Electronic Supplementary Information (ESI) for Lab on a Chip.
This journal is © The Royal Society of Chemistry 2021

Supplementary Information for

Machine learning assisted fast prediction of inertial lift in microchannels

Jinghong Su[1, 2, 4,a], Xiaodong Chen[3, a], Yongzheng Zhu[2, 4], Guoqing Hu[1,*]

[1]Department of Engineering Mechanics, State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310027, China

[2]The State Key Laboratory of Nonlinear Mechanics (LNM), Institute of Mechanics, Chinese Academy of Sciences, Beijing 100190, China

[3]School of Aerospace Engineering, Beijing Institute of Technology, Beijing 100081, China

[4]School of Engineering Science, University of Chinese Academy of Sciences, Beijing 100049, China

a Equal contribution.

MATLAB code

```
clc
clear

%% I. Specify the operating situation
shape= 'semicircular';    % The options include 'rectangular', 'triangular', and 'semicircular'.
AR=2;           % Specify channel aspect ratio
Re=50;          % Specify channel Reynolds number
kappa=0.3;    % Specify particle blockage ratio

k=0;
if strcmp(shape, 'rectangular')
    lateral_y =0:0.1:(0.9-kappa);      % Specify corresponding 2y/H
    lateral_z =0:0.1:(AR-0.1-kappa); % Specify corresponding 2z/H
    for i=1:1:length(lateral_y)
        for j=1:1:length(lateral_z)
            k=k+1;
            Output(:,k)=[AR;Re;kappa;lateral_y(i);lateral_z(j)./AR];
        end
    end
elseif strcmp(shape, 'semicircular')
    for i=1:1:20                        % Specify corresponding 2y/H
        for j=1:1:20                    % Specify corresponding 2z/H
            lateral_y=i*0.1;
            lateral_z=(j-1)*0.1;
            if lateral_y^2+lateral_z^2<(2-kappa-0.099)^2 && lateral_y>(kappa+0.099)
                k=k+1;
                Output(:,k)=[Re;kappa;lateral_y-1;lateral_z];
            end
        end
    end
elseif strcmp(shape, 'triangular')
    num=19-kappa/0.05;
    low=-(0.84-(kappa-0.1)/0.05*0.04); % The maximum value of 2y/H
    up=(0.44-(kappa-0.1)/0.05*0.04);      % The minimum value of 2y/H
    for i=1:1:num
        for j=1:1:num+1-i
            lateral_y=(i-1)*(up-low)/(num-1)+low;          % Specify corresponding 2y/H
            lateral_z=(j-1)*(up-lateral_y)/2/(num-i);       % Specify corresponding 2z/H
```

```
                if i>num-1
                    lateral_z=0;
                end
                k=k+1;
                Output(:,k)=[Re;kappa;lateral_y;lateral_z];
            end
        end
else error('Please choose the channel shape');
end

%% II. Import the database
database = importdata(['lift_database_',shape,'.data'])';
if strcmp(shape, 'rectangular')
        Input =database(1:5,:);                % Five elements: AR Re kappa 2y/H 2z/H
        Input(5,:) =Input(5,:)./Input(1,:); % Preprocessing of the 2z/H
        Target_CL =database(6:7,:);            % Two elements: CLy and CLz
else
        Input =database(1:4,:);                % Four elements: Re kappa 2y/H 2z/H
        Target_CL =database(5:6,:);            % Two elements: CLy and CLz
end
%% III. 0-1 normalization
[input, ps_input] = mapminmax(Input,0,1);
output = mapminmax('apply',Output,ps_input);
[target_CL, ps_CL] = mapminmax(Target_CL,0,1);

%% IV. Neural Network Training
% 1. Build the network
net_CL = newff(input,target_CL,[20 8],{'tansig' 'tansig' 'purelin'});

% 2. Set training parameters
net_CL.trainParam.epochs = 1000;
net_CL.trainParam.goal = 1e-5;
net_CL.trainParam.mu = 0.001;

% 3. Perform the network
net_CL = train(net_CL,input,target_CL);

%% V Neural Network Prediction
predict_CL = sim(net_CL,output);

%% VI Anti-normalization
Predict_CL = mapminmax('reverse',predict_CL,ps_CL);
```

```
%% VII. Output lift distribution for the operating situation
fp = fopen('lift.data','w+');
fprintf(fp, '%s\n',[shape,' channel']);
fprintf(fp, '%s\t','VARIABLES="2y/H","2z/H","CLy","CLz"');
fprintf(fp, '\n');
if strcmp(shape, 'rectanglar')
    fprintf(fp, '%s\t',['operating situation: ','AR=',num2str(AR),', Re=',num2str(Re),', 
kappa=',num2str(kappa,'%.2f')]);
    for i=1:length(Output(1,:))
        fprintf(fp, '\n');
        fprintf(fp, '%5f\t',Output(4,i), Output(5,i)*AR,Predict_CL(1,i),Predict_CL(2,i));
    end
else
    fprintf(fp, '%s\t',['operating situation: ','Re=',num2str(Re),', 
kappa=',num2str(kappa,'%.2f')]);
    for i=1:length(Output(1,:))
        fprintf(fp, '\n');
        fprintf(fp, '%5f\t',Output(3,i), Output(4,i),Predict_CL(1,i),Predict_CL(2,i));
    end
end
fclose(fp);
```

Mapping procedure

The inertial lift for specified operating conditions was mapped to the cross-section of the microchannel in COMSOL 5.3a by setting the Interpolation option of the user-defined function in the Global Definitions module. Here, the user-defined lift function was obtained by importing the lift data file outputted from MATLAB in the Interpolation option. There are four columns in the lift data file, which correspond to $2y/H$, $2z/H$, C_{Ly} and C_{Lz}, respectively. The first column corresponds to the short axis of the rectangular microchannel or the symmetry axis of the triangular microchannel and semicircular microchannel.

As shown in Fig. S2, geometry or orientation of designated microchannels sometimes is not the same as what we simulated, *e.g.*, the curved channel and the oblique straight channel. Therefore, the coordinate transformation should be conducted before the lift forces are mapped. Taking the rectangular microchannel for example, we assume that in a coordinate system (x_1, x_2, x_3), the x_3 - *axis* corresponds to the direction of height while the coordinate of the channel center axis is (x_{1_axis}, x_{2_axis}, x_{3_axis}). In each x_1 - x_2 plane, we have the values of the center of curvature, (x_{1c}, x_{2c}), and the medium radius of curvature, R. Here, the absolute values of $2(x_3 - x_{3\_axis})/H$ and $2Rn/H$ correspond to the $2y/H$ and $2z/H$, respectively, where the absolute value of Rn is the normal distance to the centerline of the channel in each x_1 - x_2 plane, *i.e.*, the normal distance to the short axis within the cross-section. For curved channels, the Rn can be derived by $DTocenter - R$, where $DTocenter$ is the distance between (x_1, x_2) and (x_{1c}, x_{2c}) in each x_1 - x_2 plane. For straight channels, the absolute value of Rn can be derived by calculating the distance between the particle position (x_1, x_2) and (x_{1_axis}, x_{2_axis}). When the straight channel is horizontally placed, *i.e.*, $\theta = 0$, then x_{2_axis} = constant value, $x_{1\_axis} = x_1$ and $Rn = x_2 - x_{2\_axis}$. When the straight channel is vertically placed, *i.e.*, $\theta = \pi/2$, then x_{1_axis} = constant value, $x_{2\_axis} = x_2$ and $Rn = x_1 - x_{1\_axis}$. Because our outputted inertial lift data from MATLAB are for a quarter of cross-section, we should determine the sign of the lift on the other parts of the cross-section. By taking the formula (c3 - c5) or formula (s2 - s4) into formula (b1 - b3), the lift forces on spherical particles located at arbitrary lateral positions for curved channel or straight channel can finally be derived and accurately mapped onto the cross-section of the designated channel.

The mapping of inertial lift forces on the triangular microchannel and semicircular microchannel can be achieved by similar process.

Simulated parameters

A wide range of parameters were considered, i.e., Re varies from 50 to 200 with an interval of 50, and κ varies from 0.10 to 0.30 with an interval of 0.05. In the square channel, $2y/H$ and $2z/H$ varies from 0 to 1-κ with an interval of 0.10 near the channel center and 0.05 near the channel wall. Three types of the rectangular channel were simulated with AR = 2, 3, and 4. Herein, $2y/H$ varies from 0 to 1-κ with an interval of 0.10 near the channel center and 0.05 near the channel wall, while 2z/H varies from 0 to AR-κ with an interval of 0.10AR near the channel center and 0.1 near the channel wall. The triangular channel simulated is isosceles and the base of the triangular channel is equal to the height. Herein, $2y/H$ and $2z/H$ were chosen based on that the maximum number of the simulated cases along y and z direction is 19-kappa/0.05. For the semicircular channel, $2y/H$ and $2z/H$ are chosen with an interval of 0.10 under the limitation of $2y/H$ > -1+κ and (2y/H+1)^2+(2z/H)^2 < 2-κ. The origin of the coordinate axis was set at the center of the symmetry axis of the channel. The Intervals of the parameters allow a systematical and representative investigation for 120 operating conditions. The total number of simulations is close to 15,000.

Prediction files

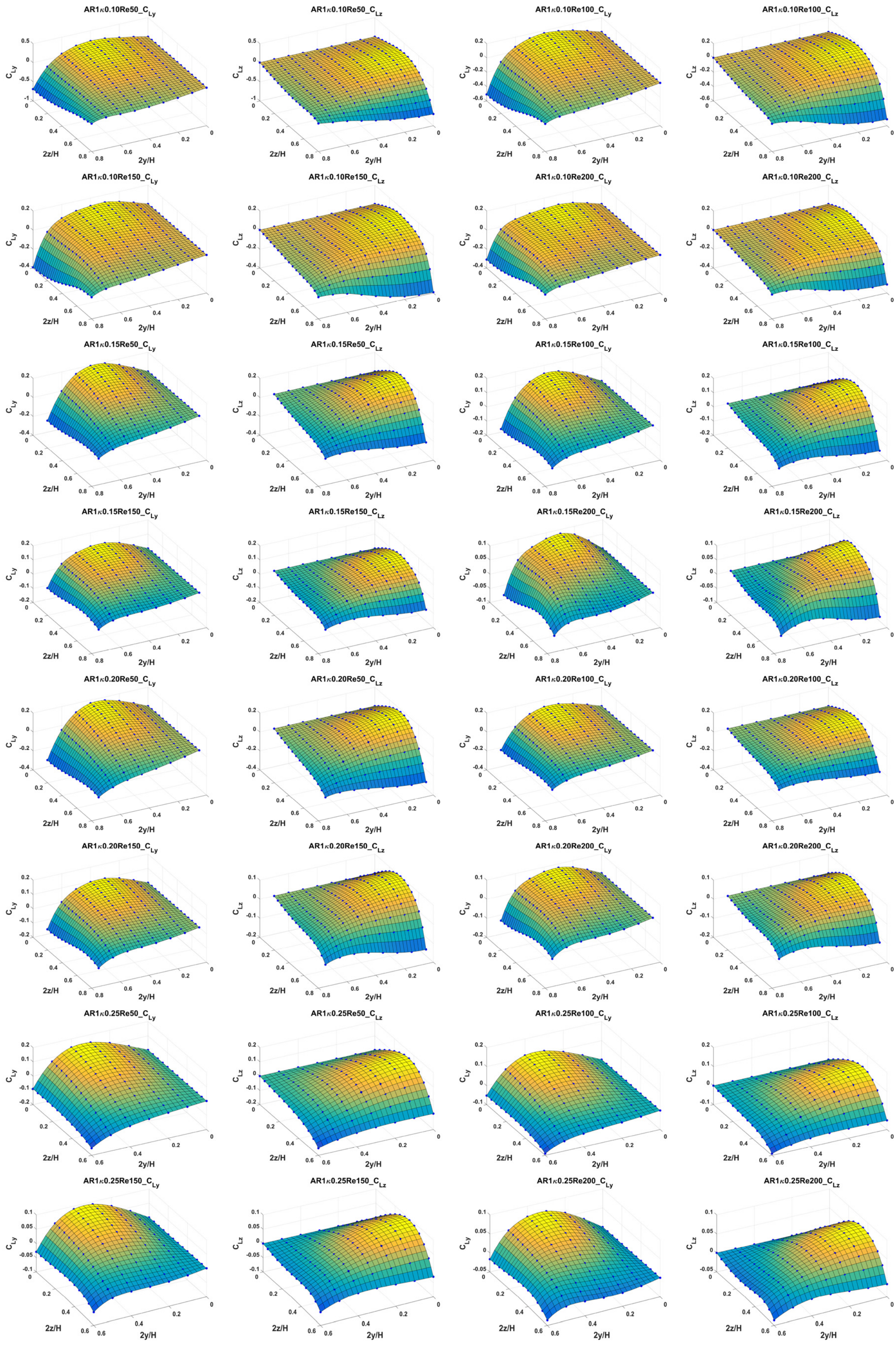
The prediction-files.zip includes the machine learning prediction model file and the inertial lift data file that needs to be learned :

lift_database_rectangular.data is an input data file that contains the distributions of inertial lift forces in the cross-section of straight rectangular microchannels in a wide parameter space. A total of 8068 cases obtained by direct numerical simulations are included in this data file.

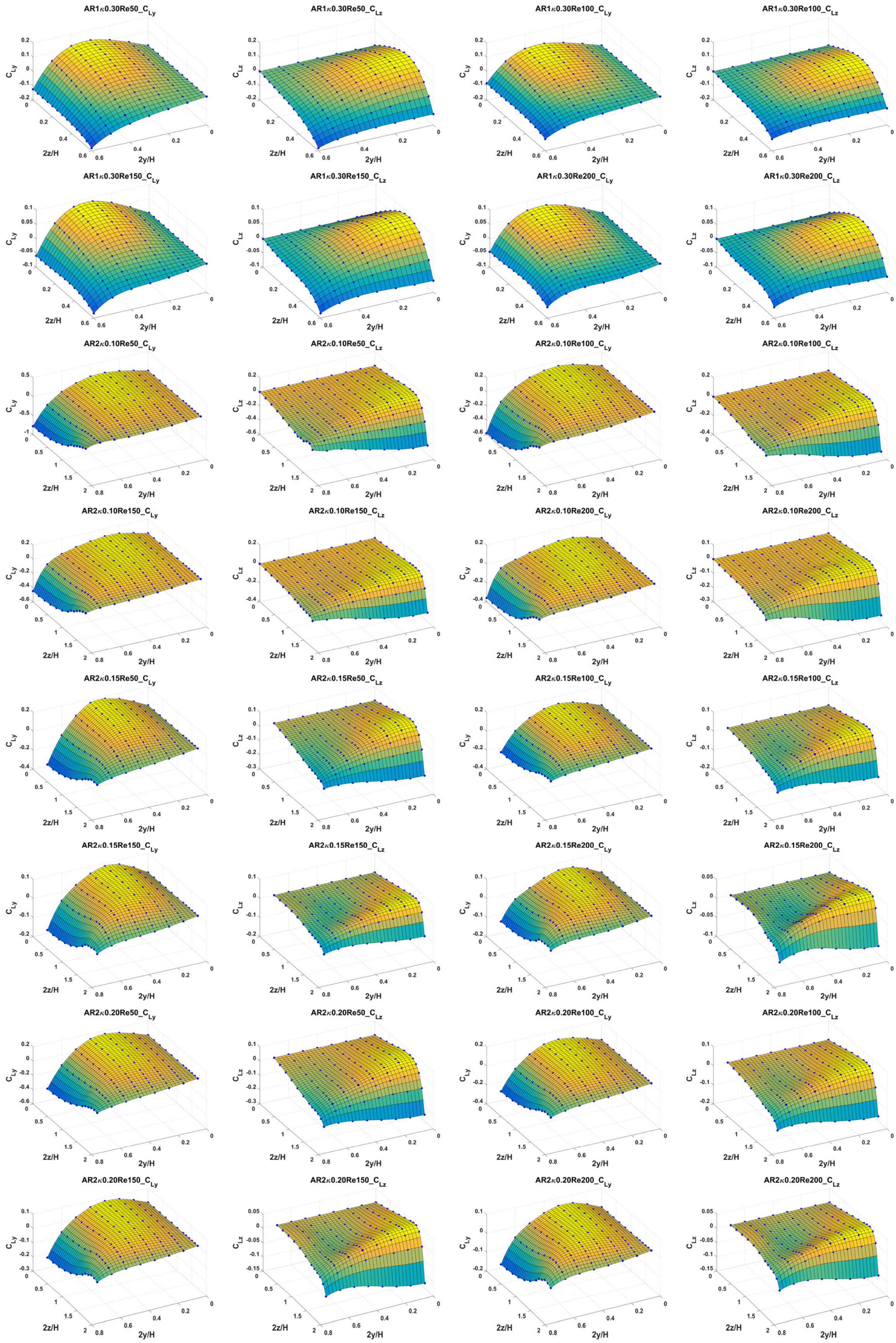
lift_database_triangular.data is an input data file that contains the distributions of inertial lift forces in the cross-section of a straight triangular microchannel in a wide parameter space. A total of 2420 cases obtained by direct numerical simulations are included in this data file.

lift_database_semicircular.data is an input data file that contains the distributions of inertial lift forces in the cross-section of a straight semicircular microchannel in a wide parameter space. A total of 3672 cases obtained by direct numerical simulations are included in this data file.

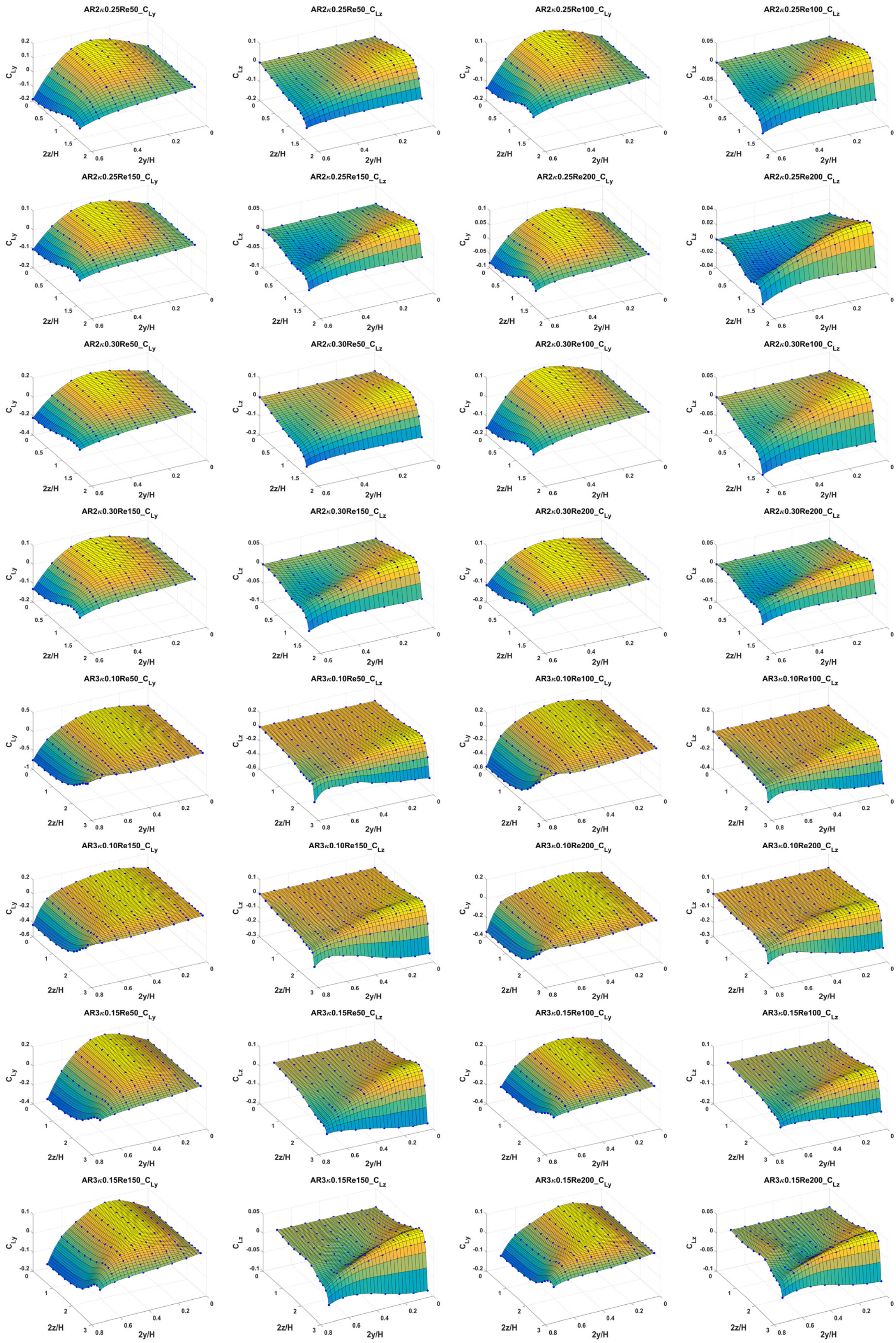
BP_Neural_Network_rectangular.m adopts the error backpropagation neural network (BP neural network) to train and predict the inertial lift in three types of microchannels, including the rectangular microchannel, triangular microchannel, and semicircular microchannel. Users can run the BP_Neural_Network.m by simply specifying the channel shape (The channel aspect ratio is only for the rectangular microchannel), Reynolds number, and particle blockage ratio. The predicted lift forces will be outputted into lift.data for futher use such as Lagrangian particle tracking.


AR1κ0.10Re50_CLy
AR1κ0.10Re50_CLz
AR1κ0.10Re100_CLy
AR1κ0.10Re100_CLz
AR1κ0.10Re150_CLy
AR1κ0.10Re150_CLz
AR1κ0.10Re200_CLy
AR1κ0.10Re200_CLz
AR1κ0.15Re50_CLy
AR1κ0.15Re50_CLz
AR1κ0.15Re100_CLy
AR1κ0.15Re100_CLz
AR1κ0.15Re150_CLy
AR1κ0.15Re150_CLz
AR1κ0.15Re200_CLy
AR1κ0.15Re200_CLz
AR1κ0.20Re50_CLy
AR1κ0.20Re50_CLz
AR1κ0.20Re100_CLy
AR1κ0.20Re100_CLz
AR1κ0.20Re150_CLy
AR1κ0.20Re150_CLz
AR1κ0.20Re200_CLy
AR1κ0.20Re200_CLz
AR1κ0.25Re50_CLy
AR1κ0.25Re50_CLz
AR1κ0.25Re100_CLy
AR1κ0.25Re100_CLz
AR1κ0.25Re150_CLy
AR1κ0.25Re150_CLz
AR1κ0.25Re200_CLy
AR1κ0.25Re200_CLz

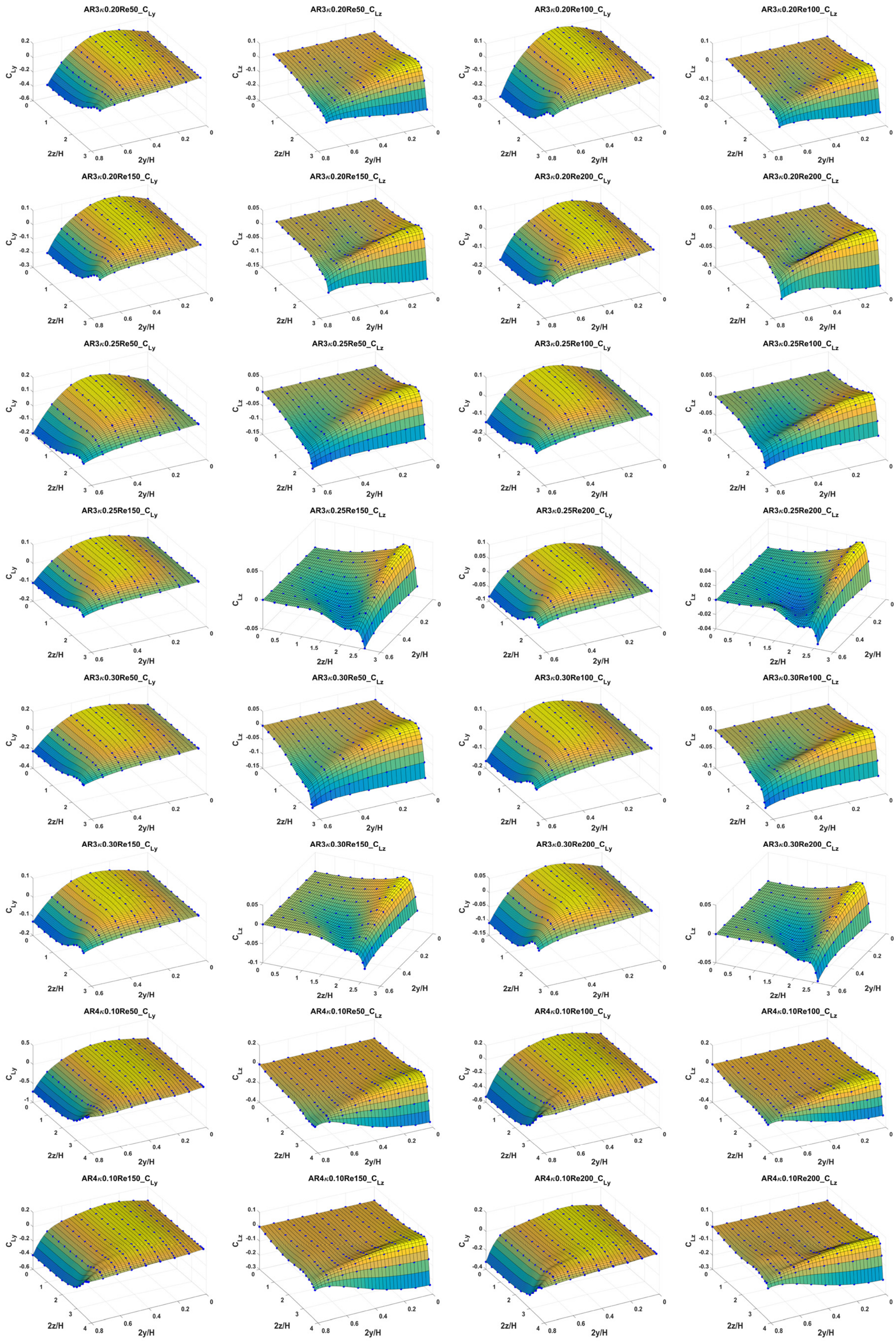
continue


AR1κ0.30Re50_C_{Ly}
AR1κ0.30Re50_C_{Lz}
AR1κ0.30Re100_C_{Ly}
AR1κ0.30Re100_C_{Lz}
AR1κ0.30Re150_C_{Ly}
AR1κ0.30Re150_C_{Lz}
AR1κ0.30Re200_C_{Ly}
AR1κ0.30Re200_C_{Lz}
AR2κ0.10Re50_C_{Ly}
AR2κ0.10Re50_C_{Lz}
AR2κ0.10Re100_C_{Ly}
AR2κ0.10Re100_C_{Lz}
AR2κ0.10Re150_C_{Ly}
AR2κ0.10Re150_C_{Lz}
AR2κ0.10Re200_C_{Ly}
AR2κ0.10Re200_C_{Lz}
AR2κ0.15Re50_C_{Ly}
AR2κ0.15Re50_C_{Lz}
AR2κ0.15Re100_C_{Ly}
AR2κ0.15Re100_C_{Lz}
AR2κ0.15Re150_C_{Ly}
AR2κ0.15Re150_C_{Lz}
AR2κ0.15Re200_C_{Ly}
AR2κ0.15Re200_C_{Lz}
AR2κ0.20Re50_C_{Ly}
AR2κ0.20Re50_C_{Lz}
AR2κ0.20Re100_C_{Ly}
AR2κ0.20Re100_C_{Lz}
AR2κ0.20Re150_C_{Ly}
AR2κ0.20Re150_C_{Lz}
AR2κ0.20Re200_C_{Ly}
AR2κ0.20Re200_C_{Lz}
2z/H
2y/H

continue


AR2κ0.25Re50_CLy
AR2κ0.25Re50_CLz
AR2κ0.25Re100_CLy
AR2κ0.25Re100_CLz
AR2κ0.25Re150_CLy
AR2κ0.25Re150_CLz
AR2κ0.25Re200_CLy
AR2κ0.25Re200_CLz
AR2κ0.30Re50_CLy
AR2κ0.30Re50_CLz
AR2κ0.30Re100_CLy
AR2κ0.30Re100_CLz
AR2κ0.30Re150_CLy
AR2κ0.30Re150_CLz
AR2κ0.30Re200_CLy
AR2κ0.30Re200_CLz
AR3κ0.10Re50_CLy
AR3κ0.10Re50_CLz
AR3κ0.10Re100_CLy
AR3κ0.10Re100_CLz
AR3κ0.10Re150_CLy
AR3κ0.10Re150_CLz
AR3κ0.10Re200_CLy
AR3κ0.10Re200_CLz
AR3κ0.15Re50_CLy
AR3κ0.15Re50_CLz
AR3κ0.15Re100_CLy
AR3κ0.15Re100_CLz
AR3κ0.15Re150_CLy
AR3κ0.15Re150_CLz
AR3κ0.15Re200_CLy
AR3κ0.15Re200_CLz
2z/H
2y/H

continue


AR3κ0.20Re50_C_Ly
AR3κ0.20Re50_C_Lz
AR3κ0.20Re100_C_Ly
AR3κ0.20Re100_C_Lz
AR3κ0.20Re150_C_Ly
AR3κ0.20Re150_C_Lz
AR3κ0.20Re200_C_Ly
AR3κ0.20Re200_C_Lz
AR3κ0.25Re50_C_Ly
AR3κ0.25Re50_C_Lz
AR3κ0.25Re100_C_Ly
AR3κ0.25Re100_C_Lz
AR3κ0.25Re150_C_Ly
AR3κ0.25Re150_C_Lz
AR3κ0.25Re200_C_Ly
AR3κ0.25Re200_C_Lz
AR3κ0.30Re50_C_Ly
AR3κ0.30Re50_C_Lz
AR3κ0.30Re100_C_Ly
AR3κ0.30Re100_C_Lz
AR3κ0.30Re150_C_Ly
AR3κ0.30Re150_C_Lz
AR3κ0.30Re200_C_Ly
AR3κ0.30Re200_C_Lz
AR4κ0.10Re50_C_Ly
AR4κ0.10Re50_C_Lz
AR4κ0.10Re100_C_Ly
AR4κ0.10Re100_C_Lz
AR4κ0.10Re150_C_Ly
AR4κ0.10Re150_C_Lz
AR4κ0.10Re200_C_Ly
AR4κ0.10Re200_C_Lz
2z/H
2y/H

continue

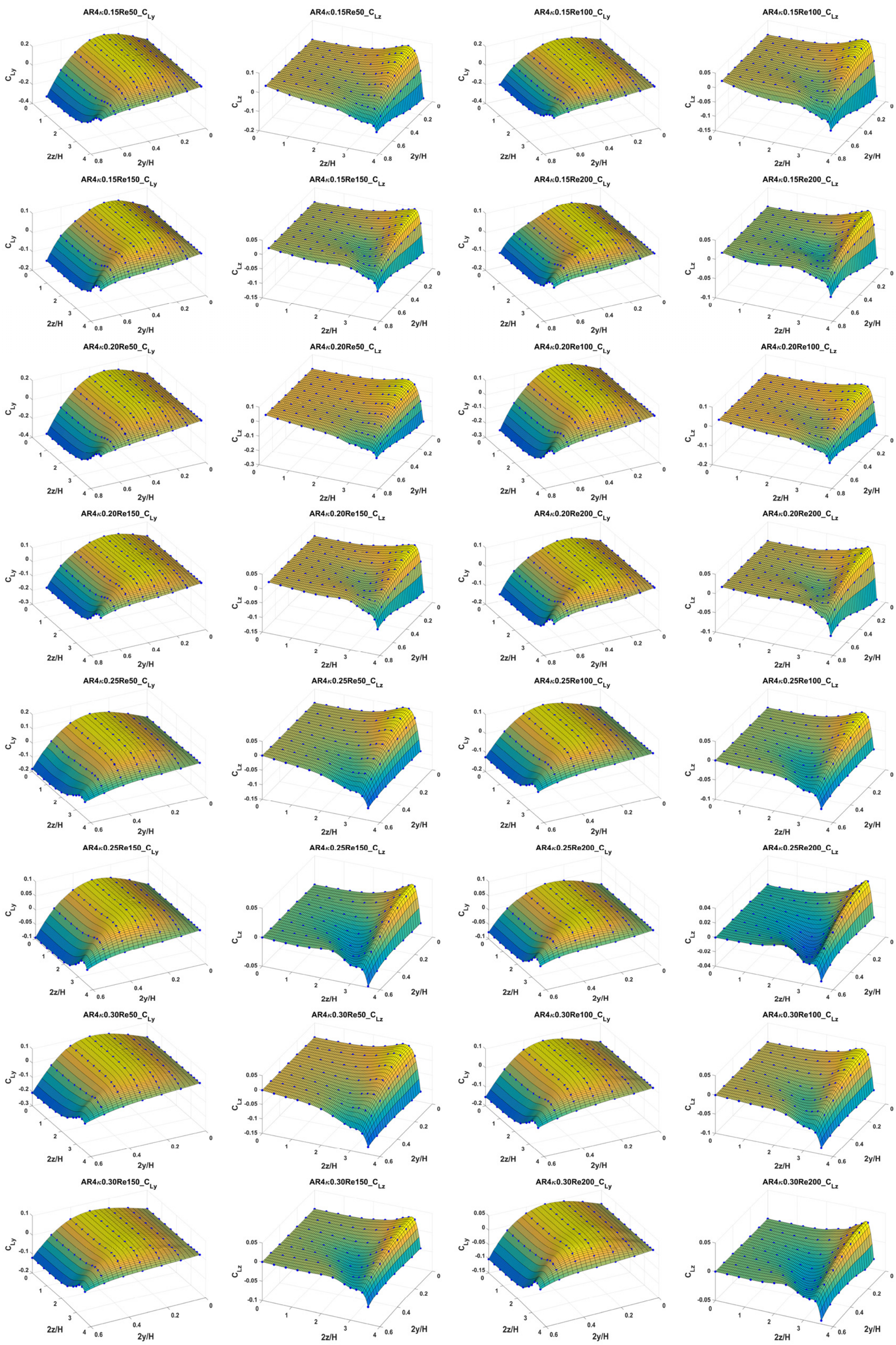


Figure S1. Distributions of the inertial lift coefficient components in quarter of channel cross-sections with the lift data in the database marked by blue dots.

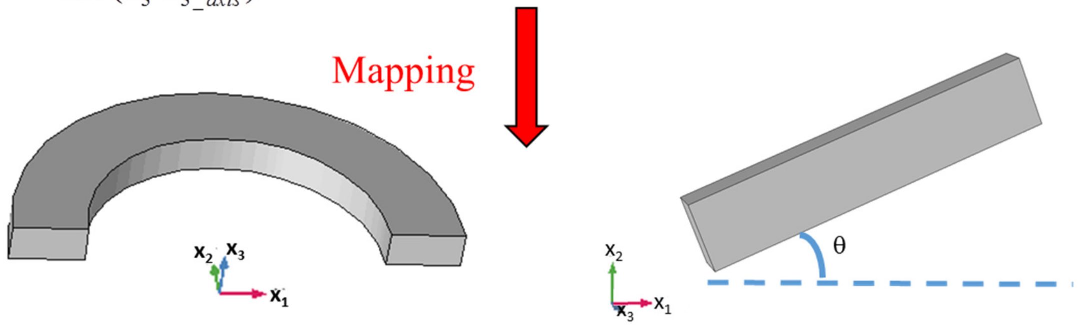
We here set the coordinate of system as (x_1, x_2, x_3), the direction of x_3 as the direction of height and the coordinate of channel center axis as (x_{1_axis}, x_{2_axis} , x_{3_axis}). In each x_1 - x_2 plane, the radius of medium curvature is R and the center of curvature is (x_{1c}, x_{2c}). We can map the inertial lift (Fx_1, Fx_2, Fx_3) by following method.

x_3

$$Fx_1 = \frac{(x_1\text{-}x_{1\_axis})}{abs(Rn)} \cdot C_{Lz}(abs(2(x_3\text{-}x_{3\_axis})/H), abs(2Rn/H)) \cdot \rho U_{\max}^2 a^2 \kappa^2 \quad \text{(b1)}$$

$$Fx_2 = \frac{(x_2\text{-}x_{2\_axis})}{abs(Rn)} \cdot C_{Lz}(abs(2(x_3\text{-}x_{3\_axis})/H), abs(2Rn/H)) \cdot \rho U_{\max}^2 a^2 \kappa^2 \quad \text{(b2)}$$

$$Fx_3 = \frac{x_3\text{-}x_{3\_axis}}{abs(x_3\text{-}x_{3\_axis})} \cdot C_{Ly}(abs(2(x_3\text{-}x_{3\_axis})/H), abs(2Rn/H)) \cdot \rho U_{\max}^2 a^2 \kappa^2 \quad \text{(b3)}$$



For curved channel

$$DTocenter = ((x_1\text{-}x_{1c})^2 + (x_2\text{-}x_{2c})^2)^{0.5} \quad \text{(c1)}$$

$$Rn = DTocenter\text{-}R \quad \text{(c2)}$$

$$\frac{(x_1 - x_{1\_axis})}{abs(Rn)} = sign(2Rn/H) \cdot \frac{(x_1 - x_{1c})}{DTocenter} \quad \text{(c3)}$$

$$\frac{(x_2 - x_{2\_axis})}{abs(Rn)} = sign(2Rn/H) \cdot \frac{(x_2 - x_{2c})}{DTocenter} \quad \text{(c4)}$$

$$x_{3\_axis} = \text{constant value} \quad \text{(c5)}$$

For straight channel

$$abs(Rn) = ((x_1\text{-}x_{1\_axis})^2 + (x_2\text{-}x_{2\_axis})^2)^{0.5} \quad \text{(s1)}$$

$$\frac{(x_1 - x_{1\_axis})}{abs(Rn)} = sign(x_1\text{-}x_{1\_axis}) \cdot abs(\sin\theta) \quad \text{(s2)}$$

$$\frac{(x_2 - x_{2\_axis})}{abs(Rn)} = sign(x_2\text{-}x_{2\_axis}) \cdot abs(\cos\theta) \quad \text{(s3)}$$

$$x_{3\_axis} = \text{constant value} \quad \text{(s4)}$$

Figure S2. Mapping method for two types of channels: curved channel with constant radius and straight channel with a tilt angle varying from 0 to π.

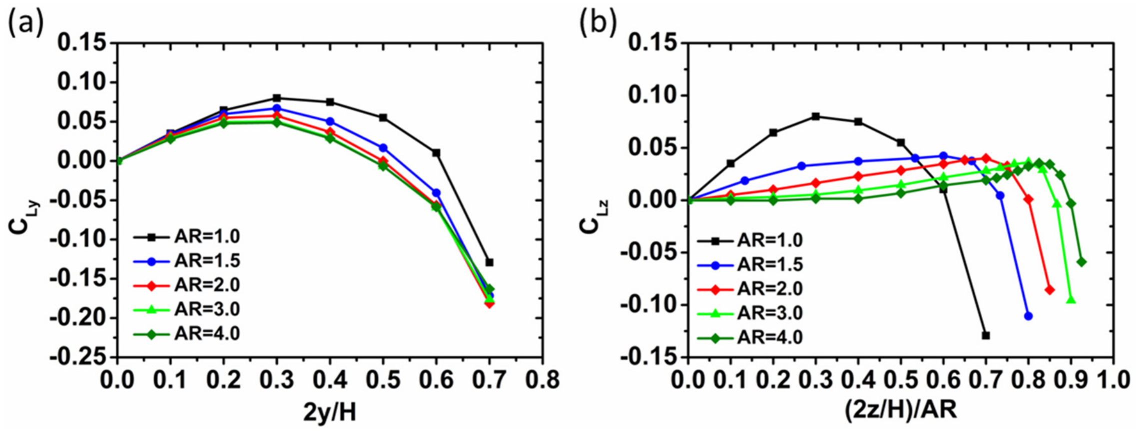


Figure S3. C_{Ly} along the short axis (a) and C_{Lz} along the long axis (b) with Re = 200 and κ = 0.2 at different *AR*.

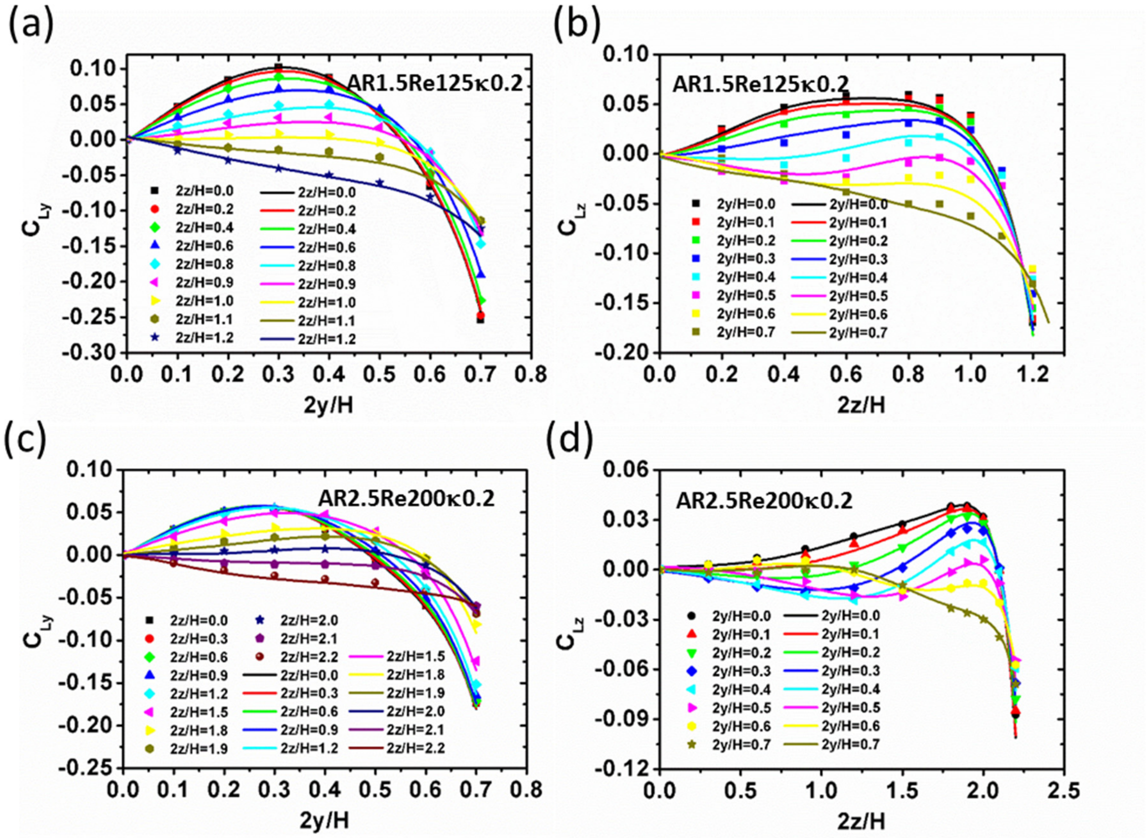


Figure S4. Comparisons between the inertial lift forces from the direct numerical simulations (symbols) and the ANN predictions (lines).