Appendix A

Process description

The reactor wall was initially heated using twelve circumferential heating rods to reach the desired temperature from room temperature (27°C). Four temperature sensors placed inside the reactor monitored temperature uniformity. Simulations were conducted up to 2500°C to identify the limiting step during deposition. However, the reactor hardware limited the maximum temperature to 1000°C. After reaching the desired temperature, MTS vapour was introduced into the reactor along with a mixture of H_2 and Ar gases. The total gas flow rate was evenly distributed across the inlet nozzles. The mixture interacted with the buoyancy-driven flow, undergoing cracking and subsequent deposition on the substrate. The centrifugal force generated by the rotating shaft spread the reactants towards the reactor wall, and the optimal rotation speed was maintained to ensure uniform deposition. The amount of deposition was calculated by measuring the difference in sample weight before and after the reaction. A reaction time of 12 h was used consistently in all experiments. Following the reaction, the reactor was purged with Ar and allowed to cool to room temperature.

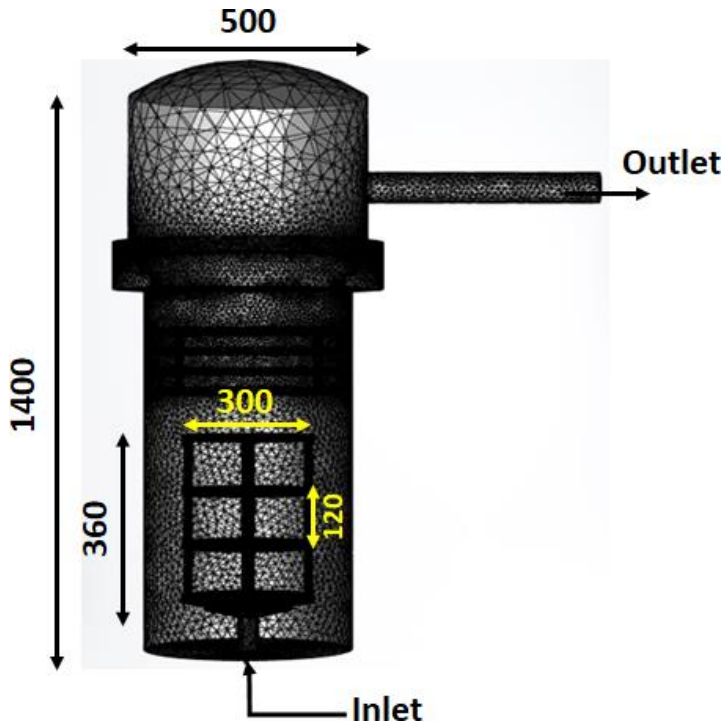


Fig. A1. Schematic of the SiC coating process (All dimensions are in mm).

Mathematical model

The CFD model for the CVD reactor is built on the conservation of mass, momentum, energy, and species and can be expressed as follows.

$$\frac{\partial \rho}{\partial t} + \nabla.(\rho \vec{u}) = 0 \tag{A1}$$

$$\frac{\partial (\rho \vec{u})}{\partial t} + \nabla.(\rho \vec{u}\vec{u}) = -\nabla p + \nabla.\tau + \rho \vec{g}\beta(T - T_a) \tag{A2}$$

Where ρ is the density,$u = (u_r, u_\theta, u_z)$ is the fluid velocity, p is the pressure, g is the gravity vector, τ is the viscous stress tensor. β and T_a is the coefficient of expansion of the gas mixture and the reference temperature, respectively.

$$\tau = \mu\left(\nabla \vec{u} + \nabla \vec{u}^T - \frac{2}{3}\nabla.\vec{u}I\right) \tag{A3}$$

uu and ∇u represent the dyadic products, and I denotes the unit tensor. The ideal gas law was used to calculate the density of the gas, and flow was assumed to be compressible. In the present study, the use of an ideal gas model is accurate because the CVD process is the following conditions are met: i) At low pressures, gas molecules are far apart, and intermolecular forces are minimal. ii) At high temperatures (>1000^0C), the kinetic energy of the gas molecules is significant, overwhelming the effects of intermolecular forces. iii) Gases with small molecules, such as hydrogen and methane, are less likely to deviate from ideal behaviour due to their reduced volume and minimal intermolecular interactions.

$$\rho = \frac{P}{RT\sum_{i=1}^{n}\frac{y_i}{M_i}} \tag{A4}$$

P, R, T, y_i and M_i are the reactor operating pressure, the universal gas constant, temperature, mass fraction, and molar mass of species i, respectively. The kinetic theory of gases was used to estimate the viscosity of gases [1].

$$\mu = 2.6693 \times 10^{-5}\frac{\sqrt{M_{avg}T}}{\sigma_i^2 \Omega} \tag{A5}$$

Where σ_i and Ω are the collision diameter and the collision integral, respectively. In general, Ω is a function of the dimensionless temperature, which can be defined as $T^{\#} = \varepsilon T/k_B$ and is evaluated using the empirical formula given in Eq. (6).

$$T^{\#} = \frac{\varepsilon}{k_B} = 0.77T_c, \text{ and } \sigma_i = 2.44(T_c/P_c)^{\frac{1}{3}} \tag{A6}$$

The critical temperature and pressure were estimated using a group-contribution method [2]. The specific heat capacity of species i at constant pressure C_{p_i} is expressed in the NASA polynomials. The coefficient of these polynomials was obtained from the literature [3].

$$\frac{C_{p_i}}{R} = \alpha_0 + \alpha_1 T + \alpha_2 T^2 + \alpha_3 T^3 + \alpha_4 T^4 \quad \text{(A7)}$$

The energy equation is written as

$$\rho C_p \frac{\partial T}{\partial t} + \nabla.\left(\rho C_p T \vec{u}\right) = \nabla.(-k\nabla T) - \sum_{i=1}^{n} M_i h_i R_i \quad \text{(A8)}$$

Where k is the thermal conductivity of the gas mixture, h_i is the enthalpy of species i, and R_i the volumetric net rate of production of species i due to the reactions (mol/m^{3}s). The mixture enthalpy is computed as

$$h_i = h_i^0 + \int_{T_0}^{T} C_{p_i} dT \text{ and } h = \sum_{i=1}^{n} y_i h_i \quad \text{(A9)}$$

Here, T_0 is the reference temperature. A simple semi-empirical formula developed by Eucken was used to approximate the thermal conductivity of a polyatomic gas [4].

$$k = \mu\left(C_p + \frac{5R}{4M}\right) \quad \text{(A10)}$$

The species transport of reacting flow is given by

$$\frac{\partial c_i}{\partial t} + \nabla.(c_i \vec{u}) = \nabla.\left(-D_i \nabla c_i - D_{T_i} \nabla T/T\right) + M_i \sum_{i=1}^{n_r} R_i \quad \text{(A11)}$$

Where D_i and D_{T_i} are the mass and thermal diffusion of species i, respectively. The diffusivity of the binary gas mixture was estimated using the Chapman-Enskog equation [4]:

$$D_{ij} = 1.8583 \times 10^{-3} \frac{\sqrt{T^3\left(1/M_i + 1/M_j\right)}}{P\sigma_{ij}^2 \Omega_{D,ij}} \quad \text{(A12)}$$

Where D_{ij} is the binary diffusivity of the mixture i-j. σ_{ij} is calculated by averaging the individual collision diameters: $\sigma_{ij} = 0.5(\sigma_i + \sigma_j)$.

The Soret diffusion coefficient was estimated using the following empirical equation [5].

$$D_{T,i} = -2.59 \times 10^{-7} T^{0.659} \left[\frac{M_{w,i}^{0.511} x_i}{\sum_{i=1}^{n} M_{w,i}^{0.511} x_i} - y_i\right]\left[\frac{\sum_{i=1}^{n} M_{w,i}^{0.511} x_i}{\sum_{i=1}^{n} M_{w,i}^{0.489} x_i}\right] \quad \text{(A13)}$$

x_i and y_i are mole and mass fractions of species i, respectively. Both the mass diffusivity and thermal diffusion were combined and specified in the simulation using a user-defined function.

The existing models commonly use a constant wall temperature profile or constant heat transfer coefficient to simplify the complexity of heat transfer phenomena. Unlike the conventional approaches, in this work, we consider a more realistic situation and propose detailed heat transfer boundary conditions to illustrate the effective heat transfer between the inner reactor gas and the outer reactor surface. More precisely, we use the well-known convective heat flux as follows:

$$n.\left(-k\nabla T\right) = q_{in} = h_{in}\left(T - T_{inlet}\right) \quad \text{(A14)}$$

In the above context, q refers to the inward convective heat fluxes, which are normal to the inner reactor wall. The term h_{in} (W/m^2/K) represents the heat transfer coefficient of the inner mixture gas. The temperatures of the inner mixture gas are indicated by T. This work proposes an empirical correlation for the heat transfer coefficients, which is based on forced convection across the inner reactor walls.

$$h = C\frac{k}{D_H}\mathrm{Re}^n\,\mathrm{Pr}^m \tag{A15}$$

In this context, k represents the thermal conductivity of the flow, and D_H denotes the characteristic length of the system. The Reynolds number Re and the Prandtl number Pr are dimensionless quantities used to describe the flow dynamics. The constants C, n, and m are dimensionless parameters that depend on the specific geometry of the reactor. These parameters are typically determined by approximating the actual reactor geometry with an equivalent standard configuration, such as a plate or cylinder. For instance, by approximating the inner reactor geometry as a hot plate subjected to forced laminar flow, the correlation parameters can be defined as [C, n, m] = [0.664, 0.5, 0.333]. Similarly, the heat transfer between the outer reactor wall and the surrounding atmosphere is characterized by free convection flux, which can be expressed as follows:

$$n.\left(-k_c\nabla T_c\right) = q_w = h_{iw}\left(T_w - T_a\right) \tag{A16}$$

The heat transfer coefficient under free convection conditions for the outer tube wall is defined as (h_w = 7.5 (W/m^2/K).

Table: The boundary conditions used in the simulation.

Boundary location	**Momentum conservation equation**	**Energy conservation equation**	**Species conservation equation**
Inlet	$u_r = u_\theta = 0,$ $u_z = u_{inlet} = 5\ cm/s$	$T = T_{inlet} = 80^0C$	$y_i = y_{i\_inlet} = 0.02$
Outlet	$p = p_0$ $\left[\mu(\nabla\vec{u} + \nabla\vec{u}^T) - \frac{2}{3}\nabla.\vec{u}\right].n = 0$	$n.(-k\nabla T) = 0$	$\eta.(-D_i c\nabla y_i) = 0$
Substrate	$u = 0$	$T = T_s$	$-n.N = R_i$ $N = -D_i c\nabla y_i + cy_i u$

Reactor wall	$u = 0$	$T = T_s$	$-n.N = R_i$ $N = -D_i c \nabla y_i + c_i u$

n represents the outward unit normal.

References

[1] B. Ning, T. Xia, Z-X. Tong, Y-L. He, Experimental and numerical studies of tungsten line growth in laser chemical vapour deposition, Int. J. Heat. Mass. Transf. 140 (2019) 564-578. https://doi.org/10.1016/j.ijheatmasstransfer.2019.06.001

[2] Y. Nannoolal, J. Rarey, D. Ramjugernath, Estimation of pure component properties: Part 3. Estimation of the vapour pressure of non-electrolyte organic compounds via group contributions and group interactions, Fluid Phase Equilibria, 269 (2008) 117-133. https://doi.org/10.1016/j.fluid.2008.04.020

[3] D. M. Matheu, A. M. Dean, J. M. Grenda, W. H. Green, Mechanism generation with integrated pressure dependence: a new model for methane pyrolysis, J. Phys. Chem. A 107 (2003) 8552-8565. https://doi.org/10.1021/jp0345957

[4] R. B. Bird, W. E. Stewart, E. N. Lightfoot, D. B. Spalding, Transport Phenomena, John Wiley & Sons (1961), pp. 338-359.

[5] K. K. Y. Kuo. Principles of Combustion. John Wiley and Sons, New York, 1986.

Appendix B

SVM and Nelder Mead algorithm code in R

```
# Import Libraries
library(tidyverse)
library(tidymodels)
library(tune)
library(workflows)
library(workflowsets)
library(caret)
library(kableExtra)
library(Rama)
library(glue)
library(flextable)
library(officedown)
library(officer)

ALL_Combinations = tibble(
  α = c(0.1, 1, 1,10, 100),
  β = c(10, 1, 10, 1, 1)
)

for(INDEX in 1:5){

  α = ALL_Combinations$α[INDEX]
  β = ALL_Combinations$β[INDEX]
```

```
# Read data
cone_data <- read_csv('Cone data_ML.xlsx - Sheet1.csv') %>%
  select(temp = Temp,pressure = Press,flow_rate =`flow rate`,mole_frac = molfract,DR,UI) %>%
  mutate(y = α*DR + β*UI)

minmax_cone = make_minmax(cone_data)

set.seed(2021)
spl <- initial_split(cone_data)
cone_train <- training(spl)
cone_test <- testing(spl)
train_5fold <- cone_train %>%  vfold_cv(5)

# Two SVM models were fitted separately for UI and DR. The test level performance measure for those are as follows:

DR_rec <- recipe(DR ~ ., data = cone_train %>% select(temp:mole_frac,DR)) %>%
  step_center(all_numeric(), -all_outcomes()) %>%
  step_scale(all_numeric(), -all_outcomes())

UI_rec <- recipe(UI ~ ., data = cone_train %>% select(temp:mole_frac,UI)) %>%
  step_center(all_numeric(), -all_outcomes()) %>%
  step_scale(all_numeric(), -all_outcomes())

cv <- trainControl(method = 'repeatedcv',number = 10,repeats = 5)
```

```
# Construct grid of hyperparameter values
hyper_grid <- expand.grid(k = 10^seq(-3, -1, by = .1))
# Tune α knn model using grid search
svmRFit_DR <- train(DR ~ ., data = cone_train %>% select(temp:mole_frac,DR),
                    method = "svmRadial",
                    tuneLength = 15,
                    preProc = c("center", "scale"),
                    trControl = cv)
svmRFit_UI <- train(UI ~ ., data = cone_train %>% select(temp:mole_frac,UI),
                    method = "svmRadial",
                    tuneLength = 15,
                    preProc = c("center", "scale"),
                    trControl = cv)
cat('\nDR')
dT_DR = tibble(observed = cone_test$DR,
               predicted = predict(svmRFit_DR,newdata = cone_test %>%
select(temp:mole_frac,DR)))

dT_DR %>% ggplot(aes(x = observed,y = predicted)) + geom_point() + geom_abline(color
= 'steelblue',size=1.2) +
  ggtitle(label = NULL,subtitle = 'Observed/Predicted Deposition Rate')

dT_DR %>% mutate(error = observed  - predicted, SL = row_number()) %>% ggplot() +
  geom_point(aes(SL, error))

dT_DR %>% mutate(error = observed  - predicted, SL = row_number()) %>% gghist(error)
```

```
cat('\nUI')
dT_UI = tibble(observed = cone_test$UI,
               predicted = predict(svmRFit_UI,newdata = cone_test %>% select(temp:mole_frac,UI)))

dT_UI %>%  ggplot(aes(x = observed,y = predicted)) + geom_point() + geom_abline(color = 'steelblue',size=1.2) +
  ggtitle(label = NULL,subtitle = 'Observed/Predicted UI')

dT_UI %>% mutate(error = observed  - predicted, SL = row_number()) %>% ggplot() +
  geom_point(aes(SL, error))

dT_UI %>% mutate(error = observed  - predicted, SL = row_number()) %>% gghist(error)

# Combined Model
y_rec <- recipe(y ~ ., data = cone_train %>% select(-DR,-UI)) %>%
  step_center(all_numeric(), -all_outcomes()) %>%
  step_scale(all_numeric(), -all_outcomes())

cv <- trainControl(method = 'repeatedcv',number = 10,repeats = 5)
# Construct grid of hyperparameter values
# Tune α knn model using grid search
svmRFit_y <- train(y ~ ., data = cone_train %>% select(-UI,-DR),
                   method = "svmRadial",
                   tuneLength = 15,
                   preProc = c("center", "scale"),
                   trControl = cv)
```

```
dT_y = tibble(observed = cone_test$y,
              predicted = predict(svmRFit_y,newdata = cone_test %>% select(temp:mole_frac)))

perform_tbl <- tibble(Measure = c('RMSE','Rsquared','MAE'),
                      DR = postResample(dT_DR$observed, dT_DR$predicted),
                      UI = postResample(dT_UI$observed, dT_UI$predicted),
                      y = postResample(dT_y$observed, dT_y$predicted)) %>%
  mutate_if(is.numeric,.funs = \(x) round(x,4))

# Optimization Setup

# The minimum and maximum values of the predictors and responses
minmax_cone %>% as.data.frame() %>%
  mutate_all(.funs = \(x) round(x,3)) %>% flextable()

## Optimized values
# The following combinations of predictors provide the optimized values of the parameters
# y=`r α`.DR + `r β`.UI

Function_to_Optimize <- function(x, model = svmRFit){
  # x = unlist(cbResults[i,1:4 ] )
  # minmax_scal e d
  if(x[1] < minmax_cone$temp[1]      | x[1] > minmax_cone$temp[2]     ) { return(10^138)}
  if(x[2] < minmax_cone$pressure[1]  | x[2] > minmax_cone$pressure[2] ) {return(10^138)}
  if(x[3] < minmax_cone$flow_rate[1] | x[3] > minmax_cone$flow_rate[2]) {return(10^138)}
```

```
  if(x[4] < minmax_cone$mole_frac[1] | x[4] > minmax_cone$mole_frac[2])
{return(10^138)}


  tmp <- as.data.frame(t(x))
  names(tmp) <- c("temp","pressure","flow_rate","mole_frac")
  y  = round(predict(model, tmp), 3)
  if(any(y > minmax_cone$y[2] , y < minmax_cone$y[1])){
    print(cbind(tmp,y))
    return(10^78)
  } else {
    return(-y)
  }
 }
  # revive_val <- function(x, min_max_i)  x*min_max_i[2] + min_max_i[1]
 set.seed(10-10-10)
 subTrain <- subset(cone_train)#, Age == 28)
 ### Center and scale the data to use dissimilarity sampling
 pp1 <- preProcess(subTrain[, 1:4], method = 'pca')#c("center", "scale"))
 scaledTrain <- predict(pp1, subTrain[, 1:4])


 ### Randomly select α few mixtures as α starting pool
 set.seed(91)
 set.seed(00)
 startMixture <- sample(1:nrow(subTrain), 1)
 starters <- scaledTrain[startMixture, 1:3]
 index <- maxDissim(starters, scaledTrain, 10)
```

```
  startPoints <- c(startMixture, index)


  starters <- subTrain[startPoints,1:4]
  startingValues <- starters
  ### For each starting mixture, optimize the Cubist model using
  ### α simplex search routine


  # optimize_val_by_model <- function(intialValues=startingValues,MODEL = svmRFit_Z,model_name = 'SVM-Reg'){
  cbResults <- startingValues
  cbResults$y <- NA


  #  optimx::optimx(unlist(cbResults[i,1:4]),Function_to_Optimize, lower = 0, upper = 6.212, method = 'Nelder-Mead')
  for(i in 1:nrow(cbResults)){
    # i = 1
    # cat('\n',glue('\n=============== ROW#{i} ==================='),'\n')
    results <- optim(unlist(cbResults[i,1:4]),fn = Function_to_Optimize,
                     method = "Nelder-Mead",
                     control=list(maxit=5000,
                                  # trace = 20,
                                  abstol = 0.01),mod = svmRFit_y)


    cbResults$y[i] <- -results$value
    cbResults[i,1:4] <- rbind(results$par %>% unlist %>% as.vector())
  }
```

```
Opt_svmReg = cbResults %>%
  mutate(UI_pred = predict(svmRFit_UI,cbResults[,-5]),
         DR_pred = predict(svmRFit_DR,cbResults[,-5])
  ) %>%
  mutate_all(.funs = \(x) round(x,3)) %>%
  mutate_at(vars(temp, pressure),.funs = \(x) round(x,0)) %>%
  mutate_at(vars(flow_rate),.funs = \(x) round(x,2)) %>%
  select(temp,pressure, flow_rate, mole_frac, DR_pred, UI_pred, y_pred = y) %>%
  arrange(-y_pred) |> print()
  writexl::write_xlsx(Opt_svmReg,paste0('Opt_svmReg_',α,'_',β,'.xlsx'))
}
```

Appendix C

Validation of the SVM model and Nelder-Mead Algorithm with a well-defined problem

Problem Formulation

The problem consists of a two-dimensional optimisation variable and two objective functions.

Minimize

$$f_1 = x_1^4 + x_2^4 + x_1 x_2 - x_1^2 x_2^2 - 8x_1^2$$

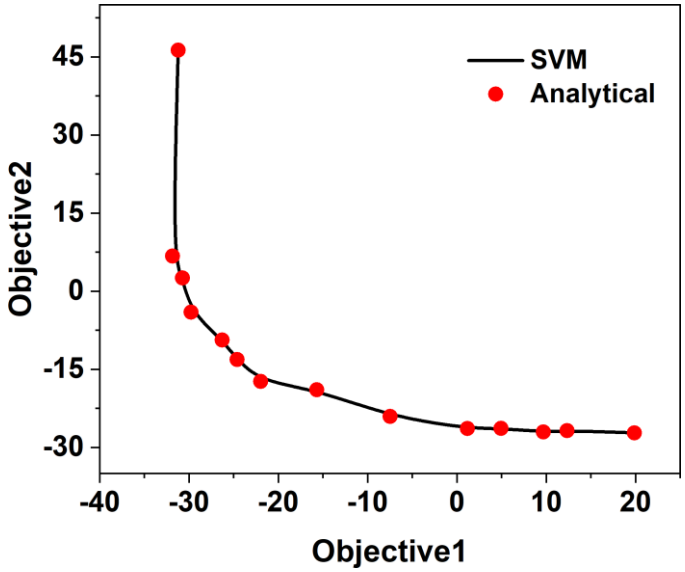
$$f_2 = x_1^4 + x_2^4 + x_1 x_2 - x_1^2 x_2^2 - 3x_1^3$$

Subject to

$$-50 \le x_1 \le 50$$

$$-50 \le x_2 \le 50$$

Plot the resulting Pareto front.



This example shows the plot of a multiobjective optimization problem using the SVM model and Nelder-Mead Algorithm proposed in this thesis.

Appendix D

The effect of thermal diffusion on the velocity profile and concentration of methane is shown in Fig. D1 and D2.

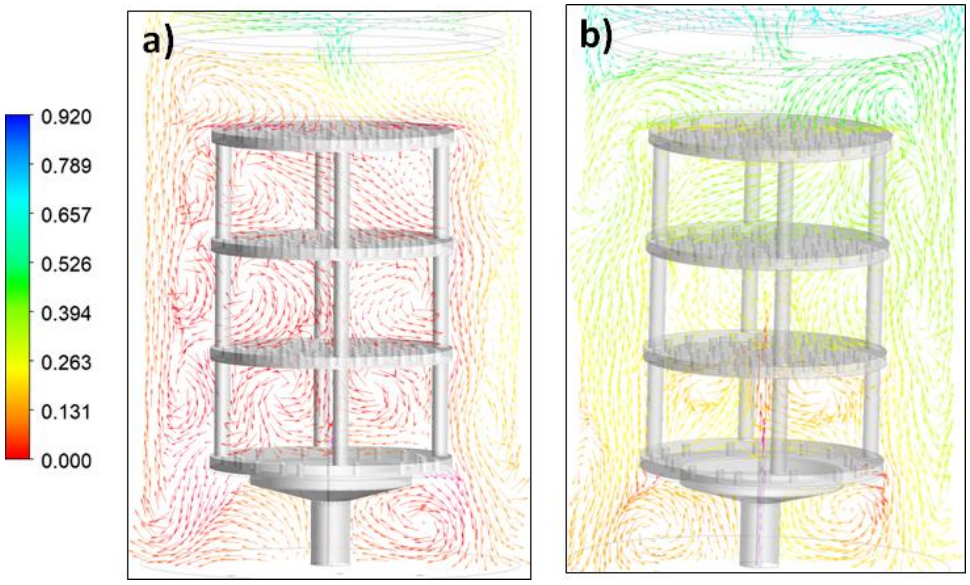


Fig D1: Velocity distribution inside the reactor a) without thermal diffusion and b) with thermal diffusion.

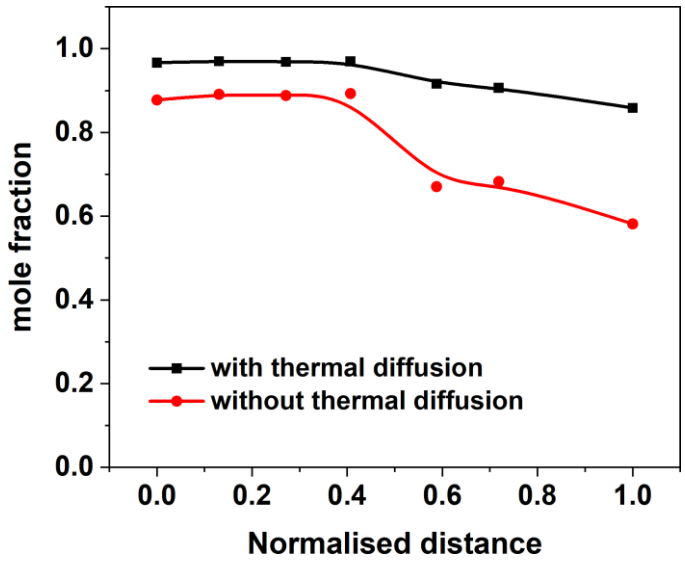


Fig D2: Distribution of mole fraction of methane inside the reactor without thermal diffusion and with thermal diffusion.

Glossary

CFD (Computational Fluid Dynamics): A branch of fluid mechanics that uses numerical analysis and algorithms to solve and analyze fluid flow problems.

CVD (Chemical Vapor Deposition): A chemical process used to produce high-purity, high-performance solid materials, commonly used in semiconductor manufacturing.

Nelder-Mead Algorithm: An optimization algorithm used to minimize an objective function in a multidimensional space without needing derivatives.

Reynolds Number (Re): A dimensionless number used to predict flow patterns in fluid dynamics. It is the ratio of inertial forces to viscous forces.

Grashof Number (Gr): A dimensionless number that estimates the ratio of buoyant to viscous forces in a fluid, important in natural convection processes.

Prandtl Number (Pr): A dimensionless number that characterizes the ratio of momentum diffusivity (viscosity) to thermal diffusivity.

Sherwood Number (Sh): A dimensionless number representing the ratio of convective mass transfer to diffusive mass transfer.

Residuals: The difference between observed and computed values in numerical simulations, used to monitor convergence.

Deposition Rate: The rate at which material accumulates on a substrate during processes like CVD, typically measured in $kg/m^2 \cdot s$.

Uniformity Index: A measure of the consistency or evenness of material deposition across a surface, with values closer to 1 indicating greater uniformity.

Boundary Conditions: Constraints applied to a computational domain in simulations, such as inlet/outlet velocities, temperatures, or heat fluxes.

Implicit Method: A numerical method used in time-dependent simulations where the solution at the next time step depends on unknown future values, often more stable than explicit methods.

Explicit Method: A numerical method where the solution at the next time step is based solely on known values from the current time step.

Time Step (Δt): The increment of time over which numerical calculations are carried out in simulations.

Mesh Independence Study: A study conducted to ensure that the results of a simulation are not dependent on the size or density of the computational mesh.

Convergence Criteria: The conditions under which a numerical solution is considered to have reached a stable and accurate result, often monitored through residuals or flux imbalances.

Kinematic Viscosity: A measure of a fluid's resistance to flow under the influence of gravity, defined as the ratio of dynamic viscosity to fluid density.

Thermal Conductivity: A material's ability to conduct heat, measured in W/m·K.

Mass Diffusivity: A measure of how quickly molecules spread out or diffuse in a given medium, often temperature-dependent.

SLM: Standard liter per minute is a unit of volumetric flow rate, specifically it is commonly used to express the flow rate of gases under standard conditions of temperature and pressure. 1slm= 1.6667×10^{-5} m^3/s.

Finite Volume Method (FVM): A numerical technique used in CFD that discretizes the equations governing fluid flow over small control volumes.

Turbulent Flow: A type of fluid flow characterized by chaotic, irregular motion, typically occurring at high Reynolds numbers.

Laminar Flow: A smooth, orderly flow regime where fluid particles move in parallel layers with little to no mixing, typically occurring at low Reynolds numbers.

Heat Flux: The rate of heat energy transfer through a given surface, measured in W/m^2.

Boundary Layer: A thin region adjacent to a solid surface in fluid flow where viscous forces dominate, affecting heat and mass transfer.

Convective Heat Transfer: The transfer of heat through a fluid (gas or liquid) due to the movement of the fluid itself.

Adiabatic Boundary Condition: A type of boundary condition where no heat transfer occurs through the surface, meaning the surface is perfectly insulated.

Thermal Expansion Coefficient (β): A property of materials that describes how their volume changes with temperature, crucial in buoyancy-driven flows.

Multiphysics Simulation: Simulations that involve solving multiple types of physical processes, such as heat transfer, fluid flow, and chemical reactions, simultaneously.

Steady-State: A condition where the variables of a system (such as temperature, velocity, and pressure) remain constant over time.

Transient Simulation: A time-dependent simulation where the variables change with time, used to study systems that are not in a steady state.

Isothermal Process: A process in which the temperature remains constant throughout, often used as a simplification in simulations.

Species Transport: The process of tracking the movement and concentration of different chemical species in a fluid, often crucial in chemical reaction modeling.

Kinematic Boundary Condition: Boundary conditions applied to velocity fields in fluid simulations, determining how the fluid interacts with the boundary (e.g., no-slip condition).

No-Slip Condition: A boundary condition where the velocity of the fluid at the boundary (e.g., a solid surface) is equal to the velocity of the surface, typically zero.

Enthalpy: A measure of the total energy of a system, including internal energy and the energy required to displace its environment, often used in heat transfer calculations.

Chemical Reaction Rate: The speed at which a chemical reaction occurs, often influenced by temperature, pressure, and concentration of reactants.

Knudsen Number: A dimensionless number used to determine the regime of gas flow, defined as the ratio of the molecular mean free path length to a characteristic physical length scale.

Diffusive Transport: The movement of particles from a region of high concentration to a region of low concentration due to random molecular motion.

Peclet Number (Pe): A dimensionless number that characterizes the relative importance of advective to diffusive transport processes.

Activation Energy (Ea): The minimum energy required for a chemical reaction to occur, an important parameter in reaction kinetics.

Surface Reaction Mechanism: The set of elementary steps describing how molecules interact and react on the surface of a material, often used in catalytic processes like CVD.

Equilibrium State: A state where all forces and reactions in a system are balanced, and there are no net changes in the system's properties over time.

Timestep Convergence: A criterion used in transient simulations to ensure that the chosen timestep size is small enough to accurately capture the system's behavior over time.

Computational Grid (Mesh): The discretized representation of the physical domain over which CFD simulations are performed, with finer meshes leading to higher accuracy at the cost of computational effort.

Residual Error: The difference between the computed and exact values in a numerical simulation, used to evaluate convergence.

Machine Learning (ML): A subset of artificial intelligence (AI) that involves the development of algorithms that enable computers to learn from and make predictions or decisions based on data.

Support Vector Machine (SVM): A supervised machine learning model used for classification and regression tasks. It works by finding the optimal hyperplane that best separates data into different classes.

Training Data: A subset of data used to teach a machine learning model how to recognize patterns and make predictions. The model is adjusted based on the data during training.

Validation Data: A set of data used to tune the parameters of a machine learning model. It helps to prevent overfitting and ensures the model generalizes well to new data.

Test Data: A separate dataset used to evaluate the performance of the machine learning model after it has been trained and validated.

Hyperparameter: Parameters whose values are set before the learning process begins, and they control the behavior of the machine learning model, such as the learning rate or regularization strength.

Regression: A type of machine learning task where the goal is to predict continuous values based on input data, such as predicting deposition rates in a CVD reactor.

Overfitting: A modeling error that occurs when a machine learning model is too complex and learns the noise in the training data rather than the actual pattern, leading to poor performance on new data.

Underfitting: A situation where a machine learning model is too simple to capture the underlying patterns in the data, resulting in poor performance on both training and new data.

Kernel Function: A function used in SVMs and other algorithms to transform data into a higher-dimensional space, making it easier to find a separating hyperplane in complex datasets.

Nelder-Mead Algorithm: An optimization algorithm used to find the minimum or maximum of an objective function in a multidimensional space, often used in conjunction with machine learning models for optimization.

PUBLICATIONS

- **Anand Gupta**, Abhisek Mudgal, Vijay M. Shinde, Harish Kumar, N. Eswara Prasad, *Optimal design of CH_4 pyrolysis in a commercial CVD reactor using support vector machines and Nelder-Mead algorithm*. **Chemical Engineering Research and Design**, 2022;178:124-135.

- **Anand Gupta,** Shikhar Nigam, Vijay M. Shinde, *Gas-phase kinetic of boron carbide chemical vapour deposition using BCl_3+CH_4+H_2 mixture*. **Journal of the American Ceramic Society**, 2022; 105:3885-3895.