha}{w_0} C_\alpha \nabla C_\beta \} \quad \dots (in \text{ bulk.}) \quad (1.6)$$

The electric potential obeys the Poisson equation given by

$$\Delta \Phi = - \frac{1}{\epsilon_o \epsilon_r} \sum_\alpha e Z_\alpha C_\alpha \quad (1.7)$$

$$= 0 \quad \dots (in \text{ bulk.}) \quad (1.8)$$

In the second equations of eq.(1.6) and eq.(1.8), we applied the local charge neutrality condition in bulk  $\sum_\alpha e Z_\alpha C_\alpha = 0$ .

### 1.2.3 Dimensionless expression for parameters and variables

In this section, we describe units for MEMFluid and dimensionless expressions by the units. We will use the MKSA unit system throughout this section.

#### Units for dimensionless physical quantities

Physical quantity	unit	meaning
length $x$	$l$	system size (ex. = $1.0 \mu m = 1.0 \times 10^{-6} m$ )
diffusion coefficient $D_\alpha$	$D^*$	(ex. = $1.0 \times 10^{-5} cm^2/sec = 1.0 \times 10^{-9} m^2/sec$ )
time $t$	$\tau$	$\tau = l^2/D^*$ (ex. = $1.0 \times 10^{-3} sec$ )
velocity $\mathbf{v}$	$v^*$	$v^* = l/\tau$ (ex. = $1.0 \times 10^3 \mu m/sec = 1.0 \times 10^{-3} m/sec$ )
volume of an ion $w_\alpha$	$w_0$	$w_0 = 1.0 ml/mol$
concentration $C_\alpha$	$C^*$	$C^* = w_0^{-1} = 1.0 mmol/l$
reaction coefficient $R1$	$R1^*$	$R1^* = \tau^{-1}$ (ex. = $1.0 \times 10^3/sec$ )
reaction coefficient $R2$	$R2^*$	$R2^* = \tau^{-1} C^{*-1}$ (ex. = $1.0 \times 10^3 l/mmol/sec$ )
electric potential $\Phi$	$\Phi_0$	(ex. = $1.0 mV$ )
electric field $\mathbf{E}$	$E^*$	$E^* = \Phi_0/l$ (ex. = $1.0 \times 10^3 V/m$ )
pressure $p$	$p^*$	$p^* = \eta_w/\tau$ (ex. = $0.89 Pa$ )
mass density $\rho$	$\rho^*$	(ex. mass density of water $\rho = 1.0 g/cm^3 = 1.0 \times 10^3 kg/m^3$ )

<sup>1</sup>A diffusion flux is usually written to be  $\mathbf{j}_\alpha = -\mathcal{L}_{\alpha\alpha'} \nabla \mu_{\alpha'} = -\mathcal{L}_\alpha \nabla \mu_\alpha$  using a transportation coefficient  $\mathcal{L}_{\alpha\alpha'} = \mathcal{L}_\alpha \delta_{\alpha\alpha'}$  and chemical potential  $\mu_\alpha$ . About the ion concentration dependance of  $\mathcal{L}_\alpha$ , it is formulizing as  $\mathcal{L}_\alpha = L_\alpha w_\alpha C_\alpha$  here so that it may be in agreement with a diffusion coefficient by single phase state. Therefore, Onsager transportation coefficient  $L_\alpha$  of (1.2) is a diffusion constant in the single phase state of ion kind  $\alpha$ .



**Dimensionless parameters**

dimensionless parameter	meanings	input
$R$	ratio of electrostatic energy and thermal energy	$R = e\Phi_0/k_B T$ (ex. $= 3.89 \times 10^{-2}$ )
$D$	$D = D^{(l)}/D^*$ $D^{(l)} = k_B T/6\pi\eta_w(l/6\pi)$	(ex. $D^{(l)} = 4.63 \times 10^{-8} \text{cm}^2/\text{sec}$ ) (ex. $D = 4.63 \times 10^{-3}$ )
$M$	number of paraticles in unit volume	$M = C^* l^3 N_B$ (ex. $= 6.02 \times 10^5$ )
$Re$	Reynolze number	$Re = \rho^* l v^*/\eta_w$ (ex. $= 1.12 \times 10^{-3}$ )
$L$	ratio of the Bjerrum length and system size	$L = l_b/l = e^2/(k_B T \epsilon_o \epsilon_r l)$ (ex. $= 8.89 \times 10^{-3}$ )

**1.2.4 Dimensionless expression for time evolution equations**

The dimensionless equation which describes MEMFluid is shown in the following. All variables become dimensionless using the above-mentioned dimensionless unit. The equation of concentration  $C_\alpha$  of  $\alpha$ 'th kind of ion is described as

$$\frac{\partial C_\alpha}{\partial t} = -\nabla \cdot (\mathbf{v} C_\alpha) - \nabla \cdot \mathbf{j}_\alpha + \sum_{\beta} R 1_{\alpha\beta} C_\beta + \sum_{\beta, \gamma} R 2_{\alpha\beta\gamma} C_\beta C_\gamma. \quad (1.9)$$

In the above equation,  $\mathbf{j}_\alpha$  is the flux density of each ion,

$$\mathbf{j}_\alpha = -D_\alpha \left[ \nabla C_\alpha + \sum_{\beta} \chi_{\alpha\beta} C_\alpha \nabla C_\beta + R Z_\alpha C_\alpha (\nabla \Phi - \mathbf{E}_0) \right]. \quad (1.10)$$

The Stokes equation of the velocity field is written as

$$Re \rho \frac{\partial \mathbf{v}}{\partial t} = -\nabla p + \nabla (\eta_w \{ \nabla \mathbf{v} + (\nabla \mathbf{v})^t \}) + MD \mathbf{K}. \quad (1.11)$$

Here, the Oseen equation of the velocity field is written as follows.

$$-\nabla p + \nabla (\eta_w \{ \nabla \mathbf{v} + (\nabla \mathbf{v})^t \}) + MD \mathbf{K} = 0 \quad (1.12)$$

The incompressible condition is  $\nabla \cdot \mathbf{v} = 0$ . Here,  $\mathbf{K}$  is the source term for Stokes equation and the dimensionless coefficient  $MD$  is about  $MD = 6.02 \times 10^5 \times 4.63 \times 10^{-3} = 2.79 \times 10^3$ .

$$\mathbf{K} = -\sum_{\alpha} \left[ \nabla C_\alpha + \sum_{\beta} \chi_{\alpha\beta} C_\alpha \nabla C_\beta + R Z_\alpha C_\alpha (\nabla \Phi - \mathbf{E}_0) \right] \quad (1.13)$$

$$= -\sum_{\alpha} \left[ \nabla C_\alpha + \sum_{\beta} \chi_{\alpha\beta} C_\alpha \nabla C_\beta \right] \quad \dots (in \text{ bulk.}) \quad (1.14)$$

The electric potential obeys the Poisson equation given by

$$\Delta \Phi = -\frac{M}{R} L \sum_{\alpha} Z_\alpha C_\alpha \quad (1.15)$$

$$= 0 \quad \dots (in \text{ bulk.}) \quad (1.16)$$

In the second equations of eq.(1.14) and eq.(1.16), we applied the local charge neutrality condition in bulk  $\sum_{\alpha} Z_\alpha C_\alpha = 0$ . The dimensionless coefficient  $ML/R$  of Poisson equation is about  $ML/R = 1.38 \times 10^5$ .

**1.3 Formularization by the Finite Element Method (FEM) of MEM-Fluid**

Using the discretization by the finite element method (FEM), the MEMFluid solves the equations mentioned above.

- The finite element method of a three-dimensional Euler picture is used as the calculation technique. A tetrahedron linear-interpolation element is used and discretization by the remainder method with weight by the Galerkin-Ritz method is performed.
- It is aimed at the slow flow. A flow field is calculated by using an approximation which ignores an advection term and considering the Oseen flow which also ignores the inertia term further. The flow-velocity correcting method is used in calculation of a flow field.<sup>2</sup>
- Dirichlet conditions, Neumann conditions, and a periodic boundary condition can be specified as a boundary condition over the quantity of a field (only when a system is a rectangle).

## 1.4 Boundary Conditions of MEMFluid

There are the following in the boundary condition (partial region conditions) which can be imposed on a field in MEMFluid.

- Periodic boundary condition :

Possible only in the mesh of the UNSTRUCTURED\_RECT type. In MEMFluid, since the periodic boundary is treated geometrically and a periodic boundary condition will be automatically applied to all physical quantity if a mesh is made into a periodic, it is not necessary to specify clearly in Input UDF.

- Dirichlet condition :

The conditions which impose a fixed value to a subregion.

- Neumann condition :

The direction component of a normal of the border plane of the gradient vector of physical quantity is given. It is not necessary to give clearly the Neumann conditions (natural boundary conditions) whose component of a normal direction at the border plane of the form of  $\mathbf{n} \cdot \nabla f = 0$  is zero in the discretization by the finite element method.

- Dirichlet conditions using a calculation result :

It is the boundary condition incorporating a more micro physical phenomenon which has taken place on a boundary and near the boundary. For example, in order that an electric double layer may generate near the boundary of electrolyte fluid, a jump (zeta-potential) of electric potential arises. The zeta-potential can be calculated by the strict solution and linear-approximation equation of a one-dimensional Poisson-Boltzmann equation. You can also calculate the zeta-potential using zooming with the Muffin-Electrolyte simulator. When there is the electro-osmosis effect on a boundary under electric fields, the velocity of flow can be calculated using the equation of Helmholtz-Smoluchowski. You can also calculate the velocity on the boundary using zooming with the Muffin-Electrolyte simulator. These calculation results can be used as a value of boundary condition of the velocity field or a value of boundary condition of the electric potential.

### 1.4.1 Boundary conditions of Concentration $C_\alpha$

The boundary conditions which can be set up to a concentration field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED\_RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$C_\alpha(x, y, z) = C_\alpha(x + L_x, y, z)$$

When imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.  $\alpha$  is an index showing a component.

---

<sup>2</sup>Refer to theory chapter of PhaseSeparation\_FEM for the detail of the flow-velocity correcting method.

- **Bulk boundary condition** (Dirichlet condition)

A bulk boundary condition can be imposed to arbitrary border planes. A bulk boundary condition presupposes that it is the bulk value (constant).

$$C_\alpha(x, y, z)|_{Boundary} = \text{Constant}_\alpha$$

Input the number and value of a component into specification of the bulk boundary condition, in initialization at subregion conditions using condition-name "I-CONSTANT-VALUE-FOR \_A-COMPONENT", and through calculation, using condition-name "D-CONSTANT-VALUE-FOR \_A-COMPONENT".

### 1.4.2 Boundary conditions of Flux $j_\alpha$ and Volume Force $K$

The boundary conditions which can be set up to a flux and  $K$  field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED-RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$j_{\alpha x}(x, y, z) = j_{\alpha x}(x + L_x, y, z)$$

When imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.  $\alpha$  is an index showing a component.

- **Wall boundary condition** (Dirichlet condition)

The value of a flux on a wall surface is specified.

$$j_\alpha(x, y, z)|_{wall} = j_{wall}$$

$|_{wall}$  means the value on a wall surface. In order to specify a wall-surface boundary condition, the number and vector quantity of a component are inputted into subregion conditions using condition-name "D-CONSTANT-VALUE-FOR \_A-COMPONENT".

- **Wall boundary condition** (Neumann condition)

Let a gradient perpendicular to a wall surface be zero so that diffusion flux may not occur on a border plane.

$$\mathbf{n} \cdot \mathbf{j}_\alpha(x, y, z)|_{wall} = 0$$

About the boundary which is not a periodic boundary, this condition is imposed automatically and the input of the value of the Neumann conditions of a concentration field is not supported in MEMFluid.

### 1.4.3 Boundary conditions of Velocity $v$

The boundary conditions which can be set up to a velocity field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED-RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$\mathbf{v}(x, y, z) = \mathbf{v}(x + L_x, y, z)$$

When a periodic boundary condition is also imposed in the direction of y, or the direction of z, the same equation is imposed to each direction.  $\alpha$  is an index showing a component.

- **Dirichlet condition**

The boundary condition it is supposed that is moved by velocity  $\mathbf{v}_0$  with a wall surface can be imposed, and it is expressed with the following equations.

$$\mathbf{v}(x, y, z)|_{wall} = \mathbf{v}_0$$

For specifying the velocity field on the boundary in an initialization, "I-VX" (X component is specified), "I-VY" (Y component is specified) or "I-VZ" (Z component is specified) is inputted into a condition

name, and a velocity is inputted into the value of conditions. Or a condition name is made into "I\_VEC" and three values, X, Y, and Z component of a velocity, are inputted into a value in an array. For specifying the velocity field on the boundary through calculation, "D\_VX" (X component is specified), "D\_VY" (Y component is specified) or "D\_VZ" (Z component is specified) is inputted into a condition name, and a velocity is inputted into the value of conditions. Or a condition name is made into "D\_VEC" and three values, X, Y, and Z component of a velocity, are inputted into a value in an array.

- **The calculation result by the electro-osmosis is set as a boundary condition.**

The boundary condition which is supposed to be moved by velocity  $\mathbf{v}_{eo}$  by electro-osmosis under electric fields can be imposed, and it is expressed with the following equations.

$$\mathbf{v}(x, y, z)|_{wall} = \mathbf{v}_{eo}(\Phi, \{C_\alpha\})$$

For specifying this condition in an initialization, "IF\_VX\_ELECTRO\_OSMOSIS" (X component is specified), "IF\_VY\_ELECTRO\_OSMOSIS" (Y component is specified), "IF\_VZ\_ELECTRO\_OSMOSIS" (Z component is specified), or

"IF\_VEC\_ELECTRO\_OSMOSIS" (X, Y, and Z components are specified) is inputted into a condition name. For specifying this condition through calculation, "DF\_VX\_ELECTRO\_OSMOSIS" (X component is specified), "DF\_VY\_ELECTRO\_OSMOSIS" (Y component is specified), "DF\_VZ\_ELECTRO\_OSMOSIS" (Z component is specified), or "DF\_VEC\_ELECTRO\_OSMOSIS" (X, Y, and Z components are specified) is inputted into a condition name. The following is incorporated as a calculus of  $\mathbf{v}_{eo}$ .

– Helmholtz-Smoluchowski equation

$$\mathbf{v}_{eo} = \frac{\epsilon_o \epsilon_r}{\eta_w} \zeta \mathbf{E} \quad (1.17)$$

Here,  $\zeta$  is the zeta-potential and  $\mathbf{E}$  is the electric field on the boundary. The dimensionless equation is as follows.

$$\mathbf{v}_{eo} = \frac{DR^2}{L} \zeta \mathbf{E} \quad (1.18)$$

Here, the dimensionless coefficient  $DR^2/L \sim 7.88 \times 10^{-4}$ .

"HELMHOLTZ\_SMOLUCHOWSKI" is inputted into the head data of the value of conditions in order to use this equation.

Following two are incorporated about the calculus of zeta-potential  $\zeta$  on the boundary which is needed in Helmholtz-Smoluchowski equation.

- \* Give a measured value, such as an experimental value, as a parameter.  
In order to calculate Zeta-potential using this, "ZETA\_POTENTIAL\_INPUT\_PARAM" is inputted into the 2nd data of the value of conditions, and the value of zeta-potential is inputted into the 3rd data.
- \* Calculate from the strict solution of a one-dimensional Poisson-Boltzmann equation.  
In one kind anion-cation system, if the absolute value of a valence is made into  $Z$  and bulk concentration (it equals in both) is made into  $C$ , the strict solution can be written as follows.

$$\zeta = \frac{2k_B T}{Ze} \ln \left[ \frac{\sigma}{(8C\epsilon_o\epsilon_r k_B T)^{1/2}} + \left( \frac{\sigma^2}{8C\epsilon_o\epsilon_r k_B T} + 1 \right)^{1/2} \right] \quad (1.19)$$

The dimensionless equation is as follows.

$$\zeta = \frac{2}{ZR} \ln \left[ \frac{\sigma}{(8C/(ML))^{1/2}} + \left( \frac{\sigma^2}{8C/(ML)} + 1 \right)^{1/2} \right] \quad (1.20)$$

Here, the dimensionless coefficient is about  $2/R = 51.4$ ,  $8/(ML) = 1.50 \times 10^{-3}$ .

"ZETA\_POTENTIAL\_POISSON\_BOLTZMANN" is inputted into the 2nd data of the value of conditions in order to calculate zeta-potential using this. Parameters  $R$ ,  $M$ ,  $L$ , and  $Z$  (valence), the bulk concentration field (Concentration) on a boundary, and the charge-density field (ChargeDensity) of an external boundary are needed.

- \* It is calculated on the boundary modeled by the linear condensor. (It is equivalent to the linear approximation of the strict solution of a 1-dimensional Poisson-Boltzmann equation.)  
3

$$\zeta = \frac{\sigma}{\epsilon_o \epsilon_r \kappa} \quad (1.21)$$

Here,  $\kappa$  is the inverse of the thickness (distance at which electric potential decreases to  $e^{-1}$ ) of an electric double layer.

$$\kappa = \left( \frac{\sum_{\alpha} Z_{\alpha}^2 C_{\alpha} e^2}{\epsilon_o \epsilon_r k_B T} \right)^{1/2} \quad (1.22)$$

The dimensionless equation is as follows.

$$\zeta = \frac{(ML)^{1/2}}{R} \frac{\sigma}{(\sum_{\alpha} Z_{\alpha}^2 C_{\alpha})^{1/2}} \quad (1.23)$$

Here, the dimensionless coefficient is about  $(ML)^{1/2}/R = 1.88 \times 10^3$ .

“ZETA.POTENTIAL.LINEAR.CONDENSOR” is inputted into the 2nd data of the value of conditions in order to calculate a zeta-potential using this. Parameters R, M, L, and Z (valence), the bulk concentration field (Concentration) in a boundary, and the charge-density field (ChargeDensity) of an external boundary are needed.

- Using Electrolyte simulator as small system on the boundary. (ZOOMING)  
It is highly precise although a machine time is large.

“MUFFIN.ELECTROLYTE” is inputted into the head data of the value of conditions in order to use this equation. Parameters D, R, M, L, and Z (valence), the bulk concentration field (Concentration) in a boundary, electric field (ElectricField), and the charge-density field (ChargeDensity) of an external boundary are needed.

- **When a pressure value is set up on a boundary**

A velocity field gradient imposes zero on the boundary.

$$\mathbf{n} \cdot \mathbf{v}(x, y, z)|_{wall} = 0$$

Since it is made automatically, there is no need for specification.

#### 1.4.4 Boundary conditions of Pressure $p$

The boundary conditions which can be set up to a pressure field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED\_RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$p(x, y, z) = p(x + L_x, y, z)$$

Imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.

- **When a velocity value is set up on a boundary (Neumann condition)**

A pressure gradient is zero.

$$\mathbf{n} \cdot \nabla p(x, y, z)|_{Boundary} = 0$$

In order to give such a Neumann boundary condition to a pressure field, a condition name is made into “N” and the values of the gradient are inputted into the head data of the value of conditions.

---

<sup>3</sup>The conditions that a linear approximation is appropriate have low electric potential  $\zeta$ , therefore the charge density is low, and are the case where  $\frac{e|\zeta Z|}{k_B T} \ll 1$  is satisfied. In the case of a 1-1 type electrolyte ( $Z = 1$ ), this condition becomes  $\zeta \leq 25mV$  at a room temperature.

- **Dirichlet condition**

Value  $p_o$  of a pressure is set up at a certain boundary.

$$p(x, y, z)|_{Boundary} = p_o$$

In order to give such a Dirichlet boundary condition to a pressure field, a condition name is made into "D" or "D.CONSTANT.VALUE", and the value of the pressure field in a boundary is inputted into the head data of the value of conditions.

### 1.4.5 Boundary conditions of Electric Potential $\Phi$

The boundary conditions which can be set up to a electric potential field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED.RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$\Phi(x, y, z) = \Phi(x + L_x, y, z)$$

Imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.

- **Give charge density  $\sigma$  on the boundary surface** (Neumann condition)

$$\mathbf{n} \cdot \nabla \Phi(x, y, z)|_{Boundary} = -\sigma/\epsilon_r \epsilon_o$$

Dimensionless equation is the follows.

$$\mathbf{n} \cdot \nabla \Phi(x, y, z)|_{Boundary} = -\sigma ML/R$$

About the boundary which is not a periodic and has no charge, the condition

$$\mathbf{n} \cdot \nabla \Phi(x, y, z)|_{Boundary} = 0$$

in MEMFluid.

- **An electric potential value is set up on a boundary.** (Dirichlet condition)

Value of an electric potential is set up in a certain boundary.

$$\Phi(x, y, z)|_{Boundary} = \Phi_o \quad (= \text{Constant})$$

In order to give such a Dirichlet boundary condition to an electric potential field, a condition name is made into "D" and the value of the field in a boundary is inputted into the head data of the value of conditions.

- **The calculation result of zeta-potential is set as a boundary condition.**

When the charging current to the boundary plane by an external electric-field has arisen in bulk, it is possible to calculate the electric double layer potential (zeta-potential) from the amount of charges of a boundary, and to set up the electric potential on a boundary.

$$\Phi(x, y, z)|_{Boundary} = \Phi_{el}(\Phi_{Applied}, \{C_\alpha\})$$

In order to specify this boundary condition, a condition name is made into "DF\_ZETA.POTENTIAL" and the calculus of zeta-potential is specified into the 2nd data for the electric potential currently impressed to the head data of conditions from the exterior on the boundary. As boundary electric potential to bulk, the value of the zeta-potential subtracted from external electric potential is given by the Dirichlet conditions.

Following four are incorporated about the calculus of the zeta-potential of a boundary.

- Give measured value, such as an experimental value, as a parameter.  
In order to calculate zeta-potential using this, “INPUT\_PARAM” is inputted into the 2nd data of the value of conditions, and the value of zeta-potential is inputted into the 3rd data.
- It is calculated from the strict solution of a one-dimensional Poisson-Boltzmann equation.  
Refer to the term of the boundary condition of a velocity field of concrete expression.  
“POISSON\_BOLTZMANN” is inputted into the 2nd data of the value of conditions in order to calculate zeta-potential using this. Parameters R, M, L, and Z (valence), the bulk concentration field (Concentration) in a boundary, and the charge-density field (ChargeDensity) of an external boundary are needed.
- It calculates on the boundary modeled by the linear condensor. (It is equivalent to the linear approximation of the strict solution of a 1-dimensional Poisson-Boltzmann equation.)  
Refer to the term of the boundary condition of a velocity field about concrete expression.  
“LINEAR\_CONDENSOR” is inputted into the 2nd data of the value of conditions in order to calculate zeta-potential using this. Parameters R, M, L, and Z (valence), the bulk concentration field (Concentration) in a boundary, and the charge-density field (ChargeDensity) of an external boundary are needed.
- Using Electrolyte simulator as small system on the boundary. (ZOOMING)  
“MUFFIN\_ELECTROLYTE” is inputted into the 2nd data of the value of conditions in order to calculate zeta-potential using this. Parameters R, M, L, and Z (valence), the bulk concentration field (Concentration) in a boundary, and the charge-density field (ChargeDensity) of an external boundary are needed.

#### 1.4.6 Boundary conditions of Charge Density $\rho$ and $\sigma$

The boundary conditions which can be set up to a charge density field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED\_RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$\rho(x, y, z) = \rho(x + L_x, y, z)$$

Imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.

- **Dirichlet condition**

The Dirichlet boundary condition which gives a fixed charge can be imposed to arbitrary boundary planes or partial regions. The value is inputted into the specification of the fixed charge conditions in initialization as condition-name ‘I’ or “I\_CONSTANT\_VALUE” at partial region conditions. In order to specify fixed charge conditions, the value is inputted into partial region conditions as condition-name “D” or “D\_CONSTANT\_VALUE”.

#### 1.4.7 Boundary conditions of Dielectric Constant $\epsilon$

The boundary conditions which can be set up to a relative dielectric constant field is as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED\_RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$\epsilon(x, y, z) = \epsilon(x + L_x, y, z)$$

Imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.

- **Dirichlet condition**

The Dirichlet boundary condition which gives a fixed dielectric constant can be imposed to arbitrary partial regions. In an area 1, in water and an area 2, an organic solvent etc. uses this when solvents differ for every area. In order to specify fixed dielectric constant conditions, the value is inputted into partial region conditions as condition-name “D” or “D\_CONSTANT\_VALUE”.

### 1.4.8 Boundary conditions of Viscosity $\eta$

The boundary conditions for the viscosity field are as follows.

- **Periodic boundary condition**

It can be applied, only when a mesh configuration type is UNSTRUCTURED\_RECT. The following equation is imposed when a periodic boundary condition is imposed in the x directions.

$$\eta(x, y, z) = \eta(x + L_x, y, z)$$

Imposing a periodic boundary condition also in the direction of y, or the direction of z, the same equation is imposed to each direction.

- **Dirichlet condition**

The Dirichlet boundary condition which gives a fixed viscosity can be imposed to arbitrary partial regions. In an area 1, in water and an area 2, an organic solvent etc. uses this, when solvents differ for every area. In order to specify fixed viscosity conditions, the value is inputted into partial region conditions as condition-name "D" or "D\_CONSTANT\_VALUE".





## Chapter 2

# Sample Problems of MEMFluid

### 2.1 Sample Problems for Micro-Reactor Chips of MEMFluid

This chapter shows the applications of micro electro mechanical fluid dynamics simulator - MEMFluid - by the finite element method for micro-reactor chips. Input and output UDF files corresponding to these applications are dedicated to the directory `MUFFIN/sample/MEMFluid/EX01,EX02,...` according to the problem, below the directory `MUFFIN/sample/MEMFluid` of the distribution version of Muffin.

#### 2.1.1 Application 1: Detection efficiency of the flow injection microchip by pressure gradient

As shown in Figure 2.1.1, in the system of component A and component B, component A and component B are passed through a capillary by pressure-gradient  $\Delta p$ . The simulation of the time development of the generation rate of the component C generated by a chemical reaction  $A + B \rightarrow C$  is carried out.

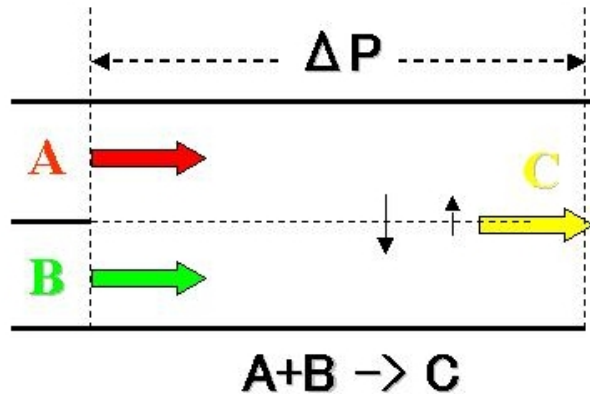


Figure 2.1: Application 1 of MEMFluid: The flow injection by a pressure gradient and the chemical reaction

#### Input UDF file:

`MUFFIN/sample/MEMFluid/EX01/EX01_in.udf`

#### The description of Input UDF file:

Here, a simulation is performed using the value of the nondimensional parameter explained in the theoretical section.

- `parameter.mesh_parameter`:  
Type of mesh : `UNSTRUCTURED_RECT`, 64x32x2 division.

- `parameter.physical_parameter[]` :

Name of Parameters(KEY)	Values
NUMBER_OF_COMPONENTS	3
VISCOSITY	1.0, 1.0, 1.0
Z	1, 1, 2
R	3.89e-2
M	6.02e2
D	4.63e-3
DIELECTRIC_CONSTANT	1.0, 1.0, 1.0
DIFFUSION_COEFFICIENT	1.0, 1.0, 1.0
AVERAGED_ION_CONCENTRATION	0.0, 0.0, 0.0
D.PRESSURE	10.0
REACTION_COEFFICIENT_2_0	0, -1.0, 0, 0, 0, 0, 0, 0, 0
REACTION_COEFFICIENT_2_1	0, -1.0, 0, 0, 0, 0, 0, 0, 0
REACTION_COEFFICIENT_2_2	0, 1.0, 0, 0, 0, 0, 0, 0, 0

- `partial_region[]`

Here, "BOUNDARY\_VERTEX\_XMIN" which is the surface where a capillary joins is divided into two partial regions. The area of the entrance of the capillary of Component A is made into the partial region "BOUNDARY\_VERTEX\_XMIN\_YUPPER", and the area of the entrance of the capillary of Component B is made into the partial region "BOUNDARY\_VERTEX\_XMIN\_YLOWER".

- `region.condition[]`

Partial region conditions are shown in the following.

partial region	field	condition	values
BOUNDARY_VERTEX_ZMIN	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_ZMAX	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_XMIN	Velocity	D_VY	0.0
BOUNDARY_VERTEX_XMIN	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_XMAX	Velocity	D_VY	0.0
BOUNDARY_VERTEX_XMAX	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_YMIN	Velocity	D_VEC	0.0, 0.0, 0.0
BOUNDARY_VERTEX_YMAX	Velocity	D_VEC	0.0, 0.0, 0.0
BOUNDARY_VERTEX_XMIN	Pressure	D	\$(D.PRESSURE)
BOUNDARY_VERTEX_XMAX	Pressure	D	0.0
BOUNDARY_VERTEX_XMIN	Concentration	D.CONSTANT_VALUE _FOR_A_COMPONENT	2, 0.0
BOUNDARY_VERTEX_XMIN _YUPPER	Concentration	D.CONSTANT_VALUE _FOR_A_COMPONENT	0, 1.0
BOUNDARY_VERTEX_XMIN _YLOWER	Concentration	D.CONSTANT_VALUE _FOR_A_COMPONENT	1, 1.0

- `dynamics_manager.registered_field[]`

All the fields that can be used by MEMFluid except for the "ElectricPotential" and "ElectricField" are specified.

- `dynamics_manager.procedures.table_for_initialization[].command.list[]`

Initialization procedure "INITIALIZE:REACTOR:UNDER\_PRESSURE" is defined as follows.

field	command for initialization
Pressure	INITIALIZE:TO_ZERO
Viscosity	UPDATE:TO_CONSTANT
Velocity	INITIALIZE:TO_ZERO
Concentration	INITIALIZE:UNIFORM
Obstacle	INITIALIZE:TO_ZERO
K_Field	SOLVE:WITH_CHARGE_NEUTRALITY
Velocity	SOLVE:STOKES_EQUATION_AND_PRESSURE
Concentration	INITIALIZE:PARTIAL_REGION_CONDITION

The procedure of initialization is as follows.

- The pressure field, the velocity field, and the ion concentration field are initialized to 0.0. The viscosity field is initialized to constant-value 1.0.
  - The volume force field is calculated under the charge neutrality condition. (Here, all become 0.0.)
  - The Stokes equation is solved under boundary conditions and the velocity field and the pressure field of a stationary state are searched.
  - The partial region conditions are imposed on the ion concentration field, and the bulk conditions of Component A and Component B are given to the surface where a capillary joins.
- `dynamics_manager.procedures_table_for_evolution[].command_list[]`

Time evolution procedure”SOLVE:REACTOR:UNDER\_PRESSURE” is defined as follows.

field	command for evolution
K_Field	SOLVE:WITH_CHARGE_NEUTRALITY
Flux	SOLVE:WITH_CHARGE_NEUTRALITY
Velocity	SOLVE:VELOCITY_AND_PRESSURE
Concentration	SOLVE:WITH_FLOW:WITH_REACTION
Concentration	UPDATE:PARTIAL_REGION_CONDITION

The procedure of time evolution is as follows.

- The volume force field is solved under the charge neutrality conditions.
- The diffusion flux of each component of ion is solved.
- Using the volume force field, the Navier-Stokes equation is solved and the time evolution of the velocity field and the pressure field is carried out.
- The time evolution of the ion concentration field is carried out on the advection, diffusion, and the chemical reaction equation using the diffusion flux field and the velocity field.
- The partial region conditions are imposed on the ion concentration field, and the bulk conditions in the surface where the capillaries joins are reconfigured.

## Results of simulation

The example which displays the ion concentration field (Concentration) by the view function of GOURMET is shown in Fig.2.1.1. From the above figure, the ion concentration distribution of component A, component B, and component C, in time  $t = 10.0, 25.0$ , and  $50.0$  from the left figure are displayed. MeshfieldShow.py is used for the Python script for a display.

The time change of the average concentration of each component ion or the velocity field on ”BOUNDARY\_VERTEX\_XMAX” surface of the down-stream side of the capillary are analyzed by a python program. Then, the graph which indicates the time change of the reaction generation rate is shown in Fig.2.1.1. Here, the graph sheet (GraphSheet[]) is created, using memfluid/ranalyze\_reactor\_pressure.py as a python program for analysis. The memfluid/plot\_average\_reaction\_rate.gp program is used for the gnuplot script for a graphical representation.

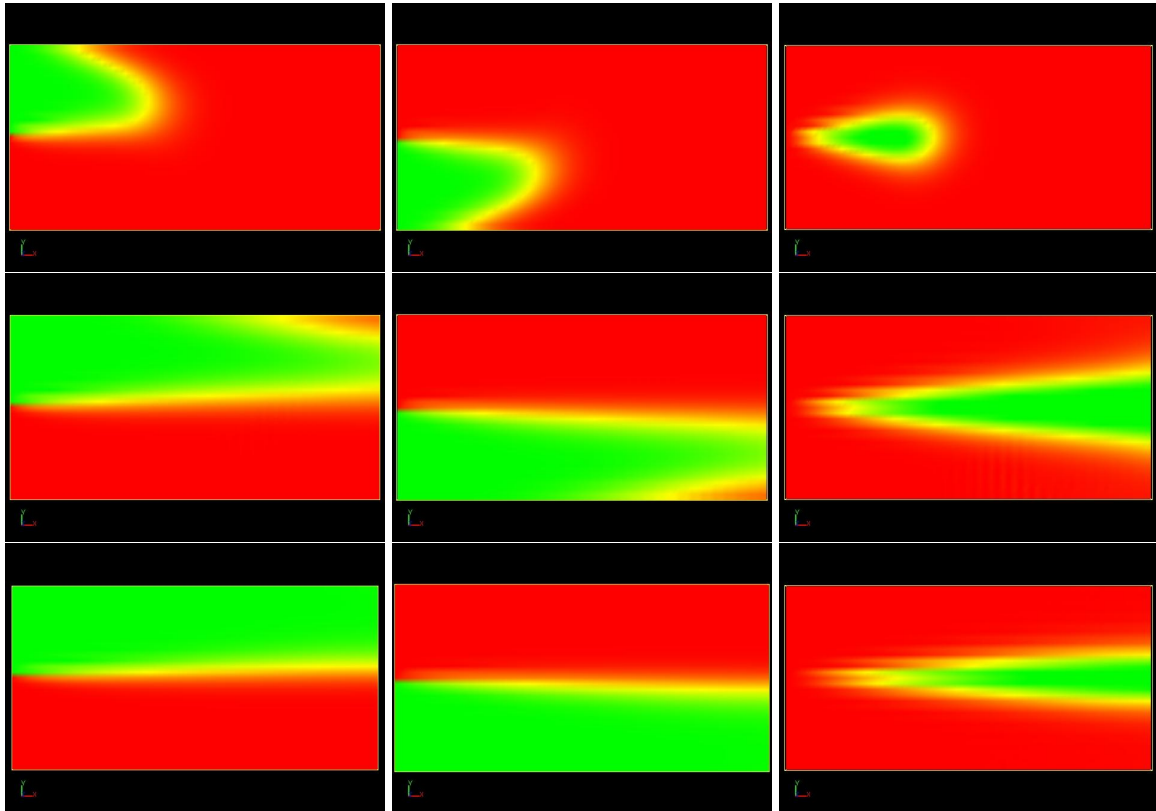


Figure 2.2: Application 1 of MEMFluid: The concentration distribution of ions in the flow injection by a pressure gradient

Figure 2.3: Application 1 of MEMFluid: Temporal change of the reaction generation rate in the flow injection by a pressure gradient

### 2.1.2 Application 2: Detection efficiency of the flow injection microchip by electro-osmosis

Here, in the system in which the capillary of component A and component B join, component A and component B are passed by the electro-osmosis under the electric field, the simulation of the time development of the generation rate of the component C generated by chemical reaction  $A + B \rightarrow C$  is carried out.

**Input UDF file:**

MUFFIN/sample/MEMFluid/EX01/EX02\_in.udf

**The description of Input UDF file:**

Here, a simulation is performed using the value of the nondimensional parameter explained in the theoretical section.

- parameter.mesh\_parameter:

Type of mesh : UNSTRUCTURED\_RECT, 64x32x2 division.

- parameter.physical\_parameter[] :

Name of Parameters(KEY)	Values
NUMBER_OF_COMPONENTS	3
VISCOSITY	1.0, 1.0, 1.0
Z	1, 1, 2
R	3.89e-2
M	6.02e2
D	4.63e-3
L	8.89e-3
DIELECTRIC_CONSTANT	1.0, 1.0, 1.0
DIFFUSION_COEFFICIENT	1.0, 1.0, 1.0
AVERAGED_ION_CONCENTRATION	0.0, 0.0, 0.0
EXTERNAL_ELECTRIC_FIELD	10.0, 0.0, 0.0
ZETA_POTENTIAL	100.0
REACTION_COEFFICIENT_2.0	0, -1.0, 0, 0, 0, 0, 0, 0, 0
REACTION_COEFFICIENT_2.1	0, -1.0, 0, 0, 0, 0, 0, 0, 0
REACTION_COEFFICIENT_2.2	0, 1.0, 0, 0, 0, 0, 0, 0, 0

- partial\_region[]

Here, "BOUNDARY\_VERTEX\_XMIN" which is the junction of two capillaries is divided into two partial regions. The area of the entrance of the capillary of component A is made into the partial region "BOUNDARY\_VERTEX\_XMIN\_YUPPER", and the area of the entrance of the capillary of component B is made into the partial region "BOUNDARY\_VERTEX\_XMIN\_YLOWER".

- region.condition[]

Partial region conditions are shown in the following.

partial region	field	condition	values
BOUNDARY_VERTEX_ZMIN	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_ZMAX	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_XMIN	Velocity	D_VY	0.0
BOUNDARY_VERTEX_XMIN	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_XMAX	Velocity	D_VY	0.0
BOUNDARY_VERTEX_XMAX	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_YMIN	Velocity	DF_VX _ELECTRO_OSMOSIS	HELMHOLTZ _SMOLUCHOWSKI, ZETA.POTENTIAL _INPUT.PARAM, \$(ZETA.POTENTIAL)
BOUNDARY_VERTEX_YMIN	Velocity	D_VY	0.0
BOUNDARY_VERTEX_YMIN	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_YMAX	Velocity	DF_VX _ELECTRO_OSMOSIS	HELMHOLTZ _SMOLUCHOWSKI, ZETA.POTENTIAL _INPUT.PARAM, \$(ZETA.POTENTIAL)
BOUNDARY_VERTEX_YMAX	Velocity	D_VY	0.0
BOUNDARY_VERTEX_YMAX	Velocity	D_VZ	0.0
BOUNDARY_VERTEX_XMIN	Pressure	D	0.0
BOUNDARY_VERTEX_XMAX	Pressure	D	0.0
BOUNDARY_VERTEX_XMIN	Concentration	D.CONSTANT.VALUE _FOR_A_COMPONENT	2, 0.0
BOUNDARY_VERTEX_XMIN _YUPPER	Concentration	D.CONSTANT.VALUE _FOR_A_COMPONENT	0, 1.0
BOUNDARY_VERTEX_XMIN _YLOWER	Concentration	D.CONSTANT.VALUE _FOR_A_COMPONENT	1, 1.0

- dynamics\_manager.registered\_field[]

All the fields that can be used by reaction except for “ElectricPotential” are specified.

- `dynamics_manager.procedures_table_for_initialization[].command_list[]`

Initialization procedure”INITIALIZE:REACTOR:UNDER\_ELECTRO\_OSMOSIS” is defined as follows.

field	command for initialization
Pressure	INITIALIZE:TO_ZERO
Viscosity	UPDATE:TO_CONSTANT
Velocity	INITIALIZE:TO_ZERO
Concentration	INITIALIZE:UNIFORM
Obstacle	INITIALIZE:TO_ZERO
ElectricField	SOLVE:BY_ELECTRIC_POTENTIAL:WITH_EXTERNAL_FIELD
K_Field	SOLVE:WITH_CHARGE_NEUTRALITY
Velocity	SOLVE:STOKES_EQUATION_AND_PRESSURE
Concentration	INITIALIZE:PARTIAL_REGION_CONDITION

The procedure of initialization is as follows.

- The pressure field, the velocity field, and the ion concentration field are initialized to 0.0. The viscosity field is initialized to constant-value 1.0.
  - The electric field is initialized by external electric field (parameter EXTERNAL\_ELECTRIC\_FIELD).
  - The volume force field is calculated under the charge neutrality condition. (Here, all become 0.0.)
  - The Stokes equation is solved under boundary conditions.
  - The partial region conditions are imposed on the ion concentration field, and the bulk conditions of component A and component B are given to the surface where a capillary joins.
- `dynamics_manager.procedures_table_for_evolution[].command_list[]`

Time evolution procedure”SOLVE:REACTOR:UNDER\_ELECTRO\_OSMOSIS” is defined as follows.

field	command for evolution
K_Field	SOLVE:WITH_CHARGE_NEUTRALITY
Flux	SOLVE:WITH_ELECTRIC_POTENTIAL:WITH_EXTERNAL_FIELD
Velocity	SOLVE:VELOCITY_AND_PRESSURE
Concentration	SOLVE:WITH_FLOW:WITH_REACTION
Concentration	UPDATE:PARTIAL_REGION_CONDITION

The procedure of time evolution is as follows.

- The volume force field is solved under the charge neutrality conditions.
- The diffusion flux of each component of ion under the external electric field is solved.
- Using the volume force field, the Navier-Stokes equation is solved and the time evolution of the velocity field and the pressure field is carried out.
- The time evolution of the ion concentration field is carried out the advection, diffusion, and chemical reaction equation using the diffusion flux field and the velocity field.
- The partial region conditions are imposed on the ion concentration field, and the bulk conditions in the surface where the capillaries joins are reconfigured.

## Results of simulation

The example which displays the ion concentration field (Concentration) by the view function of GOURMET is shown in Fig.2.1.2. From the figure, the ion concentration distribution of component A, component B, and component C, in time  $t = 10.0, 25.0$ , and  $50.0$  from the left figure are displayed. MeshfieldShow.py is used for the Python script for a display.

The temporal change of the average concentration of each component of ion or the velocity field on ”BOUNDARY\_VERTEX\_XMAX” surface of the down-stream side of the capillary is analyzed by a python



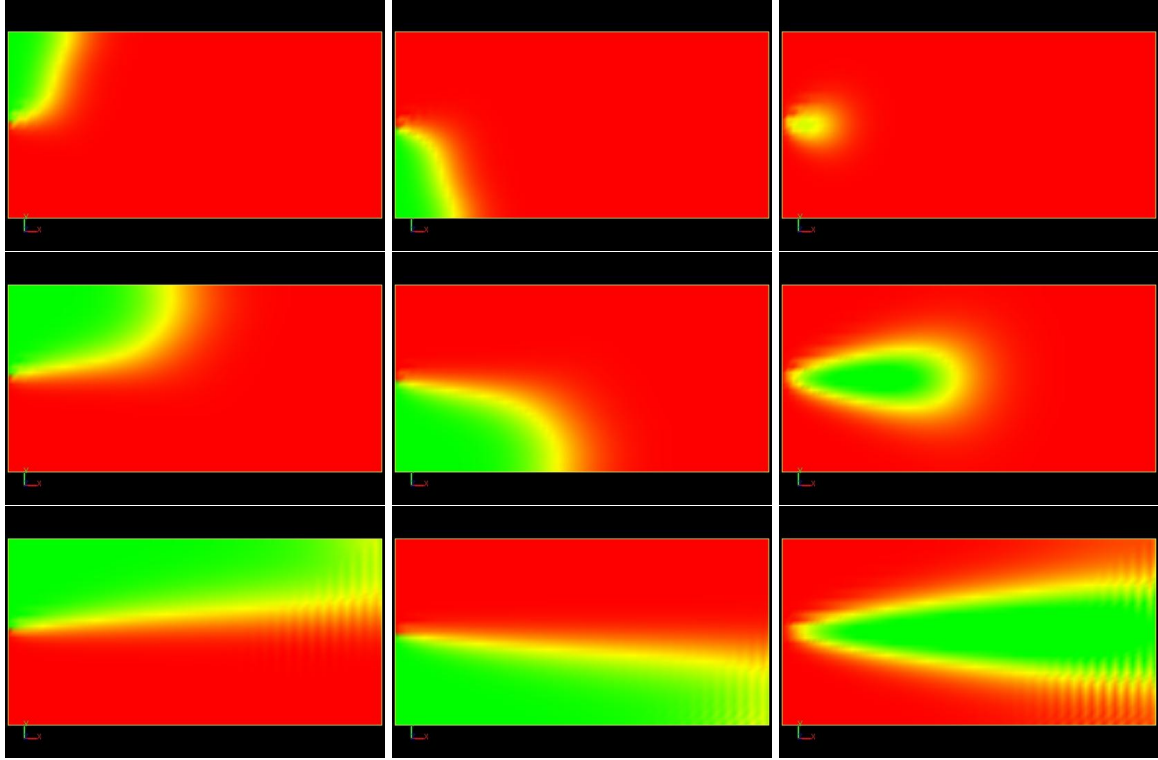


Figure 2.4: Application 2 of MEMFluid: The concentration distribution of ions by the flow injection under the electric field

program. Then, the graph of the temporal change of the reaction generation rate is shown in Fig.2.1.2. Here, the graph sheet (GraphSheet[]) is created, using `memfluid/analyze_reactor_electro_osmosis.py` as a python program for analysis. The `memfluid/plot_average_reaction_rate.gp` program is used for the gnuplot script for graphical representation.

Figure 2.5: Application 2 of MEMFluid: Temporal change of the reaction generation rate in the flow injection under the electric field



## Chapter 3

# Operation Guide of MEMFluid

### 3.1 Input Parameters of MEMFluid

#### 3.1.1 Solver control parameters of MEMFluid

Name of Parameters	Meanings and notations in theory
SEED_OF_RANDOM_NUMBER	initial random number for ion concentration field.
DT.FOR.V	time step interval for Stokes flow calculation.
MAX_ITERATION_FOR_VELOCITY_SOLVER	maximum number of iterations for Stokes flow calculation
CONVERGENCE_CRITERION_FOR_VELOCITY_SOLVER	convergence criterion for Stokes flow calculation $> 0$ : monitor relative change of velocity (default: $1.0^{-3}$ ) $< 0$ : monitor absolute change of velocity (default: $1.0^{-3}$ )
PENALTY_NUMBER	A penalty number to handle Dirichlet condition (a very large number). The default value is $10^{13}$ .
MATRIX_SOLVER_FOR_ELECTRIC_FIELD	matrix solver name to be used for electric field. Either "ICCG" or "CG" Default is "ICCG".
MATRIX_SOLVER	matrix solver name to be used in pressure solution. Either "ICCG" or "CG" Default is "ICCG".

#### 3.1.2 Physical parameters of MEMFluid

Name of Parameters	Meanings and notations in theory
NUMBER_OF_COMPONENTS	number of components
DEVIATION_FROM_AVERAGED_ION_CONCENTRATION	magnitude of noise given to initial value of ion concentration
AVERAGED_ION_CONCENTRATION	averaged ion concentration as initial value $C_{\alpha 0}$
DIFFUSION_COEFFICIENT	diffusion coefficient of each ion component $D_{\alpha}$
GRAVITY_X	X component of external force on fluid.
GRAVITY_Y	Y component of external force on fluid.
GRAVITY_Z	Z component of external force on fluid.
EXTERNAL_ELECTRIC_FIELD	uniform external electric field vector. $\mathbf{E}_0$
VALENCY	valence of each ion component $Z_{\alpha}$
Z	valence of each ion component $Z_{\alpha}$
R	ratio of electrostatic and thermal energy $R = e\Phi_0/k_B T$
CHL <sub>mn</sub>	$\chi$ -parameter of components $n, m$ (only for $m < n$ )
D	Dimensionless parameter $D$ (Coefficient of the volume force)
L	ratio of the Bjerrum length and the system size
M	Dimensionless parameter $M$

	(Number of particles in unit volume)
REYNOLDS	Reynolds number
RE	Reynolds number
DIELECTRIC_CONSTANT	relative dielectric constants for that of pure water $\epsilon_\alpha$ . Give for each components.
VISCOSITY	viscosity coefficient of each component $\eta_\alpha$
REACTION_COEFFICIENT_1_α	reaction coefficient $R1_{\alpha\beta}$ index of array is $\beta$ . ( $\beta = 0, 1, \dots, Nc - 1$ )
REACTION_COEFFICIENT_2_α	reaction coefficient $R2_{\alpha\beta\gamma}$ index of array is $Nc\beta + \gamma$ . ( $Nc\beta + \gamma = 0, 1, \dots, Nc^2 - 1$ )

## 3.2 Fields and Commands for Fields of MEMFluid

### 3.2.1 List of fields for MEMFluid

Name of Field	Meanings and notations in theory
Concentration	Ion concentration field $C_\alpha$
Flux	Flux field of ion $\dot{j}_\alpha$
K_Field	Volume force field $\mathbf{K}_\alpha$
Velocity	Velocity field $\mathbf{v}$
Pressure	Pressure field $P$
ElectricPotential	Electric potential field $\Phi$
ElectricField	Electric field $\mathbf{E}$
ChargeDensity	Electric charge density field $\rho_e$
DielectricConst	Dielectric constant field $\epsilon$
Viscosity	Viscosity field $\eta$
Obstacle	Obstacle filed (defined on FEM cells)

### 3.2.2 List of commands for MEMFluid

#### Concentration : ion concentration field - commands

Concentration	Name
Initialization	"INITIALIZE:PARTIAL_REGION_CONDITION"
Initialization	"INITIALIZE:ADD_NOISE"
Initialization	"INITIALIZE:UNIFORM"
Time evolution	"UPDATE:PARTIAL_REGION_CONDITION"
Time evolution	"SOLVE:WITH_FLOW:WITH_REACTION"
Time evolution	"SOLVE:WITHOUT_FLOW:WITH_REACTION"
Time evolution	"SOLVE:WITH_FLOW:WITHOUT_REACTION"
Time evolution	"SOLVE:WITHOUT_FLOW:WITHOUT_REACTION"
Analysis	"OUTPUT:AVS"
Evaluation	"EVALUATE:TRUE"

#### 1. Concentration - initialization commands

Name	"INITIALIZE:PARTIAL_REGION_CONDITION"
Function	Initialize field using initialization partial region conditions ("I_xxx").
Dependent parameter	NUMBER_OF_COMPONENTS

Name	"INITIALIZE:ADD_NOISE"
Function	add random noise.
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	DEVIATION_FROM_AVERAGED_CONCENTRATION
Dependent parameter	SEED_OF_RANDOM_NUMBER

Name	<b>"INITIALIZE:UNIFORM"</b>
Function	Initialize concentration of each ion component to a constant value.
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	AVERAGED_ION_CONCENTRATION

## 2. Concentration - time evolution commands

Name	<b>"UPDATE:PARTIAL_REGION_CONDITION"</b>
Function	apply partial region conditions.
Dependent parameter	NUMBER_OF_COMPONENTS

Name	<b>"SOLVE:WITH_FLOW:WITH_REACTION"</b>
Function	One step time integration of equation for ion concentration $\partial C_\alpha / \partial t = -\nabla \cdot (v C_\alpha) - \nabla \cdot \mathbf{j}_\alpha + \sum_\beta R1_{\alpha\beta} C_\beta + \sum_{\beta,\gamma} R2_{\alpha\beta\gamma} C_\beta C_\gamma$
Dependent field	Flux
Dependent field	Velocity
Dependent field	Obstacle
Dependent parameter	DT
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	DIFFUSION_COEFFICIENT

Name	<b>"SOLVE:WITHOUT_FLOW:WITH_REACTION"</b>
Function	One step time integration of equation for ion concentration $\partial C_\alpha / \partial t = -\nabla \cdot \mathbf{j}_\alpha + \sum_\beta R1_{\alpha\beta} C_\beta + \sum_{\beta,\gamma} R2_{\alpha\beta\gamma} C_\beta C_\gamma$
Dependent field	Flux
Dependent field	Obstacle
Dependent parameter	DT
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	DIFFUSION_COEFFICIENT

Name	<b>"SOLVE:WITH_FLOW:WITHOUT_REACTION"</b>
Function	One step time integration of equation for ion concentration $\partial C_\alpha / \partial t = -\nabla \cdot (v C_\alpha) - \nabla \cdot \mathbf{j}_\alpha$
Dependent field	Flux
Dependent field	Velocity
Dependent field	Obstacle
Dependent parameter	DT
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	DIFFUSION_COEFFICIENT

Name	<b>"SOLVE:WITHOUT_FLOW:WITHOUT_REACTION"</b>
Function	One step time integration of equation for ion concentration $\partial C_\alpha / \partial t = -\nabla \cdot \mathbf{j}_\alpha$
Dependent field	Flux
Dependent field	Obstacle
Dependent parameter	DT
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	DIFFUSION_COEFFICIENT

## 3. Concentration - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
I.CONSTANT.VALUE.FOR.A.COMPONENT	initialize $\psi_\alpha$ to a constant value. Give component index $\alpha$ and $\psi_\alpha$ .
D.CONSTANT.VALUE.FOR.A.COMPONENT	set $\psi_\alpha$ to a constant value (Dirichlet condition). Give component index $\alpha$ and $\psi_\alpha$ .

## 4. Concentration - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

## 5. Concentration - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

## Flux : flux field of ion, K\_Field : volume force field - commands

K_Field	Name
Initialization	"INITIALIZE:TO_ZERO"
Initialization	"INITIALIZE:TO_CONSTANT_FORCE"
Time evolution	"UPDATE:TO_ZERO"
Time evolution	"SOLVE:WITH_ELECTRIC_POTENTIAL"
Time evolution	"SOLVE:WITH_ELECTRIC_POTENTIAL:WITH_EXTERNAL_FIELD"
Time evolution	"SOLVE:WITH_CHARGE_NEUTRALITY"
Time evolution	"UPDATE:PARTIAL_REGION_CONDITION"
Analysis	"OUTPUT:AVS"
Evaluation	"EVALUATE:TRUE"

## 1. K\_Field - initialization commands

Name	<b>"INITIALIZE:TO_ZERO"</b>
Function	set field values to zero.
Dependent parameter	NUMBER_OF_COMPONENTS

Name	<b>"INITIALIZE:TO_CONSTANT_FORCE"</b>
Function	apply a constant external force.
Dependent parameter	GRAVITY_X
Dependent parameter	GRAVITY_Y
Dependent parameter	GRAVITY_Z

## 2. K\_Field - time evolution commands

Name	<b>"UPDATE:TO_ZERO"</b>
Function	set field values to zero.
Dependent parameter	NUMBER_OF_COMPONENTS

Name	<b>"SOLVE:WITH_ELECTRIC_POTENTIAL"</b>
Function	calculate flux for each ion component. $\mathbf{j}_\alpha = \nabla C_\alpha + \sum_\beta \chi_{\alpha\beta} C_\alpha \nabla C_\beta + R Z_\alpha C_\alpha \nabla \Phi$
Dependent field	Concentration
Dependent field	ElectricPotential
Dependent field	Obstacle
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	Z (or VALENCY)
Dependent parameter	R
Dependent parameter	CHI <sub>mn</sub>

Name	<b>"SOLVE:WITH_ELECTRIC_POTENTIAL :WITH_EXTERNAL_FIELD"</b>
Function	calculate flux for each ion component. $\mathbf{j}_\alpha = \nabla C_\alpha + \sum_\beta \chi_{\alpha\beta} C_\alpha \nabla C_\beta + RZ_\alpha C_\alpha (\nabla \Phi - \mathbf{E}_0)$
Dependent field	Concentration
Dependent field	ElectricPotential
Dependent field	Obstacle
Dependent parameter	EXTERNAL_ELECTRIC_FIELD
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	Z (or VALENCY)
Dependent parameter	R
Dependent parameter	CHI <sub>mn</sub>

Name	<b>"SOLVE:WITH_CHARGE_NEUTRALITY"</b>
Function	calculate flux for each ion component. $\mathbf{j}_\alpha = \nabla C_\alpha + \sum_\beta \chi_{\alpha\beta} C_\alpha \nabla C_\beta$
Dependent field	Concentration
Dependent field	Obstacle
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	CHI <sub>mn</sub>

Name	<b>"UPDATE:PARTIAL_REGION_CONDITION"</b>
Function	apply partial region conditions.
Dependent parameter	NUMBER_OF_COMPONENTS

### 3. K\_Field - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
D_CONSTANT_VALUE_FOR_A_COMPONENT	set flux of a component to a constant value. (Dirichlet condition) give component index $\alpha$ , $\mathbf{j}_{\alpha x}$ , $\mathbf{j}_{\alpha y}$ and $\mathbf{j}_{\alpha z}$ .

### 4. K\_Field - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

### 5. K\_Field - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

## Velocity : velocity field - commands

Velocity	Name
Initialization	"INITIALIZE:TO_ZERO"
Initialization	"INITIALIZE:DIRICHLET_CONDITION"
Initialization	"INITIALIZE:PARTIAL_REGION_CONDITION"
Time evolution	"SOLVE:VELOCITY_AND_PRESSURE"
Time evolution	"SOLVE:STOKES_EQUATION_AND_PRESSURE"
Time evolution	"UPDATE:DIRICHLET_CONDITION"
Analysis	"OUTPUT:AVS"
Evaluation	"EVALUATE:TRUE"



## 1. Velocity - initialization commands

Name	<b>"INITIALIZE:TO_ZERO"</b>
Function	set field values to zero.

Name	<b>"INITIALIZE:DIRICHLET_CONDITION"</b>
Function	initialize using Dirichlet boundary conditions in partial region conditions.
Dependent field	ElectricField
Dependent field	ChargeDensity
Dependent field	Concentration
Dependent parameter	D
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	Z or VALENCY
Dependent parameter	NUMBER_OF_COMPONENTS

Name	<b>"INITIALIZE:PARTIAL_REGION_CONDITION"</b>
Function	initialize using initial conditions in partial region conditions.
Dependent field	ElectricField
Dependent field	ChargeDensity
Dependent field	Concentration
Dependent parameter	D
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	Z or VALENCY
Dependent parameter	NUMBER_OF_COMPONENTS

## 2. Velocity - time evolution commands

Name	<b>"SOLVE:VELOCITY_AND_PRESSURE"</b>
Function	One time step evolution of velocity by Navier Stokes equation, $Re\rho\partial\mathbf{v}/\partial t = -\nabla p + \nabla(\eta_w\{\nabla\mathbf{v} + (\nabla\mathbf{v})^t\}) + MD\mathbf{K}$
Dependent field	Pressure
Dependent field	ElectricField
Dependent field	ChargeDensity
Dependent field	Concentration
Dependent field	K_Field
Dependent field	Viscosity
Dependent parameter	DT
Dependent parameter	RE or REYNOLDS
Dependent parameter	D
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	Z or VALENCY
Dependent parameter	NUMBER_OF_COMPONENTS

Name	<b>"SOLVE:STOKES_EQUATION_AND_PRESSURE"</b>
Function	Solve Stokes flow equation for velocity and pressure. $-\nabla p + \nabla(\eta_w\{\nabla \mathbf{v} + (\nabla \mathbf{v})^t\}) + MD\mathbf{K} = 0$
Dependent field	Pressure
Dependent field	ElectricField
Dependent field	ChargeDensity
Dependent field	Concentration
Dependent field	K_Field
Dependent field	Viscosity
Dependent parameter	DT
Dependent parameter	DT_FOR_V
Dependent parameter	MAX_ITERATION_FOR_VELOCITY_SOLVER
Dependent parameter	CONVERGENCE_CRITERION_FOR_VELOCITY_SOLVER
Dependent parameter	D
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	VALENCY
Dependent parameter	Z

Name	<b>"UPDATE:DIRICHLET_CONDITION"</b>
Function	Apply Dirichlet boundary conditions in partial region conditions. Using command "SOLVE:VELOCITY_AND_PRESSURE" or "SOLVE:STOKES_EQUATION_AND_PRESSURE", partial region conditions are applied automatically, so you may not need to use this command explicitly.
Dependent field	Velocity
Dependent field	ElectricField
Dependent field	ChargeDensity
Dependent field	Concentration
Dependent parameter	D
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	Z or VALENCY

### 3. Velocity - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
I_VX I_VY I_VZ I_VEC	initialize $v_x$ to a constant value initialize $v_y$ to a constant value initialize $v_z$ to a constant value initialize $\mathbf{v}$ to a constant value
D_VX D_VY D_VZ D_VEC	set $v_x$ to a constant value (Dirichlet condition) set $v_y$ to a constant value (Dirichlet condition) set $v_z$ to a constant value (Dirichlet condition) set $\mathbf{v}$ to a constant value (Dirichlet condition)
IF_VX_ELECTRO_OSMOSIS IF_VY_ELECTRO_OSMOSIS IF_VZ_ELECTRO_OSMOSIS IF_VEC_ELECTRO_OSMOSIS	initialize $v_x$ by electro-osmosis initialize $v_y$ by electro-osmosis initialize $v_z$ by electro-osmosis initialize $\mathbf{v}$ by electro-osmosis
DF_VX_ELECTRO_OSMOSIS DF_VY_ELECTRO_OSMOSIS DF_VZ_ELECTRO_OSMOSIS DF_VEC_ELECTRO_OSMOSIS	set $v_x$ by electro-osmosis set $v_y$ by electro-osmosis set $v_z$ by electro-osmosis set $\mathbf{v}$ by electro-osmosis
Parameters of ELECTRO_OSMOSIS	
value[0]	Method to calculate the velocity of electro-osmosis HELMHOLTZ.SMOLUCHOWSKI or MUFFIN_ELECTROLYTE
value[1]	Method to calculate $\zeta$ -potential (only when, HELMHOLTZ.SMOLUCHOWSKI) ZETA_POTENTIAL_INPUT_PARAM or ZETA_POTENTIAL_POISSON_BOLTZMANN or ZETA_POTENTIAL_LINEAR_CONDENSOR or ZETA_POTENTIAL_MUFFIN_ELECTROLYTE
value[2]	Value of $\zeta$ -potential (only when, ZETA_POTENTIAL_INPUT_PARAM)

#### 4. Velocity - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

#### 5. Velocity - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

### Pressure : pressure field - commands

Pressure	Name
Initialization	<b>"INITIALIZE:TO_ZERO"</b>
Time evolution	<b>"SOLVE:WITH_VELOCITY"</b>
Analysis	<b>"OUTPUT:AVS"</b>
Evaluation	<b>"EVALUATE:TRUE"</b>

#### 1. Pressure - initialization commands

Name	<b>"INITIALIZE:TO_ZERO"</b>
Function	set field values to zero.

## 2. Pressure - time evolution commands

Name	<b>"SOLVE:WITH_VELOCITY"</b>
Function	Solve Poisson equation for pressure $\Delta p = 1/\Delta t \nabla \cdot \mathbf{v}^*$
Dependent field	Pressure
Dependent field	Velocity
Dependent parameter	DT
Dependent parameter	DT_FOR_V
Dependent parameter	PENALTY_NUMBER
Dependent parameter	MATRIX_SOLVER

## 3. Pressure - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
D or D_CONSTANT_VALUE	set to a constant value (Dirichlet condition)
N	$\mathbf{n} \cdot \nabla p = \bar{p}_n$ (Neumann condition)

## 4. Pressure - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

## 5. Pressure - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

**ElectricPotential : electric potential field - commands**

ElectricPotential	Name
Time evolution	<b>"SOLVE:POISSON"</b>
Time evolution	<b>"SOLVE:POISSON:WITH_CHARGE_NEUTRALITY"</b>
Analysis	<b>"OUTPUT:AVS"</b>
Evaluation	<b>"EVALUATE:TRUE"</b>

## 1. ElectricPotential - time evolution commands

Name	<b>"SOLVE:POISSON"</b>
Function	Solve Poisson equation for electric potential $\Delta \Phi = -(M/R)L \sum_{\alpha} Z_{\alpha} C_{\alpha}$
Dependent field	DielectricConst
Dependent field	ChargeDensity
Dependent field	ElectricPotential
Dependent field	Concentration
Dependent parameter	PENALTY_NUMBER
Dependent parameter	MATRIX_SOLVER_FOR_ELECTRIC_FIELD
Dependent parameter	MATRIX_SOLVER
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	Z or VALENCY
Dependent parameter	NUMBER_OF_COMPONENTS

Name	<b>"SOLVE:POISSON:WITH_CHARGE_NEUTRALITY"</b>
Function	Solve Laplace equation for electric potential with the local charge neutrality. $\Delta\Phi = 0$
Dependent field	DielectricConst
Dependent field	ElectricPotential
Dependent field	ChargeDensity
Dependent field	Concentration
Dependent parameter	PENALTY_NUMBER
Dependent parameter	MATRIX_SOLVER_FOR_ELECTRIC_FIELD
Dependent parameter	MATRIX_SOLVER
Dependent parameter	R
Dependent parameter	L
Dependent parameter	M
Dependent parameter	Z or VALENCY
Dependent parameter	NUMBER_OF_COMPONENTS

## 2. ElectricPotential - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
D	set to a constant value (Dirichlet condition)
DF_ZETA_POTENTIAL	set a calculated value by electro-osmosis
Parameters of ZETA_POTENTIAL	
value[0]	Electric potential applied on outer boundary.
value[1]	Method to calculate $\zeta$ -potential INPUT_PARAM or POISSON_BOLTZMANN or LINEAR_CONDENSOR or MUFFIN_ELECTROLYTE
value[2]	Value of $\zeta$ -potential (only when, INPUT_PARAM)

## 3. ElectricPotential - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

## 4. ElectricPotential - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

## ElectricField : electric field - commands

ElectricField	Name
Time evolution	<b>"SOLVE:BY_ELECTRIC_POTENTIAL"</b>
Time evolution	<b>"SOLVE:BY_ELECTRIC_POTENTIAL:WITH_EXTERNAL_FIELD"</b>

### 1. ElectricField - time evolution commands

Name	<b>"SOLVE:BY_ELECTRIC_POTENTIAL"</b>
Function	Solve electric field by the gradient of electric potential $\mathbf{E} = -\nabla\Phi$
Dependent field	ElectricPotential

Name	<b>"SOLVE:BY ELECTRIC POTENTIAL :WITH EXTERNAL FIELD"</b>
Function	Solve electric field by the gradient of electric potential and external electric field. $\mathbf{E} = -\nabla\Phi + \mathbf{E}_0$
Dependent field	ElectricPotential
Dependent parameter	EXTERNAL_ELECTRIC_FIELD

### ChargeDensity : charge density field - commands

ChargeDensity	Name
Initialization	"INITIALIZE:PARTIAL_REGION_CONDITION"
Initialization	"INITIALIZE:UNIFORM"
Time evolution	"UPDATE:TO_ZERO"
Time evolution	"SOLVE:BY_ION_CONCENTRATION"
Time evolution	"UPDATE:PARTIAL_REGION_CONDITION"
Analysis	"OUTPUT:AVS"
Evaluation	"EVALUATE:TRUE"

#### 1. ChargeDensity - initialization commands

Name	<b>"INITIALIZE:PARTIAL_REGION_CONDITION"</b>
Function	Initialize field using initialization partial region conditions ("I_xxx").
Name	<b>"INITIALIZE:UNIFORM"</b>
Function	Initialize by a uniform constant charge density.
Dependent parameter	UNIFORM_CHARGE_DENSITY

#### 2. ChargeDensity - time evolution commands

Name	<b>"UPDATE:TO_ZERO"</b>
Function	set field values to zero.
Name	<b>"SOLVE:BY_ION_CONCENTRATION"</b>
Function	calculate charge density from ion concentration, $\rho_e = \sum_{\alpha} Z_{\alpha} C_{\alpha}$
Dependent field	Concentration
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	Z or VALENCY
Name	<b>"UPDATE:PARTIAL_REGION_CONDITION"</b>
Function	apply partial region conditions.

#### 3. ChargeDensity - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
I or I_CONSTANT_VALUE	initialize to a constant value.
D or D_CONSTANT_VALUE	set to a constant value (Dirichlet condition)

#### 4. ChargeDensity - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

#### 5. ChargeDensity - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

**DielectricConst : dielectric constant field - commands**

DielectricConst	Name
Time evolution	"UPDATE:TO_CONSTANT"
Time evolution	"SOLVE:BY_CONCENTRATION"
Time evolution	"UPDATE:PARTIAL_REGION_CONDITION"
Analysis	"OUTPUT:AVS"
Evaluation	"EVALUATE:TRUE"

**1. DielectricConst - time evolution commands**

Name	"UPDATE:TO_CONSTANT"
Function	set uniform dielectric constant with $\epsilon_0$ (component 0).
Dependent parameter	DIELECTRIC_CONSTANT
Name	"SOLVE:BY_CONCENTRATION"
Function	calculate dielectric constant from that of each component weighting by ion concentration $\epsilon = \sum_{\alpha} \epsilon_{\alpha} C_{\alpha}$
Dependent field	Concentration
Dependent parameter	NUMBER_OF_COMPONENTS
Dependent parameter	DIELECTRIC_CONSTANT
Name	"UPDATE:PARTIAL_REGION_CONDITION"
Function	apply partial region conditions.

**2. DielectricConst - partial region condition (boundary condition) commands**

Partial region condition	meanings and parameters
D or D.CONSTANT_VALUE	set to a constant value (Dirichlet condition)

**3. DielectricConst - analysis commands**

Name	"OUTPUT:AVS"
Function	output calculation results on an AVS format file(ucd-data).

**4. DielectricConst - evaluation commands**

Name	"EVALUATE:TRUE"
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

**Viscosity : viscosity field - commands**

Viscosity	Name
Time evolution	"UPDATE:TO_CONSTANT"
Time evolution	"UPDATE:PARTIAL_REGION_CONDITION"
Analysis	"OUTPUT:AVS"
Evaluation	"EVALUATE:TRUE"

**1. Viscosity - time evolution commands**

Name	"UPDATE:TO_CONSTANT"
Function	set uniform viscosity coefficient with $\eta_0$ (component 0).
Dependent parameter	VISCOSITY
Name	"UPDATE:PARTIAL_REGION_CONDITION"
Function	apply partial region conditions.

## 2. Viscosity - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
D or D_CONSTANT_VALUE	set to a constant value (Dirichlet condition)

## 3. Viscosity - analysis commands

Name	<b>"OUTPUT:AVS"</b>
Function	output calculation results on an AVS format file(ucd-data).

## 4. Viscosity - evaluation commands

Name	<b>"EVALUATE:TRUE"</b>
Function	Always return "true" flag. This function is used to perform an analysis command with a constant time step interval.

## Obstacle : obstacle field - commands

Obstacle	Name
Initialization	<b>"INITIALIZE:TO_ZERO"</b>
Time evolution	<b>"UPDATE:PARTIAL_REGION_CONDITION"</b>
Evaluation	<b>"EVALUATE:TRUE"</b>

## 1. Obstacle - initialization commands

Name	<b>"INITIALIZE:TO_ZERO"</b>
Function	set field values to zero.

## 2. Obstacle - time evolution commands

Name	<b>"UPDATE:PARTIAL_REGION_CONDITION"</b>
Function	apply partial region conditions.

## 3. Obstacle - partial region condition (boundary condition) commands

Partial region condition	meanings and parameters
D or D_CONSTANT_VALUE	set to a constant value (Dirichlet condition)