Progress in Nuclear Science and Technology
Volume 8 (2025) pp. 209-215

DOI: 10.15669/pnst.8.209

ARTICLE

A Novel Computational Framework for Extraction Chromatography Calculation: ANSYS Fluent Coupled with Python

Masahiko NAKASE [1*], Sou WATANABE [2], Yuichi SANO [2] and Masayuki TAKEUCHI [2]

[1] *Institute of Science Tokyo, 2-12-1, Ookayama, Meguro-ku, Tokyo 152-8550, Japan*
[2] *Japan Atomic Energy Agency, 2-4 Shirakata, Tokai-mura, Naka-gun, Ibaraki-ken, 319-1195, Japan*

Extraction chromatography is practical for small-scale, high-level separations. Numerical simulation can help find an appropriate operational condition when scaling up and designing a separation flowsheet. The simulation can also be applied to safety analysis, such as the transient situation of the flow stop. However, parameter fitting requires iterative calculation attempts and many varied parameters sometimes cause errors and mistakes. Therefore, we have developed an environment that enables efficient and iterative calculations by connecting ANSYS Fluent and Python using the PyFluent module on Jupyter Notebook. This development promises a more efficient future for the technology, making the audience feel optimistic about its potential.

KEYWORDS: Chromatographic Calculation, ANSYS Fluent, ANSYS Workbench, PyFluent

I. Introduction

Extraction chromatography is a reliable method for separating substances that are difficult to separate on a small scale. A numerical simulation is essential for scale-up and application studies. The linear driving force (LDF) model is the most frequently used and reliable. However, some parameters in the calculation, such as the effective mass transfer coefficient and other input conditions, require adjustment. Repeating many numerical simulations while managing extensive data is essential in this case. Still, managing input and result data tends to be complicated, and mistakes can occur in calculation settings. To solve this problem, we focused on the Python module for ANSYS Fluent, PyFluent, which offers many convenient modules for manipulating Fluent codes.[1,2] We established the efficient calculation environment based on PyFluent, and this paper outlines the scheme.

II. Code Description

1. Code Description

Figure 1 illustrates the schematic of the calculation using PyFluent. Jupyter Notebook, an open-source web application, plays a crucial role in our process, allowing us to write, execute, and share Python code interactively.[3] The Python code can be run in separate cells on Jupyter Notebook, which will enable us to test and debug the code in parts and see the results immediately. We can also use Markdown and LaTeX to describe explanations and mathematical equations, thereby enhancing our understanding of the code. Additionally, we can visualize data using libraries such as Matplotlib and Pandas. Namely, many Python codes can be efficiently combined with Fluent calculation. First, launch the Jupyter Notebook and open the prepared IPython Notebook file, which contains the necessary information for calculation. Launch the required Python modules, including PyFluent, and define the parameters. Fluent fluid flow and thermal conductivity calculations are performed using the prepared function, and adsorption/desorption is handled through a User-Defined Function (UDF) with a User-Defined Scalar (UDS), which enables customization and extension of the software's capabilities. UDFs are custom functions that we can write in the C language to enhance the functionality of ANSYS Fluent. These functions allow us to define our original models, boundary conditions, source terms, and properties. UDFs enable us to tailor Fluent's functionality to meet specific simulation needs. UDS is the scalar variable users can define and solve in a Fluent simulation. These scalars are often used to model additional physical phenomena not covered by the standard equations provided by Fluent. UDS allows us to add supplemental transport equations to the simulation. After

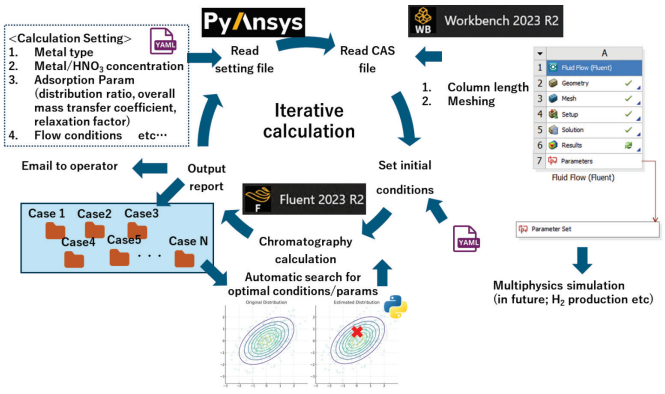

Fig. 1 Calculation extraction chromatography using PyFluent environment.

*Corresponding author, E-mail: nakase.m.878c@m.isct.ac.jp

launching Fluent, compilation and hooking scheme of of pre-prepared UDF, loading of pre-prepared CAS file which has various settings (e.g., setting of UDF source functions to adsorption layer, setting of boundary conditions), report definition and monitor settings, automatic save settings, initialization (e.g., initialization of initialization), run the calculation, save the data, and close Fluent. Thanks to the scheme, the thermodynamic parameters can be systematically changed using a loop in Python code. It is also combined with some modules to handle the calculation life cycle.

As shown in **Fig. 2,** ANSYS SpaceClaim creates the geometries, and ANSYS Meshing performs meshing sequentially within the ANSYS Workbench workflow, followed by the given strategy,[6-8] Then, the boundary conditions are set, including the inlet and outlet of solutions and metal ions, as well as the temperatures of the solution and the solid region (adsorption region, outer frame, and inlet filter). Based on the prepared meshing strategy, the mesh is created automatically after modifying the geometries if they are changed. By naming the geometric parts in advance, such as inlet, outlet, fluid region, and solid region, the setup automatically proceeds if the geometry changes due to connections in the workflow within ANSYS Workbench, which enables efficient parametric simulation with changing geometry. The setting of conditions, including the thermodynamic parameters of adsorption, is compiled and loaded by the Jupyter Notebook rather than using a GUI, as shown in **Fig. 3**.

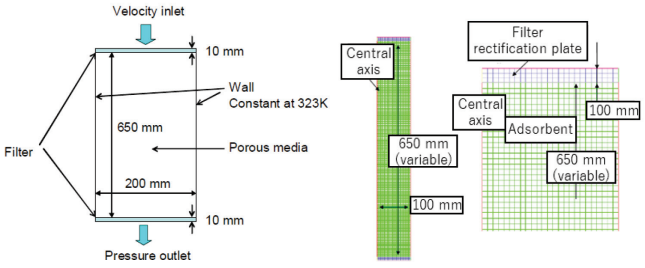

Fig. 2 The simple geometry of the column and the generated mesh are used for calculation.

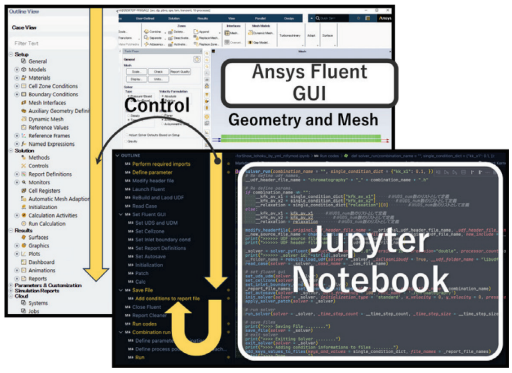

Fig. 3 Utilization of Jupyter Notebook to run ANSYS software.

There are many parameters, especially when handling multiple variables to fit experimental and calculated data, which can lead to errors in the setting. Therefore, many parameters are set in the YAML file and override Fluent's CAS file. The YAML is a human-readable data serialization standard that can represent complex data structures in a simple and readable format. The loop can systematically change the parameters to vary fitting parameters. When a calculation is enabled, the parameters related to adsorption are adjusted according to the settings within the loop, the input data for Fluent is generated, and the calculation proceeds. The output data, including the concentration of UDS (nitric acid and metal ions) from the outlet after every specific time step, are compiled and stored in the designated directory.

2. Calculation Description

The adsorption model used in this study is a simple LDF model as follows,

$$\frac{\partial C_k}{\partial t} = -u_z \frac{\partial C_k}{\partial z} - \frac{K_{fk} a_v}{\varepsilon_h}(C_k - C_k^*) \quad (1)$$

$$(1-\varepsilon_b)K_k \frac{\partial C_k^*}{\partial t} = K_{fk} a_v (C_k - C_k^*) \quad (2),$$

where C_k: concentration of ions in each cell (kg/m^3), t: time (s), u_z: axial flow (m/s), K_{fk}: Overall mass transfer coefficient [m/s] (function of nitric acid solution concentration), a_v: surface area (surface area of adsorbent per unit volume) (1/m), ε_b: porosity = 0.37, C_{k*}: Concentration of ions on the fluid side at equilibrium (kg/m^3), and K_k: distribution coefficient (-) (function of the concentration of aqueous nitric acid solution). The model calculates C_k and C_{k*} from the above two equations at a particular time, obtained at a position ∂z classified in the flow direction. Using the obtained C_k and C_{k*}, the mass transfer rate, $\dot{m}$(kg/s) (adsorption volume) in each cell (microvolume V_c = ∂dz $\times$ area S) is calculated as follows;

$$\dot{m} = V_c K_{fk} a_v (C_k - C_k^*) \quad (3)$$

The model is one-dimensional but two-dimensional (axisymmetric), with the radial direction (y direction) added to the advection term. Therefore, Eq. (4) is calculated at every iteration (UDS transport equation in Fluent), and C_k is calculated (in this calculation, C_{k*} is calculated as a constant).

$$\frac{\partial \rho C_k}{\partial t} + \nabla \cdot (\rho u C_k) = \frac{K_{fk} a_v}{\varepsilon_h}(C_k - C_k^*)\rho \quad (4)$$

The right-hand side of the Eq. (4) is the generating term. When the adsorption amount reaches saturation in the cell, it is assumed that no adsorption (zero production) occurs. Eq. (2) is calculated for each iteration to obtain the equilibrium concentration, and this calculation is performed in the UDF (DEFINE_ADJUST), where Ck is a constant in this calculation. Equation (2) is described as the differential Eq. (5) and can be converted to Eq. (6), which is then solved at every iteration.

$$\frac{C_k^{*n+1} - C_k^{*n}}{\Delta t} = \frac{K_{fk} a_v}{(1-\varepsilon_h)K_k}(C_k^{n+1} - C_k^{*n+1}) \quad (5)$$

$$C_k^{*n+1} = \frac{(1-\varepsilon_b)K_k}{\Delta t K_{fk} a_v + (1-\varepsilon_b)K_k} C_k^{*n} + \frac{\Delta t K_{fk} a_v}{\Delta t K_{fk} a_v + (1-\varepsilon_b)K_k} C_k^{n+1} \quad (6),$$

where the subscript n represents the current time, and n+1 represents the time one step forward. The amount of adsorption (kg/m^3) in each cell is stored in the last iteration

of each time step. **Table 1** summarizes the coefficients used in this manuscript assuming four metal ions. Heat generation and the irradiation of α- and β-rays are considered, and energy sources are added as Eq. (7), a function of ion concentration, as follows.

$$S=A_{rA}(C_{kA}+C_{ksA})+A_{rB}(C_{kB}+C_{ksB})+A_{rC}(C_{kC}+C_{ksC})+A_{rD}(C_{kD}+C_{ksD})+B_{rA}(C_{kA}+C_{ksA})+B_{rB}(C_{kB}+C_{ksB})+B_{rC}(C_{kC}+C_{ksC})+B_{rD}(C_{kD}+C_{ksD})\ [W/m^3] \quad (7)$$

where C_{ki} is the concentration of each ion in solution (kg/m³), C_{ksi} is the adsorbed concentration of each ion (kg/m³), and Ar and Br are the calorific values associated with α- and β-rays, respectively (W/kg). **Table 1** summarizes the coefficients used in this manuscript assuming four metal ions. The $K_{fk}\times a_v$ is a parameter related to kinetics, given by the function of nitric acid and a correlation equation derived from experiments.

Table 1 Parameters for each element A to D

Element	A	B	C	D
Molecular weight (kg/mol)	0.08762	0.09122	0.243	0.144
$Q_{o,\,max}$ (mol/m³)	0	12.5	130	130
$K_{fk}\times a_v$ (1/s)	0	0.0265	0.1219	0.1186
y; nitric acid conc.(mol/L)		- 0.00228y	-0.01668y	-0.0221y
$K_{fk}\times a_v$ (1/s)	0	6.6067	7.157	13.13
y; nitric acid conc.(mol/L)		+ 7322y	+ 19.54y	+ 29.52y
a (m³/mol)	0	0.04323	0.1049	0.01051
A_r (W/kg)	0	0	2946	0
B_r (W/kg)	0	1.35×10^{-3}	0	0

The saturation amount of each ion, Q_{sum_i} (mol/m³), is specified as the adsorption saturation amount of each ion in Eqs. (8) and (9).

$$Q_{sum\_i} = \frac{Q_{maxi} a_i c_i^m}{1+\sum_j a_j c_j^m} \quad (8)$$

$$Q_{maxi} = Q_{0maxi} \times (-0.3 \times \log_{10}[y] + 0.7) : y > 0.005$$
$$\text{where } Q_{maxi} = 0.0 : y < 0.005 \quad (9),$$

where i, j: indicate each ion A, B, C, D, ai: coefficient of saturation of element i (m³/mol), Q_{maxi}: coefficient related to saturation (m³/mol), y: concentration of nitric acid (mol/L), Cm: molar concentration of each ion in solution (mol/m³). Adsorption is assumed to be an independent reaction for each ionic component, and the maximum adsorption amount and the total amount limit were set to simplify the system. If only one ion is present in a cell and no other ions are present, the saturation adsorption capacity of the ions is set as Q_{maxi} for each ion rather than Q_{sum_i}. If the amount of a particular component is 10,000 times greater than the sum of the amounts of the other components, we regard it as if that component were the only one present in the solution (Eq. (10)). For example, for ion B, Q_{maxB} is used as the threshold for saturation instead of Q_{max_all} if it is equal to or less than Q_{maxB}.

$$\frac{C_A+C_C+C_D}{C_B} < 0.0001 \quad (10)$$

where C is the concentration of each ion in solution (kg/m³) (UDS) in the 4-ion system, A, B, C, and D, coexisting in solution. The saturation adsorption amount is calculated and updated for each iteration.

3. Setting of a Fluent Solver

The transient calculation of fluid flow, including thermal conductivity and species transport, is implemented using constant time steps that depend on the mesh size and the phenomena being modeled. Gravity was not considered, and the operational pressure was set at 101,325 Pa. The transport calculation of chemical species was implemented using a UDS (values are in (kg/m³)). A laminar flow model is used since the operational volume rate is small. A mixture of nitric acid solution with or without mixed ions and pure water is flown from the inlet. These species are calculated using the Fluent chemical species transport model. The definition of the mixed fluid (nitric acid and water) is specified individually, and the mixing law is defined). Density, specific heat, viscosity, thermal conductivity, volume-weighted mixing law, mass-weighted mixing law, and mass-weighted mixing law were used, respectively. As the diffusion coefficient values are not known, 1.74×10^{-9} (m²/s) was used as a reasonable estimate for both. The molecular weight of each component is used in the mixing-law calculation of specific heat; therefore, the molecular weight of the water and nitric acid solution is the same value. As the property of nitric acid, the pure nitric acid and water were mixed in a simple weighted average of 20 wt%. The physicochemical properties of water, noted as 'Water-liquid' in Fluent, are set as follows: density: 998.2 kg/m³, specific heat: 4182 J/kg·K, heat transfer coefficient: 0.6 W/m·K, viscosity: 0.001003 kg/m·s, and molar weight: 18.0152 kg/mol. The physicochemical properties of nitric acid are as follows: 100% nitric acid, with a density of 1.504 kg/m³ and a specific heat capacity. Equations (11) and (12) illustrate the model for Cp (J/kg·K) as a function of temperature.

$$\text{300-1000 (K): } Cp = 178.5483 + 2.929248\ T - 0.002610976\ T^2 + 1.157688 \times 10^{-6}\ T^3 - 2.188183 \times 10^{-10}\ T^4 \quad (11)$$

$$\text{1000-5000 (K): } Cp = 924.134 + 0.7668073\ T - 0.0003079356\ T^2 + 5.658948 \times 10^{-8}\ T^3 - 3.904811 \times 10^{-12}\ T^4 \quad (12)$$

Heat transfer coefficient: 0.0454 (W/m·K), viscosity: 0.0011 (kg/m·s), molar mass: 63.01287 (kg/kmol). The mixing rule specifies such physicochemical properties. The mixing law determines how the physical properties of an aqueous solution formed by mixing water and nitric acid are calculated based on their mass fractions in a chemical species analysis. For example, suppose the thermal conductivity is entered as a constant in the mixing-template. In that case, the thermal conductivity of the aqueous solution will be treated as constant regardless of the fractions of water and nitric acid. If the change in value is small, irrespective of the fraction (concentration), it is adequate to simplify the calculation by treating it as a constant. The physical properties of “Water Liquid” and “Nitric Acid” were set by following properties of aqueous nitric acid solution: density of 20 wt% nitric acid solution at 25°C is obtained from Spencer's Eqs. (13) and (14), which is an experimental approximation and a quadratic function of

temperature (polynomial).

$$\frac{1}{\rho_{25}} = 1.0003124 - 3.364529 \times 10^{-2}\, c_{H,25} + 1.219254 \times 10^{-3}\, c_{H,25}^2 - 1.681297 \times 10^{-3}\, c_{H,25}^3 \quad (13)$$

$$\frac{1}{\rho_\theta} = \frac{1}{\rho_{25}}\{(1 + 1.647365 \times 10^{-3} + 1.897063 \times 10^{-3}\, \rho_{25} + 2.017796 \times 10^{-6}\, \theta)(\theta - 25)\} \quad (14),$$

where ρ_θ (g/cm^3) is density at temperature θ°C, $C_{H,25}$ (mol/l) : molar concentration of nitric acid at 25 °C. In Eq. (14), the regression gave ρ25 = 0.999794402 (g/cm³). The volume-weighted mixing law of density calculates the density, ρ of an aqueous solution using the following equation.

$$\rho = \frac{1}{\sum_i \frac{Y_i}{\rho_i}} \quad (15)$$

where Yi is the mass fraction of chemical species i and ρi is the density of chemical species i. The mass diffusion coefficient (m²/s) can be considered a coefficient that describes the diffusion of a chemical species, such as nitric acid and metal ions, in an aqueous solution. UDS (metallic element) diffusivities are also considered, but not in this study. The density of water-liquid prepared in Fluent is 998.2 kg/m³, and the density of 100% nitric acid is 1,504 kg/m³. The difference between the calculated example of mixing density in FLUENT and the actual density of aqueous nitric acid solution is shown for reference. The individual temperature dependence of water alone or 100% nitric acid can be defined, but the mass fraction can only be considered in Eq. (15). The temperature dependence is the viscosity coefficient, as shown in **Table 2**. The data in the table below is determined by linear interpolation, and the values outside the temperature range are determined by zero-order interpolation.

Table 2 Temperature dependency of viscosity

Params	Units	Values			
Temperature	(K)	283.15	293.15	300.15	310.15
Viscosity	(10^{-3}Pa・s)	1.32	1.05	0.86	0.71

The specific heat at constant pressure of a 20 wt% nitric acid solution at 20°C is 3,390 J/kg·K. The following first-order equation considers the temperature dependence of the specific heat.

$$Cp_\theta = 3390 + 2.0 \cdot (\theta - 20)\ \theta\ Cp \quad (16),$$

where Cp_θ (J/kg·K) is the specific heat at constant pressure at θ°C. Thermal conductivity is assumed to be independent of temperature, and the thermal conductivity of a 20 wt% nitric acid solution at 20°C, 0.546 W/m·K, is used as a constant value. The temperature dependence of physical properties is taken from the Thermophysical Properties of Fluids (The Japan Society of Mechanical Engineers[9]), as listed in **Table 3**. Dependency on such parameters can be regarded in Fluent using UDF or direct input in the user interface.

Table 3 Thermophysical properties of fluids[9]

Params	Units	Values				
Temp.	K	273.15	300	320	340	360
Density	kg/m^3	999.83	996.66	989.47	979.48	967.23
Sp. heat	J/kgK	4217	4179	4180	4188	4202
Thermal cond.	W/mK	0.562	0.6104	0.6369	0.6569	0.671
Viscosity coeff.	kg/ms	0.0017 917	0.0008 544	0.0005 772	0.0004 225	0.0003 267

UDS defines the physical properties of adsorbents. For this reason, physical properties include density, specific heat, and viscosity coefficient. The diffusion coefficient is unknown but was set to 0.01 for all four species. The larger the diffusion coefficient, the more stable the calculation, so it was specified to stabilize it. The properties of the filter section are equivalent to those of stainless steel. Density: 7,890 (kg/m^3), specific heat: 590 (J/kg·K), thermal conductivity: 16.7 (W/m·K), porosity: 0.1. The properties of the adsorbent-filled layer are defined by the following: density: 930.159 (kg/m^3), thermal conductivity: 0.472984 (W/m·K), and specific heat: 2019.048 (J/kg·K). The above values were calculated based on the physical properties provided by the customer. The temperature of the nitric acid solution was 25°C. Although the value will be different when pure water flows into the system, it can be approximated as follows: If pure water flows in, it will be a different value, but a fixed value is used as if a 20 wt% nitric acid solution always flows in as an approximation). filling density: 586 kg/m^3, porosity (γ): 0.37, Effective thermal conductivity of filled layer: 0.5 (W/m·K), effective specific heat of filled layer: 2.53 (kJ/kg·K). The definitions of effective thermal conductivity, adequate specific heat, and the solid properties calculated from them are as follows. Note that k represents thermal conductivity (W/m·K), γ represents porosity (-), Cp represents specific heat (J/kg·K), the subscript f represents the nitric acid solution, and s represents the adsorbent.

$$k_{eff} = \gamma k_f + (1 - \gamma) k_s \quad (17)$$

$$Cp_{eff} = \gamma Cp_f + (1 - \gamma) Cp_s \quad (18)$$

$$k_s = \frac{k_{eff} - \gamma k_f}{1 - \gamma} = \frac{0.5 - 0.37 * 0.546}{0.63} = 0.472984 \quad (19)$$

$$k_s = \frac{k_{eff} - \gamma k_f}{1 - \gamma} = \frac{0.5 - 0.37 * 0.546}{0.63} = 0.472984 \quad (20)$$

$$\rho_s = \frac{586}{0.63} = 930.159 \quad (21)$$

The inlet of the solution is assumed to be a velocity inlet, with a velocity of 6.6666667e-04 m/s (equivalent to 4 cm/min) and a cross-sectional area of 0.1 × 0.1 × π = 0.0314 m². The temperature of the solution is set to 323 K. The concentration of the chemicals at the inlet is a function of time, as follows: t = 0 - 487.5 (s): 20 wt% nitric acid solution containing ions A, B, C, and D flows in (equivalent to 0.5-bed volume (BV), filled layer volume BV = 3.14 × 0.1 × 0.1 × 0.65 = 0.02041 (m^3). 1 BV = 0.02041 / (6.666667e-04 × 0.0314) = 975 (s)), t = 487.5 - 2437.5 (s): 20 wt% nitric acid solution without ions flows in (equivalent to 0.5 - 2.5 BV), t = 2437.5 - 5118.75 (s): pure water flows in (equivalent to 2.5 - 10.5 BV). Such a condition is described by the expression function prepared in Fluent, and this

information is also documented in PyFluent. The inlet ion concentrations (defined by UDS) are as follows: A, 0.47 kg/m³; B, 0.15 (kg/m³); C, 2.28 (kg/m³); D, 3.90 (kg/m³). The flow outlet is a pressure outlet at a constant static pressure (gauge pressure = 0, equivalent to atmospheric pressure). The inlet fluid temperature with backflow is 323 K, and the UDS value is 0 when backflow occurs. This temperature is not used when there is no backflow. The side wall temperature is also fixed at 323 (K). The adsorbent filling layer and filter layer are defined as the porous body, where the loss (Pa·m) is assumed to be due to viscous resistance, as shown in Eqs. (22) and (23), where u is the flow velocity in each direction.

$$\frac{\Delta P}{L} = (1.45 \times 10^{8})u \quad (22),$$

$$\frac{\Delta P}{L} = (1.45 \times 10^{10})u \quad (23)$$

The following Eq. (24) is calculated and returned to the solver, which then inputs 1/a for those regions.

$$\frac{\Delta P}{L} = \frac{1}{a}\mu \times u \quad (24)$$

Calculations were performed using the UDF, enabling us to simulate the adsorption and elution of each ion. The calculations can be unstable when a large number of species are assumed. In such a case, the deficiency relaxation factor was decreased.

4. Setting of User-Defined-Function

The header part of UDF has UDS setting information, as shown in **Table 4**. The IP is a coefficient representing the porosity (0~1) of the adsorbent packing layer and is used to calculate the equilibrium concentration of metal ions absorbed by the adsorbent. Fluent's cell zone settings (porous media) can also give the porosity, which was not used in this calculation. These constants change the number of arrays according to the value of UDS_num.

Table 4 Lists of constants

Variable	Contents	units	types
Thread_num	Zone ID of the adsorbent-filled layer	-	integer
UDS_num	Number of metallic elements to be calculated	-	integer
Nitric_Id	Nitric Acid Speciation ID	-	integer
Water_Id	Water Species ID	-	integer
ips	Porosity of adsorbent-filled layer (0 ~ 1)	-	real number

UDS and user-defined memory (UDM) were utilized according to the number of metal elements (UDS_num) to solve the adsorption equations for the respective elements. **Table 5** lists the descriptions of valuables used in the calculation. The K_k is the distribution ratio as a function of nitric acid concentration, described by two parameters, kk_x1, and kk_x2, to express the nitric acid concentration dependency on Kk obtained in the batch adsorption experiments. The overall mass transfer coefficient, $K_{fk} \times a_v$ (1/s), the parameter related to adsorption kinetics, is also described as the function of the nitric acid concentration. The procedure of column chromatography, conditioning, forming the absorption band, and desorption is simulated. As shown in UDM in **Table 6**, [Number of UDM] = ([UDS_num] × 5 + 1) to solve the differentiated equation using time steps.

Table 5 Valuables used in the calculation

Variable	Contents	units
M_x[UDS_num]	The molecular weight of the element	kg/mol
kk_x1[UDS_num]	Distribution ratio, K_k (-) as a function of nitric acid conc. y [mol/L], kk_x1[i] + kk_x2[i] * y	-
kk_x2[UDS_num]		L/mol
kfk_av_x1[UDS_num]	Overall mass transfer coefficient, K_{fk} × av [1/s], a function of nitric acid concentration. y [mol/L], kfk_av_x1[i] + kfk_av_x2[i] * y	1/s
kfk_av_x2[UDS_num]		L/mol・s
Ar_x[UDS_num]	The heat generated by alpha rays	W/kg
Br_x[UDS_num].	Heating value due to gamma rays	W/kg
MAX_ARRAY_SIZE	Size of temperature array to account for the temperature dependence of $Q_{0,\,max}$	-
temp[MAX_ARRAY_SIZE]	Temperature array for the temperature dependence of $Q_{0,\,max}$	K
Qzm_x[UDS_num] [MAX_ARRAY_SIZE]	$Q_{0,\,max}$ to calculate the saturated adsorption of each element	mol/m³
a_x[UDS_num]	a_i: Coefficient on saturation	m³/mol
Relaxation[UDS_num]	Relaxation coefficient of adsorption	-

Table 6 List of User-Defined-Memory (UDM)

Variable	units	Contents
UDM-0	mol/L	Molar nitrate concentration
UDM-1	kg/m³	Equilibrium concentration C_k^* calculated at the current time
UDM-2	kg/m³	Ion concentration that has already been adsorbed at the current time C_{sk}
UDM-3	kg/m³	Calculated migration of the element at the current time
UDM-4	kg/m³	Equilibrium concentration C_k^* calculated at the previous time

The compiled UDF contains some functionality; then, the functions are hooked to the respective part in Fluent. The boundary condition is set, followed by the already set condition. As an inlet boundary, the flow velocity is set as the speed at which aqueous solution flows from the inlet. The temperature at which the aqueous solution flows from the inlet is also set. Nitric acid concentration is set at the inlet boundary. By using the expression function prepared in Fluent, the injection of chemicals can be described as follows: IF(t <= 50 [s], 2 [mol/L] / Density * 63.012 [g/mol], 0). Note that mass fractions are dimensionless, with a range of 0 to 1. The UDS conditions for the number of metal elements are set at the inlet boundary. The thermal conditions, as specified in the problem, are set to adiabatic (Heat Flux = 0), fixed temperature, or a specified heat transfer coefficient. First, the overall initial values are set using the standard initialization method. The patch function is then used to override the nitric acid concentration. Finally, the resulting output is set up after defining parameters. The parameters are to be displayed during the calculation in the Fluent GUI, and screenshots can also be defined for creating animations. That setting is also overridden in the Jupyter Notebook. Then, the CAS file and the corresponding UDF files are placed in the appropriate directory for reading by the Jupyter Notebook.

5. Implementation of Calculation

From the Jupyter Notebook, launch the Fluent by specifying the number of cores, the solver's dimension (2D), and types (pressure-based). Then, read the CAS data, compile the UDF based on the information in the previously prepared YAML file, hook and set the compiled functions, and set the UDS number and values. The CAS file already has the setting, including monitoring and exporting the results, but it can be overridden.

6. Implementation of Calculation

To see the progress of transient calculation and the contours, setting by PyFluent or GUI of Fluent is needed. A time series monitor of volume integral (Volume Integral) to check changes in the total amount of each metal element in solution [kg] (UDS-*) and the total amount of metal elements adsorbed on the adsorption layer [kg] is convenient to see the progress at a given time in calculation. For example, convergence residuals for each time step, element concentrations from the element outlet, element distribution inside the column, and temperature distribution can also be monitored. During the calculation, the progress of the loop in the parametric survey is stored in a JSON file, allowing for the restart of the parametric study.

7. Example of the Calculation Results

The Inlet and outlet of ions A to D are shown in **Fig. 4(1)**, and the calculated chromatogram is shown in **Fig. 4(2)**. The chromatogram indicates that ions A are not absorbed, and B and D do not adsorb significantly under the given solution conditions. Due to the difference in kinetics, the shapes of B and D were different. Ion C absorbed a significant amount in the batch experiments, and this effect was evident in the results. The ion C did not come out during the flushing with 20 wt% nitric acid, which is preferable for the absorption of ion C. When the concentration of nitric acid is changed to zero, which is not preferable for the absorption of ion C but preferable for leaching, the ion C remains. **Figure 5** displays the contours of the calculation results. Fig. 5(1) describes the advection-diffusion phenomenon with adsorption. Ion C was absorbed into the column and was not initially eluted. As mentioned above and shown in Fig. 5(2), pure water as eluent was started from at t = 2437.5 - 5118.75 (s) (equivalent to 2.5 - 10.5 BV), and elution of the ion C immediately occurred, which resulted in the rapid peak rise as shown in Fig. 4(2). This result is obtained by varying

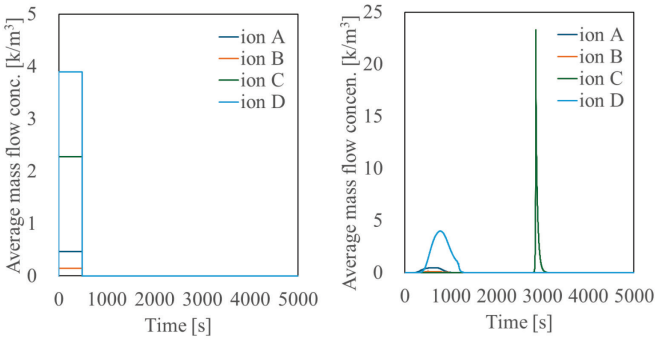

Fig. 4 Inlet and outlet of ions A to D: (1) inlet property, and (2) Chromatogram by the calculation

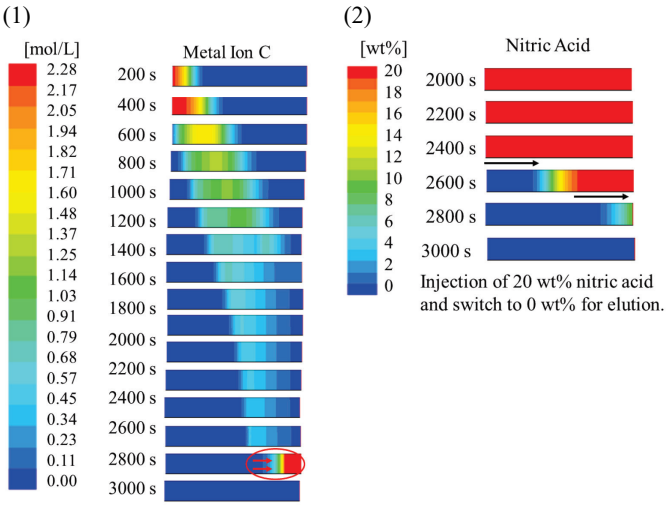

Fig. 5 Display of contours of calculation results: (1) concentration of ion C and (2) nitric acid concentration

multiple parameters and stored in respective folders, following the previously set rule.

8. Post-processing

With the predefined contour settings, the visualization data, as shown in Figs. 5(1) and 5(2), are also stored in predefined folders. The progress of the calculation and, occasionally, unexpected stops in the calculation loop can be communicated through various channels, such as Slack and Microsoft Teams. Even if such a case occurs, restarting the parametric survey loop is possible using the previously mentioned JSON file, and restarting the chromatographic calculation is possible by immediately outputting the Fluent data file.

III. Conclusion

We have developed an environment for efficiently executing chromatography calculations to handle multiple conditions by coupling ANSYS Workbench and ANSYS fluent Python (Pyfluent). The efficient calculation scheme, theory, and setup of the chromatographic calculation are illustrated, along with some typical calculated chromatograms and contours. To complete the calculation scheme shown in Fig. 1, we will incorporate some informatics techniques to anticipate or derive fitting parameters without numerous calculation attempts, utilizing machine learning techniques, which will enable the on-site determination of appropriate operation conditions for column chromatography in actual operational situations.

Acknowledgment

This work was part of the "Basic Research Programs on Vitrification Technology for Waste Volume Reduction (JPJ010599)" research project, commissioned by the Ministry of Economy, Trade and Industry, from fiscal year 2019 to 2023. I appreciate Dr. Takahiro Nishihara of the Institute of Science and Technology and Mr. Daisuke Tomihara of Cybernet System Co. for their discussions on creating the environment.

References

1) G. Van Rossum, F. L. Drake, (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace. Retrieved from http://www.python.org
2) ANSYS Inc., "PyANSYS," 2023. Available: https://pypi.org/project/pyansys/.
3) T. Kluyver, et al., (2016). Jupyter Notebooks - a publishing format for reproducible computational workflows. F. Loizides and B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents, and Agendas, pp. 87-90. IOS Press. https://eprints.soton.ac.uk/403913/
4) H. John D, Matplotlib: "A 2D Graphics Environment." *Computing in Science & Engineering*, **9**[3], 90-95 (2007).
5) W. McKinney, "Data Analysis in Python with Pandas." *Proc 9th Python in Science Conference**, 51-56 (2010).
6) ANSYS Inc. (2023). ANSYS SpaceClaim, Version 2023 R2, Retrieved from https://www.ansys.com
7) ANSYS Inc. (2023). ANSYS Workbench, Version 2023 R2, Retrieved from https://www.ansys.com
8) ANSYS Inc. (2023). ANSYS Meshing, Version 2023 R2, Retrieved from https://www.ansys.com
9) JSME Data book: Thermophysical Properties of Fluids, The Japan Society of Mechanical Engineers, **87**[788], 750 (1984), Online ISSN 2424-2675, https://doi.org/10.1299/jsmemag.87.788_750