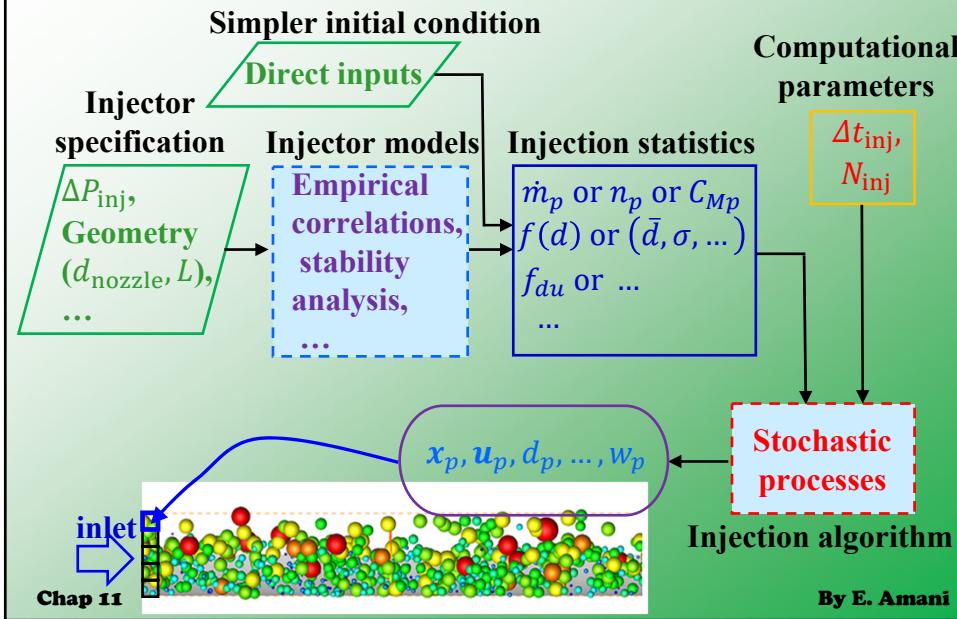


Initial condition – injection models

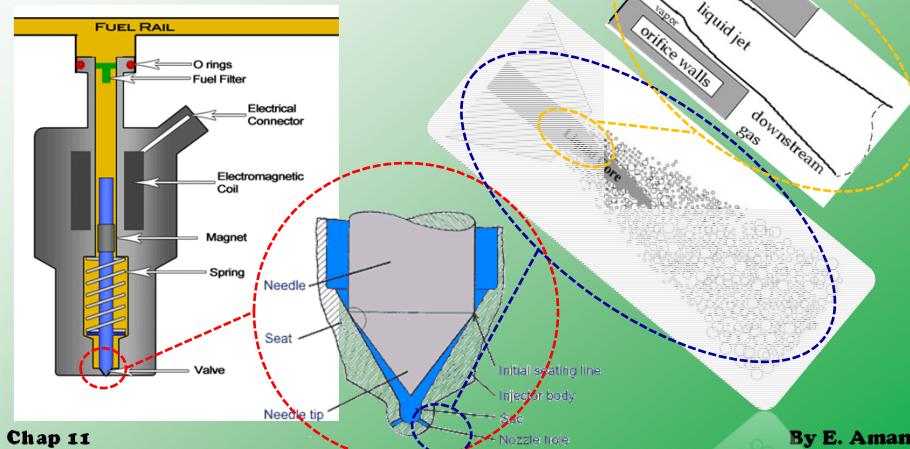
- Importance:
 - accurate particle evolution calculation
 - + inaccurate initial condition
 - = ?!!!
- Goal:
 - Determining the state vector for each parcel
$$(x_p, \mathbf{u}_p, d_p, \dots, \mathbf{u}_s, \dots, w_p)$$
- Injection model:
 - A part or the whole of the following flowchart

Injection modeling hierarchy



Injector models

- For sprays: called “atomizer models”, or “primary breakup models”, or ...
 - (Plain-)orifice atomizers



Injector models

- For sprays: called “atomizer models”, or “ primary breakup models”, or ...
 - (Plain-)orifice atomizers



Chap 11



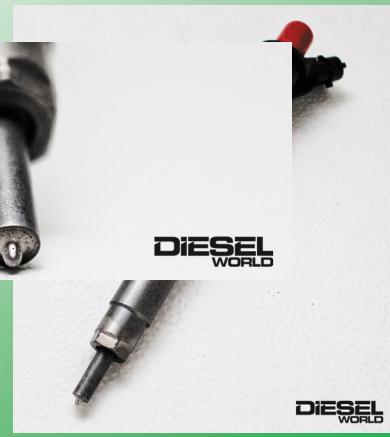
By E. Amani

Injector models

- For sprays: called “atomizer models”, or “ primary breakup models”, or ...
 - (Plain-)orifice atomizers



Chap 11



By E. Amani

Injector models

- For sprays: called “atomizer models”, or “primary breakup models”, or ...
 - (Plain-)orifice atomizers



Chap 11

By E. Amani

Injector models

- For sprays: called “atomizer models”, or “primary breakup models”, or ...
 - (Plain-)orifice atomizers (non-cavitating mode) [1]

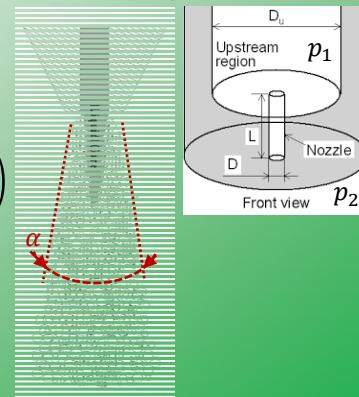
$$V_{\text{inj}} = c_v \sqrt{\frac{2\Delta p}{\rho_l}}$$

$$\frac{\alpha}{2} = \tan^{-1} \left[\frac{4\pi}{A} \left(\frac{\rho_g}{\rho_l} \right)^{0.5} \frac{\sqrt{3}}{6} \right], A = 3 + 0.278 \left(\frac{L}{d} \right)$$

$$d_{32} = 133.0 \left(\frac{d}{8} \right) \left(\frac{\text{We}_l}{8} \right)^{-0.74}, \text{We}_l = \frac{\rho_l V_{\text{inj}}^2 d}{\sigma}$$

(Rosin-Rammler, $n = 3.5$)

$$\dot{m}_{\text{nozzle}} = c_d (A_{\text{nozzle}} \rho_l V_{\text{inj}})$$



Chap 11

By E. Amani

Injector models

- For sprays: called “atomizer models”, or “ primary breakup models”, or ...
 - (Plain-)orifice atomizer
 - Pressure-swirl atomizer
 - ...
 - See HW#11

Chap 11

By E. Amani

Injection algorithm

- Choosing computational parameters:

$$\text{Injection time step} \leftarrow \Delta t_{\text{inj}} = n \Delta t_L \longrightarrow \text{Overall Lagrangian time step}$$

N_{inj} → Number of parcels injected per Δt_{inj}
for a specific injector

$$\left(\sum_{\text{injectors}} N_{\text{inj}} \frac{FTT}{\Delta t_{\text{inj}}} \leq N_{\max} \right)$$

Flow through time
Maximum number of
parcels in domain

Chap 11

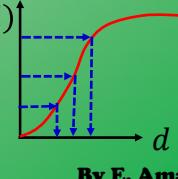
By E. Amani

Injection algorithm

- Choosing computational parameters
- Steps (for each injector):
 1. Generating N_{inj} size samples $d_1, d_2, \dots, d_{N_{\text{inj}}}$ from the size distribution PDF $f(d)$:
 - Specific random generation methods
 - General “inverse transform method”

$$f(d) \longrightarrow Y = F(d) = \int_0^d f(x)dx \xrightarrow{\substack{\text{inverse} \\ \text{Analytical or} \\ \text{numerical (lookup table)}}} d = F^{-1}(Y)$$

$$Y \in U(0,1) \longrightarrow d = F^{-1}(Y)$$



Chap 11

By E. Amani

Injection algorithm

- Choosing computational parameters
- Steps (for each injector):
 1. Generating N_{inj} size samples $d_1, d_2, \dots, d_{N_{\text{inj}}}$ from the size distribution PDF $f(d)$
 2. Setting other particle properties, given proper statistics:

$$\begin{aligned} & x_p \\ & \boldsymbol{u}_p = \langle \boldsymbol{u}_p \rangle + \boldsymbol{u}'_p \end{aligned}$$

...

Chap 11

By E. Amani

Injection algorithm

- Choosing computational parameters
- Steps (for each injector):
 1. Generating N_{inj} size samples $d_1, d_2, \dots, d_{N_{\text{inj}}}$ from the size distribution PDF $f(d)$
 2. Setting other particle properties, given proper statistics
 3. Determining parcel weights:
 - The weights of all samples are equal, otherwise $f(d)$ is violated:
$$w_p = w; p = 1, 2, \dots, N_{\text{inj}}$$
 - w can be computed so as to meet one additional constraint

Chap 11

By E. Amani

Injection algorithm

- Determining w (for each injector):
 - Mass conservation: $w = \frac{(\dot{m}_p'' A_{\text{inj}}) \Delta t_{\text{inj}}}{\sum_{p=1}^{N_{\text{inj}}} m_p}$
 - Number conservation: $w = \frac{(\dot{m}_p'' A_{\text{inj}}) \Delta t_{\text{inj}}}{\frac{\pi}{6} \rho_p \langle d_p^3 \rangle N_{\text{inj}}}$
- If \dot{m}_p'' is not given directly, it can be approximated by:

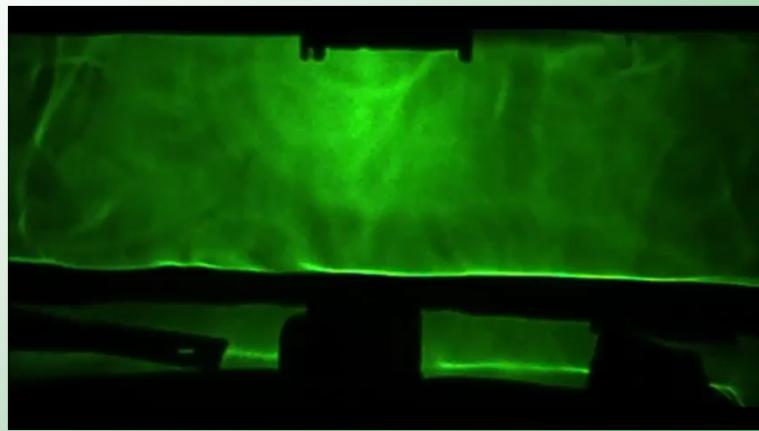
$$\dot{m}_p'' \simeq n_M u_{p,\perp} \simeq \frac{\pi}{6} \rho_p \langle d_p^3 \rangle n u_{p,\perp} \simeq \frac{\pi}{6} \rho_p \langle d_p^3 \rangle \dot{n}_p''$$

number concentration
mass concentration Particle velocity normal to boundary Number flow rate

Chap 11

By E. Amani

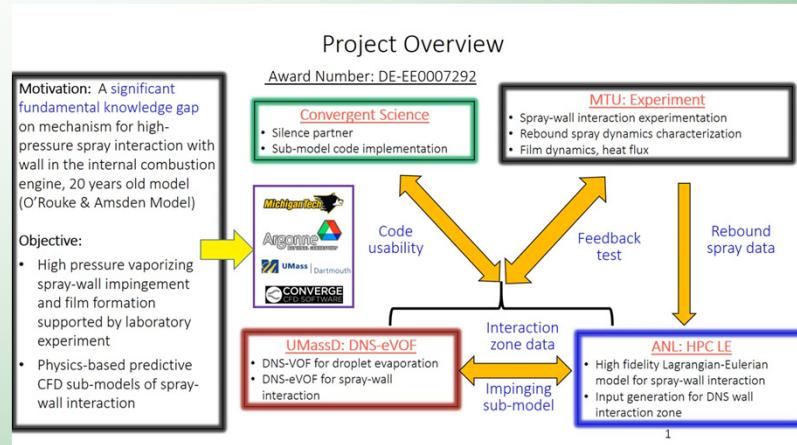
Case study #4: Spray-wall impact



Chap 11

By E. Amani

Case study #4: Spray-wall impact



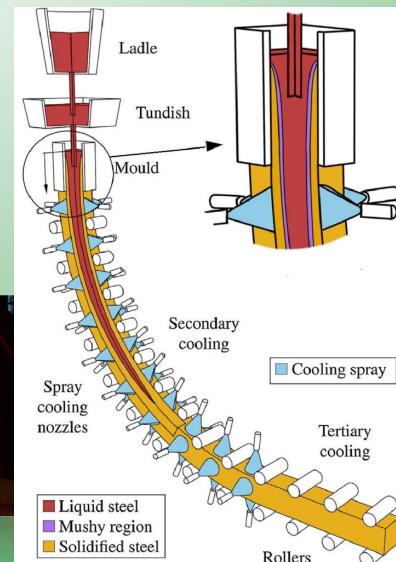
Chap 11

By E. Amani

Case study #4: Spray-wall impact

- Applications:
 - Spray cooling

Continuous casting ▼ ▶



Chap 11

By E. Amani

Case study #4: Spray-wall impact

- Applications:
 - Spray cooling
 - Thermal spray coating



Chap 11

By E. Amani

Case study #4: Spray-wall impact

- Applications:

- Spray cooling
- Thermal spray coating
- Laser metal deposition



Chap 11

By E. Amani

Case study #4: Spray-wall impact

- Applications:

- Spray cooling
- Thermal spray coating
- Laser metal deposition
- Spray painting



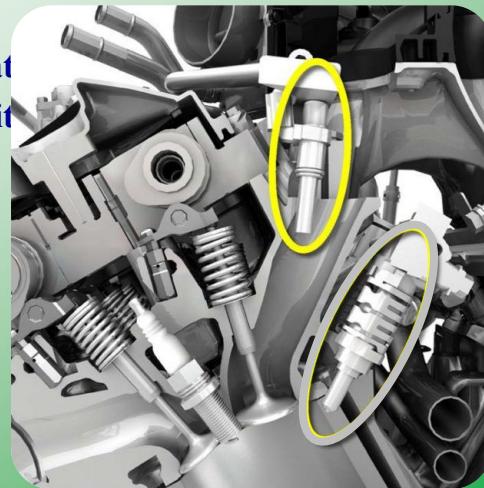
Chap 11

By E. Amani

Case study #4: Spray-wall impact

- Applications:

- Spray cooling
- Thermal spray coating
- Laser metal deposition
- Spray painting
- Port fuel injection
- ...



Chap 11

By E. Amani

Case study #4: Spray-wall impact

- Problem definition:

$$H_d = 10 \text{ cm}$$

$$L_d = 50 \text{ cm}$$

Injector specification:

Type: Plain-orifice injector

Liquid: Water

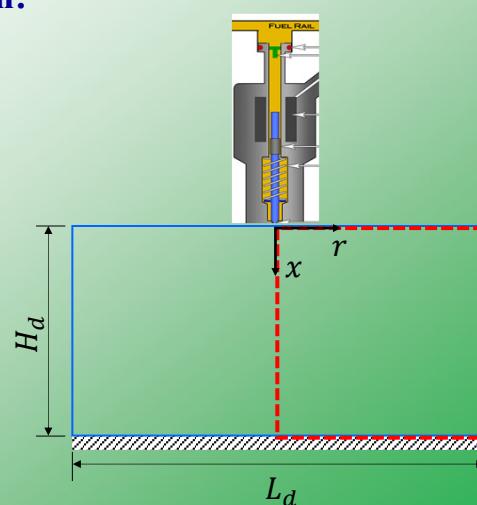
$$D = 500 \mu\text{m}$$

$$L/D = 10$$

$$\Delta p = 1 \text{ MPa}$$

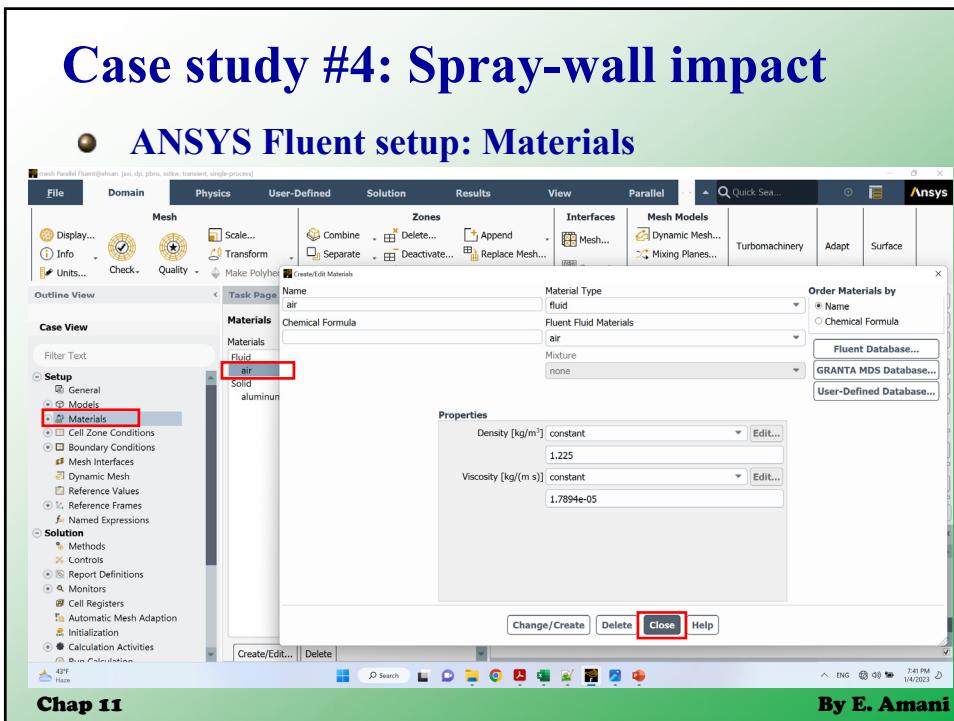
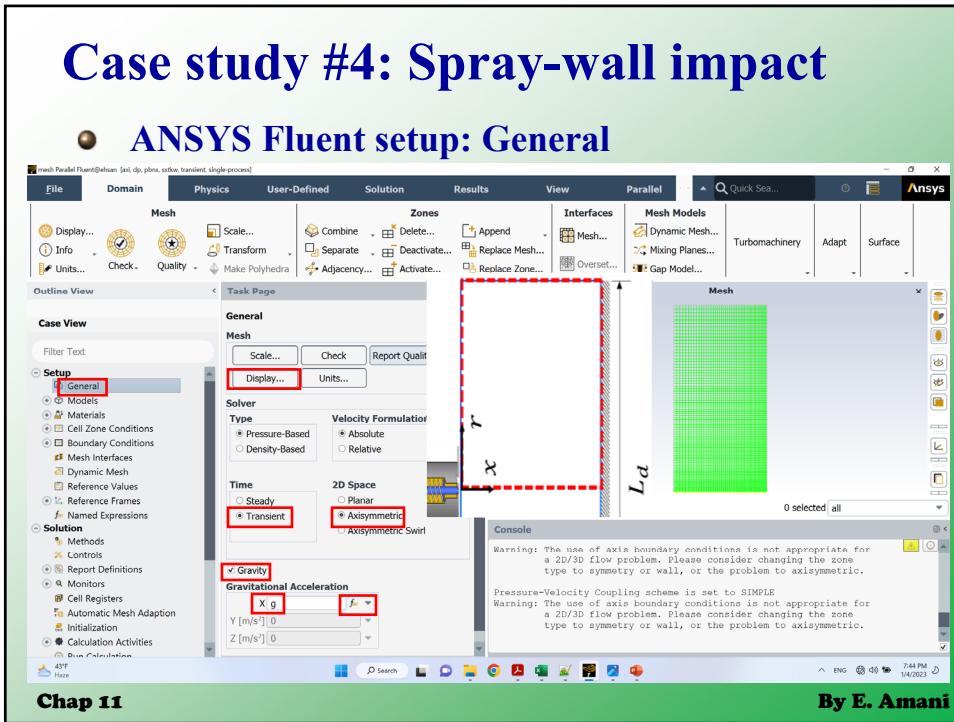
$$c_v = 0.8, c_d = 0.6$$

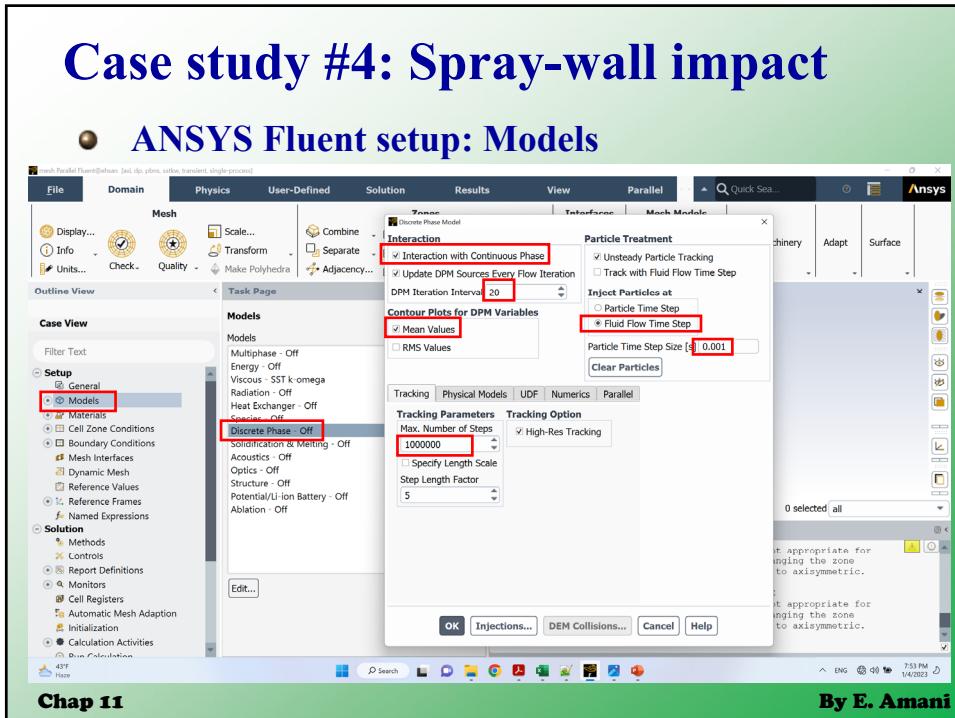
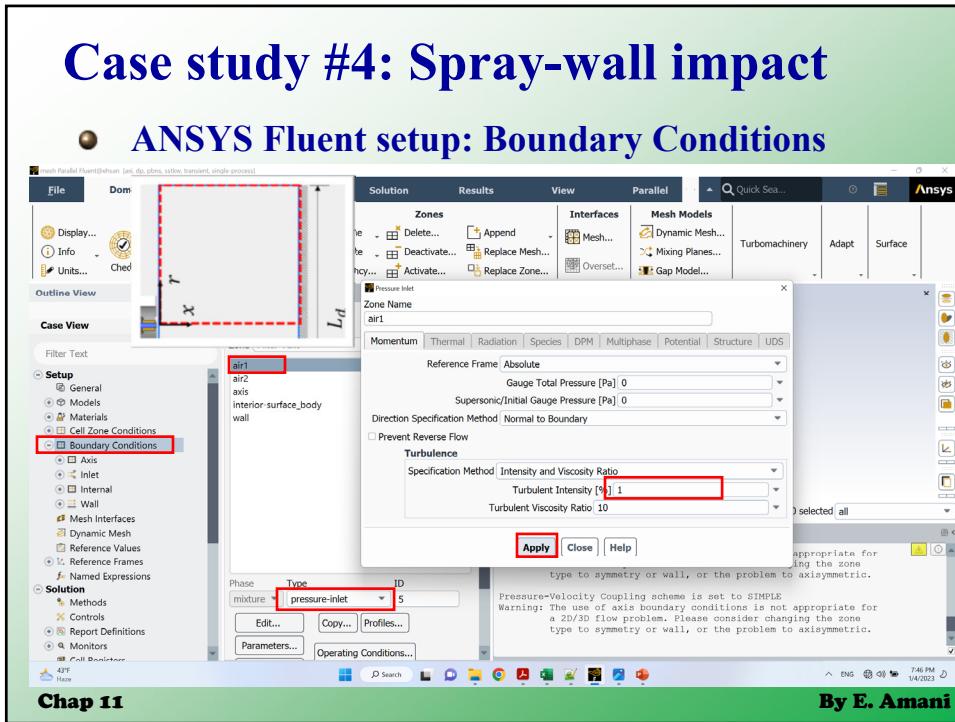
$$\frac{\alpha}{2} = 20 \text{ deg}$$

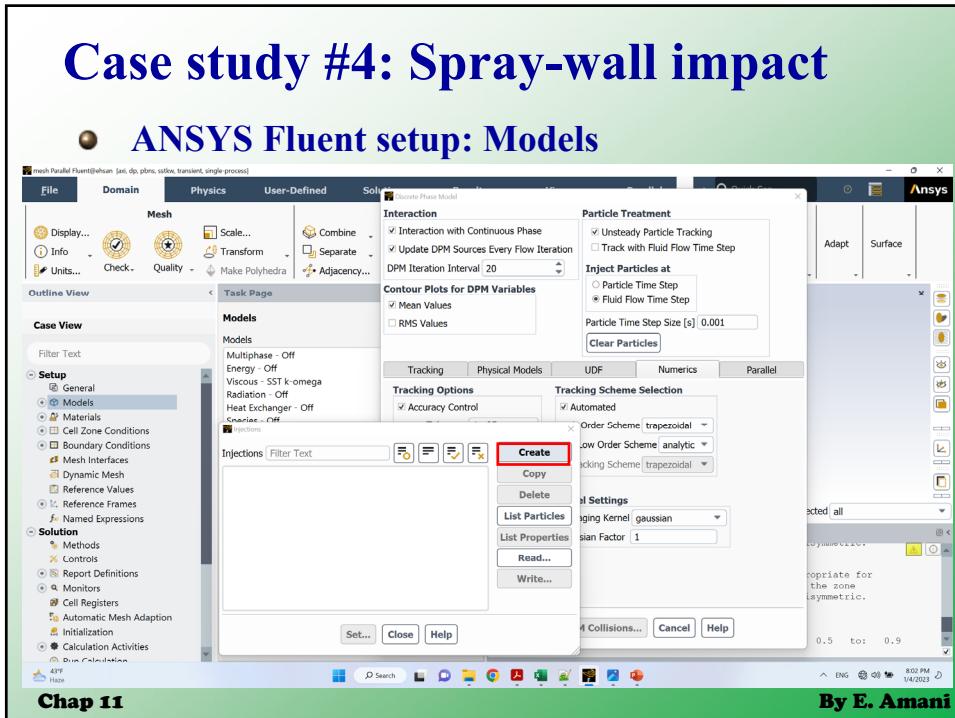
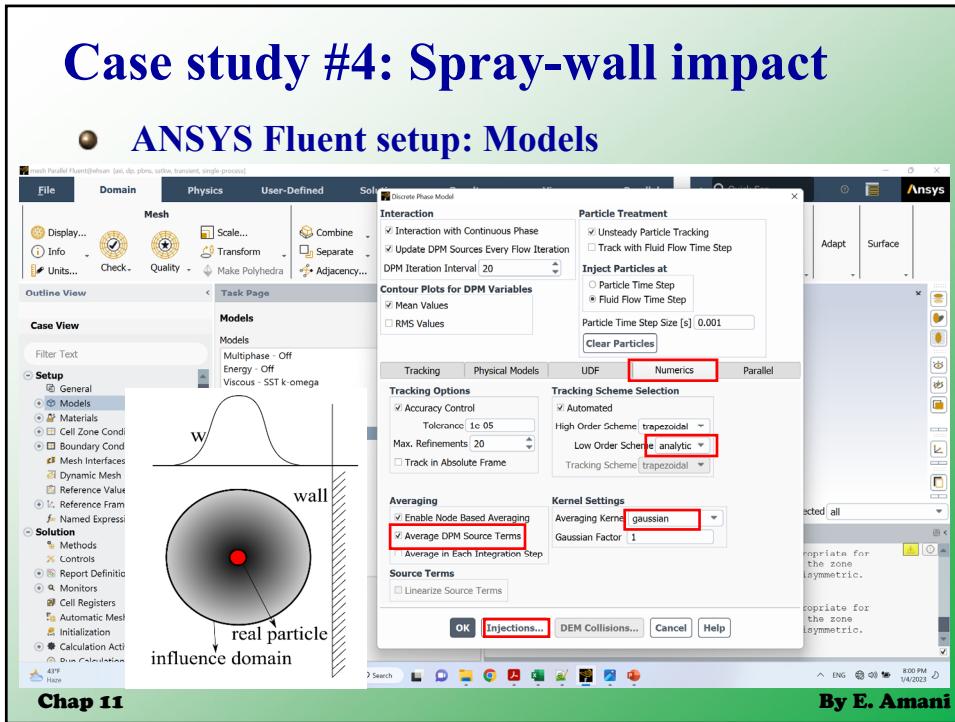


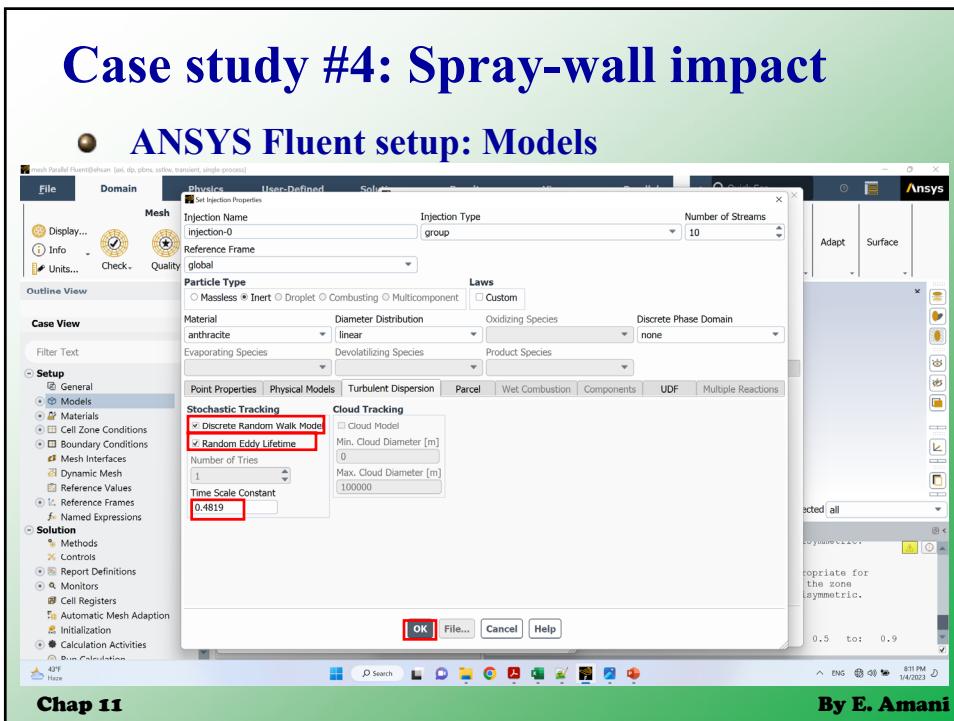
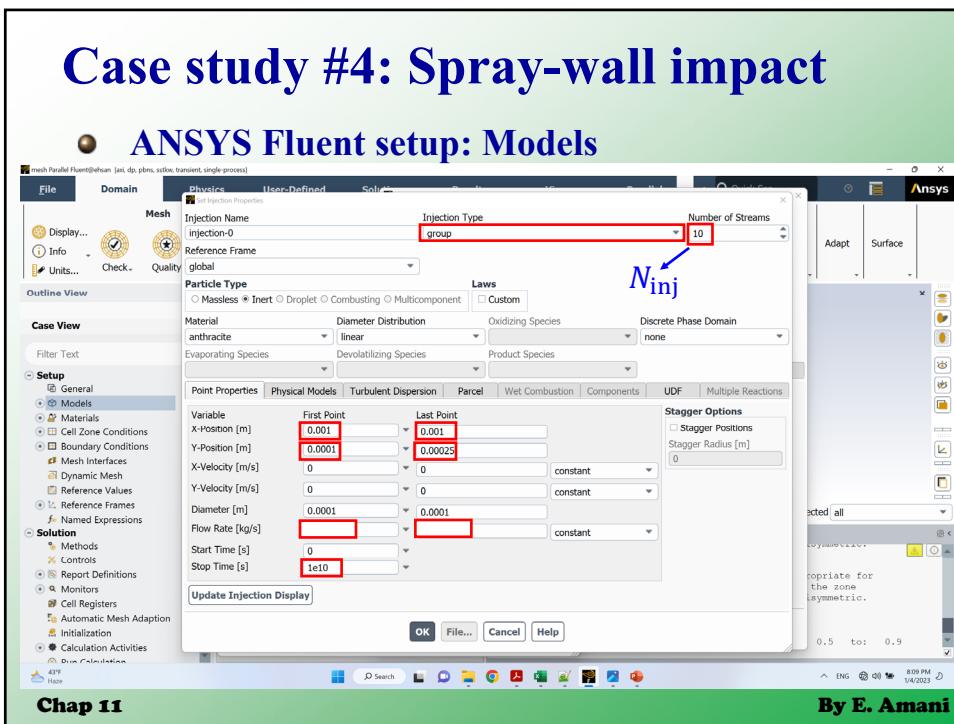
Chap 11

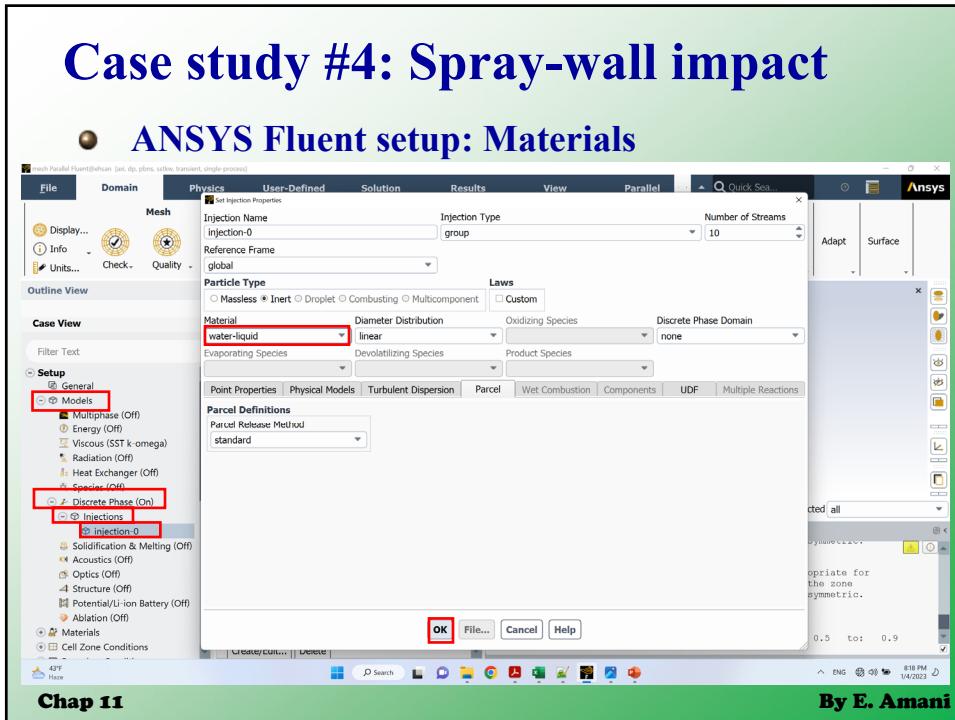
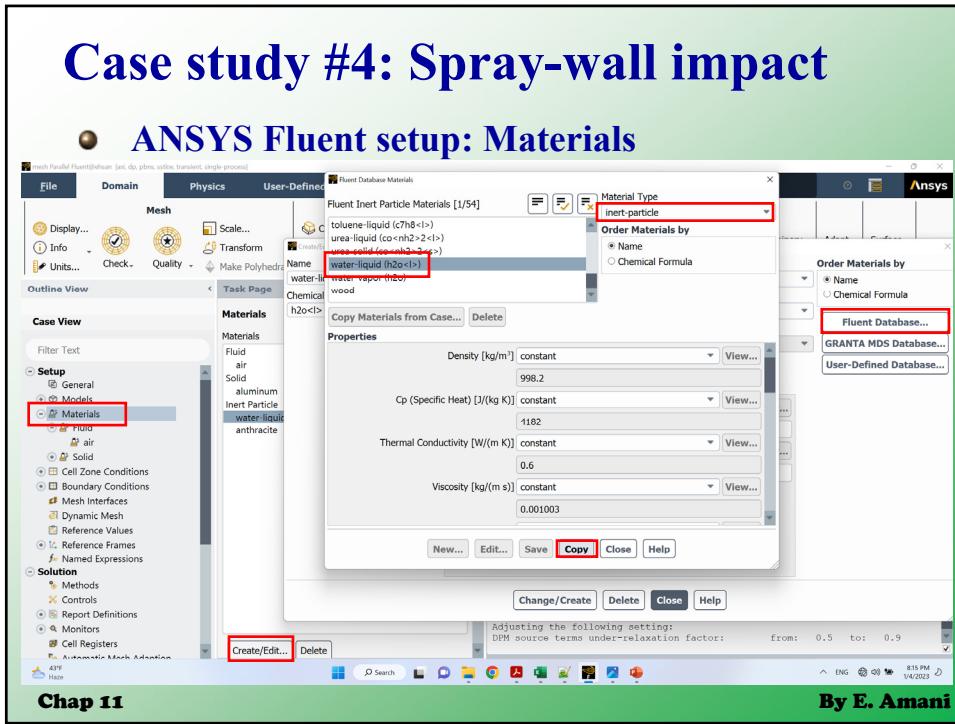
By E. Amani

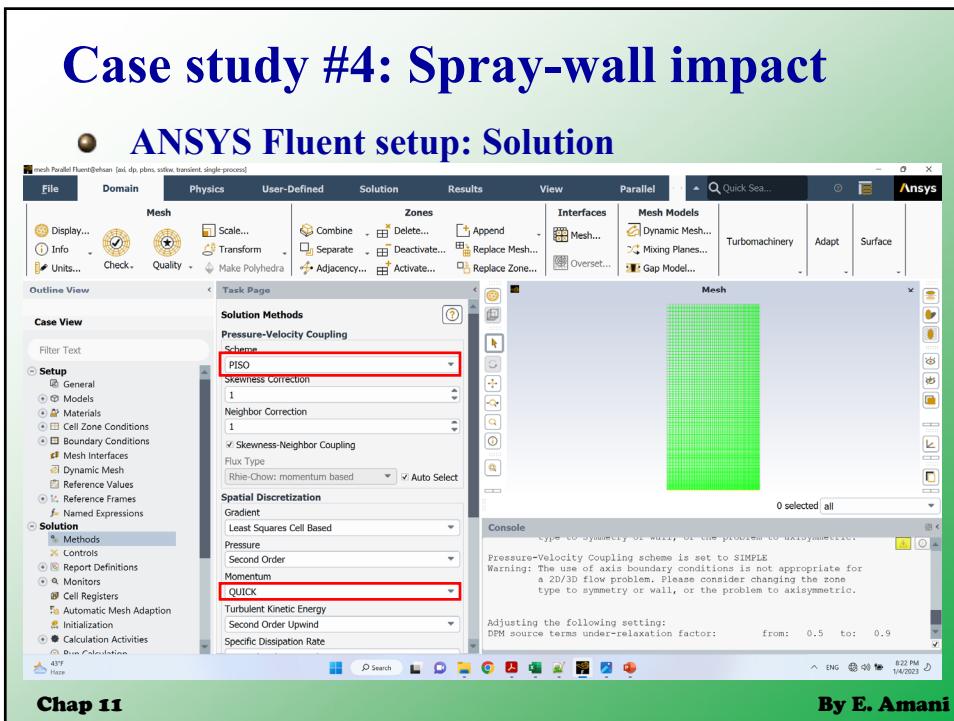
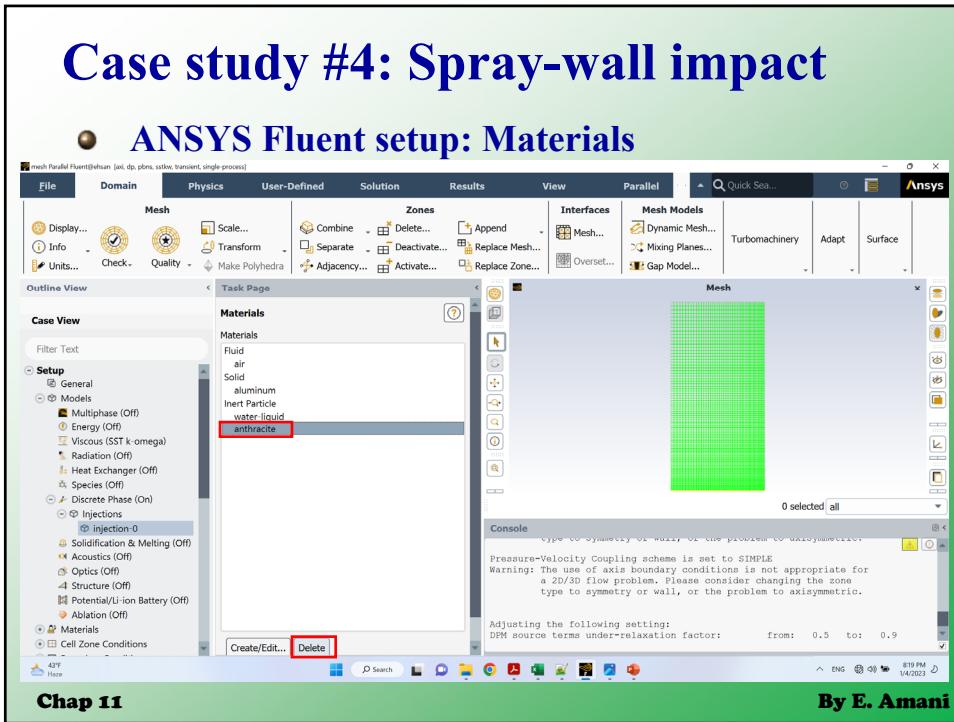


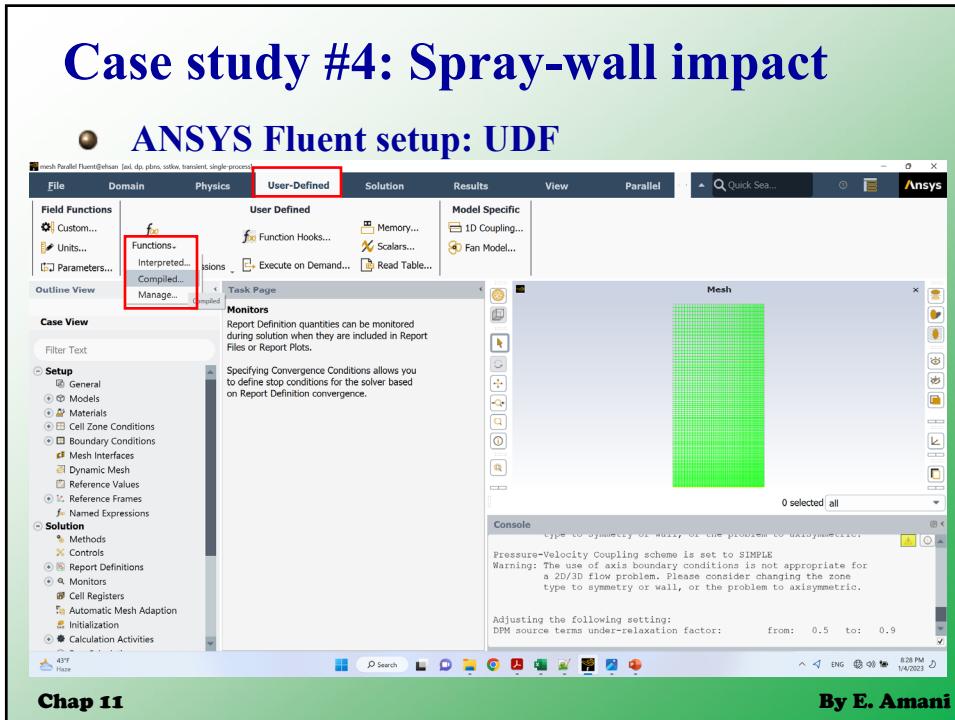
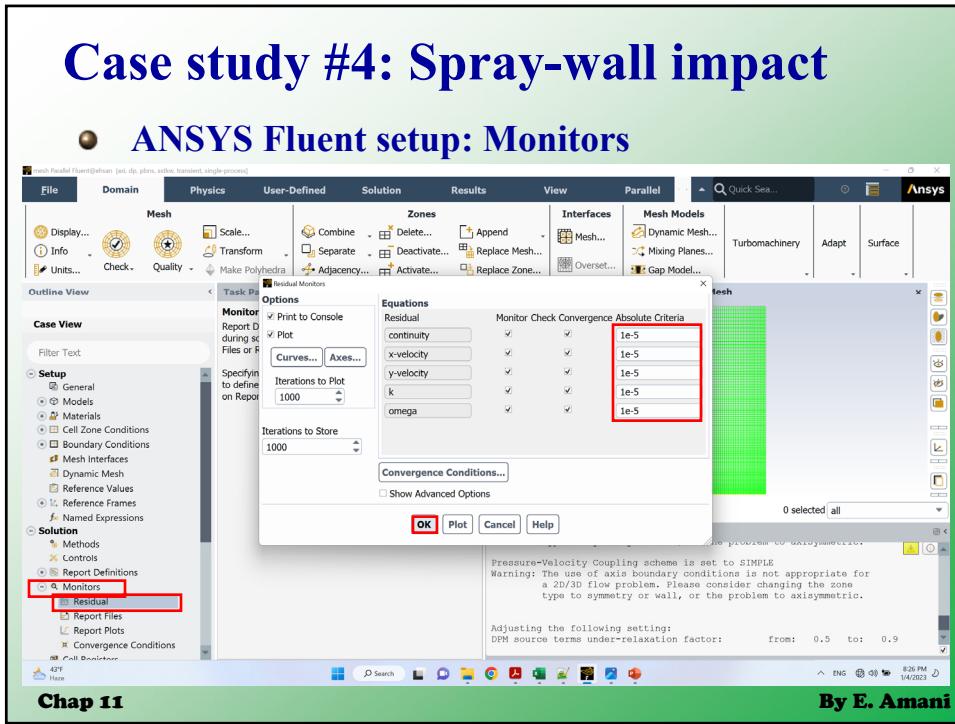


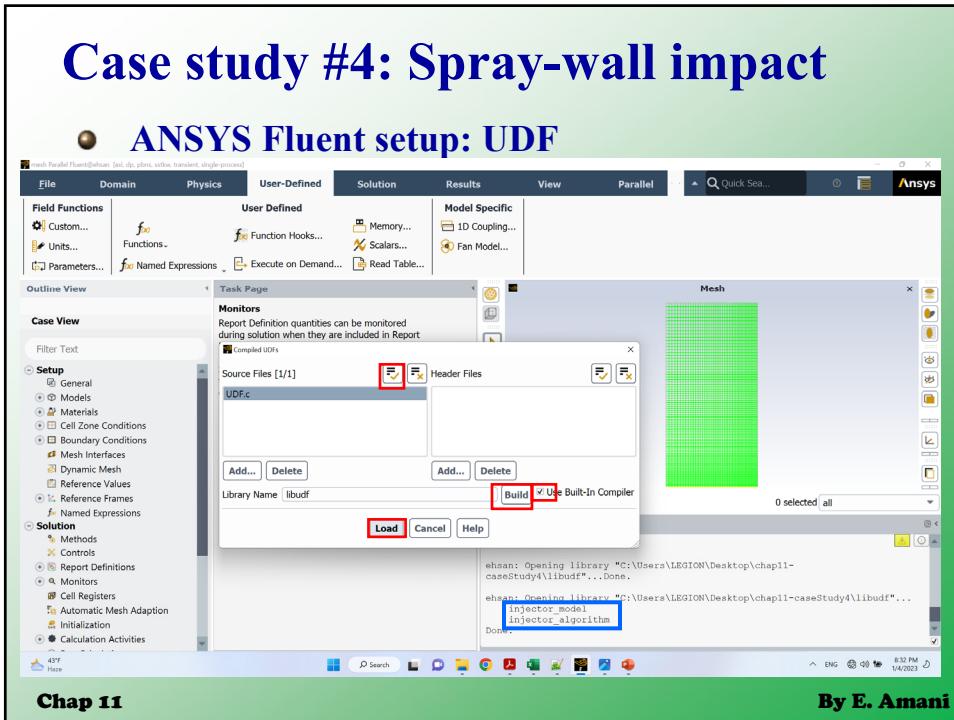
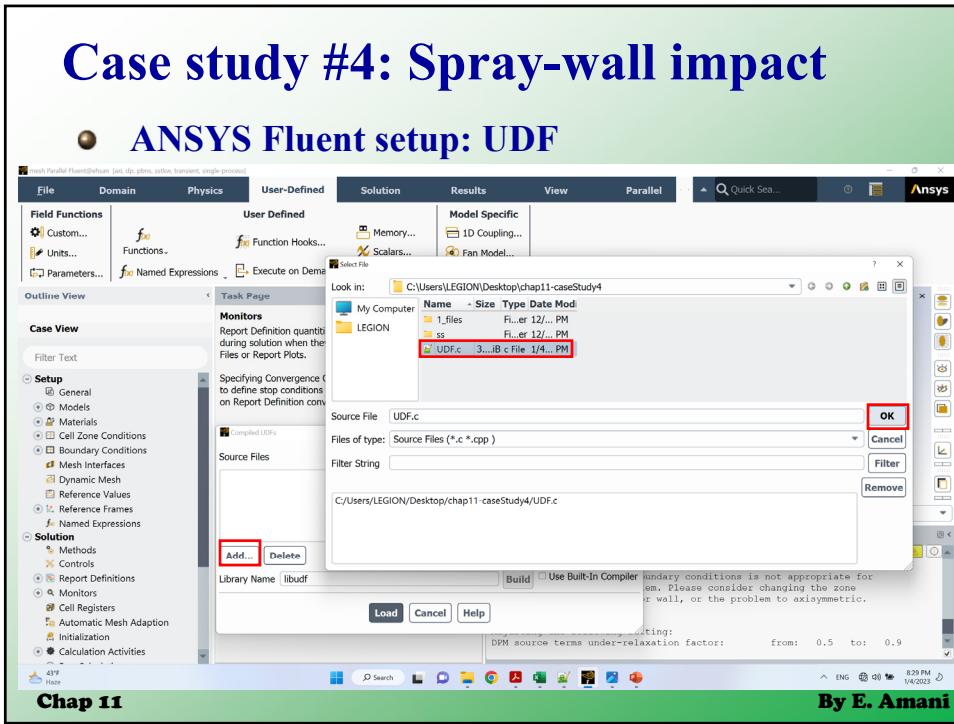




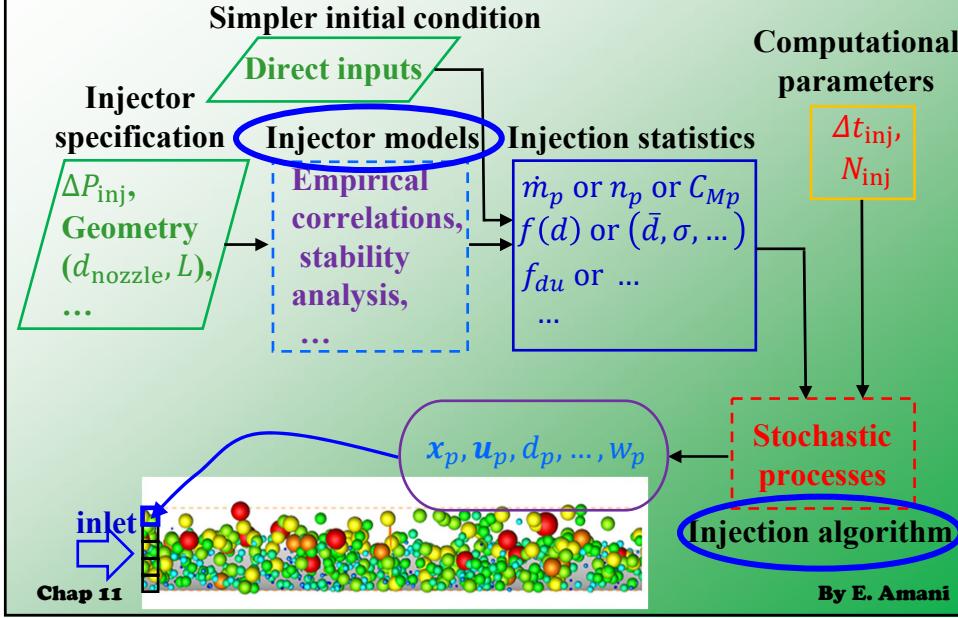






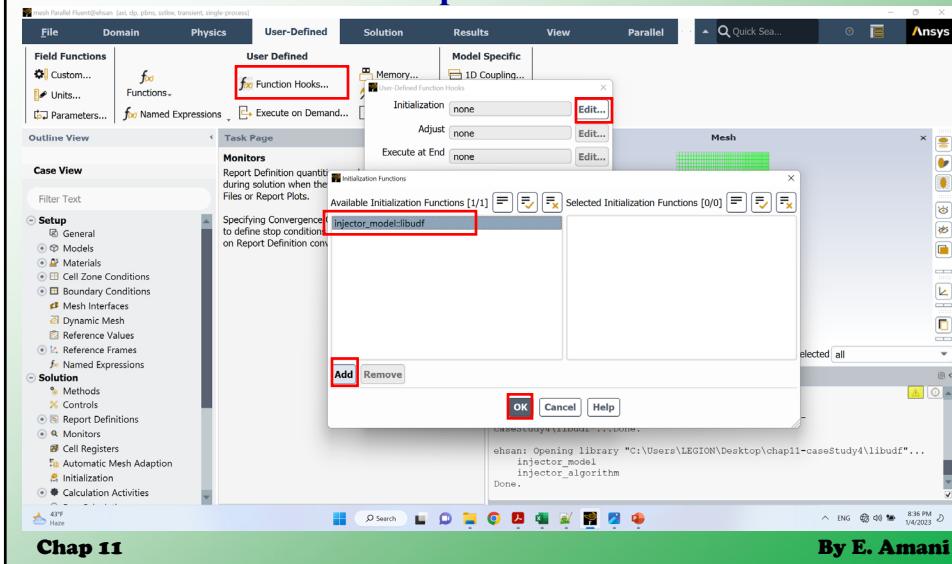


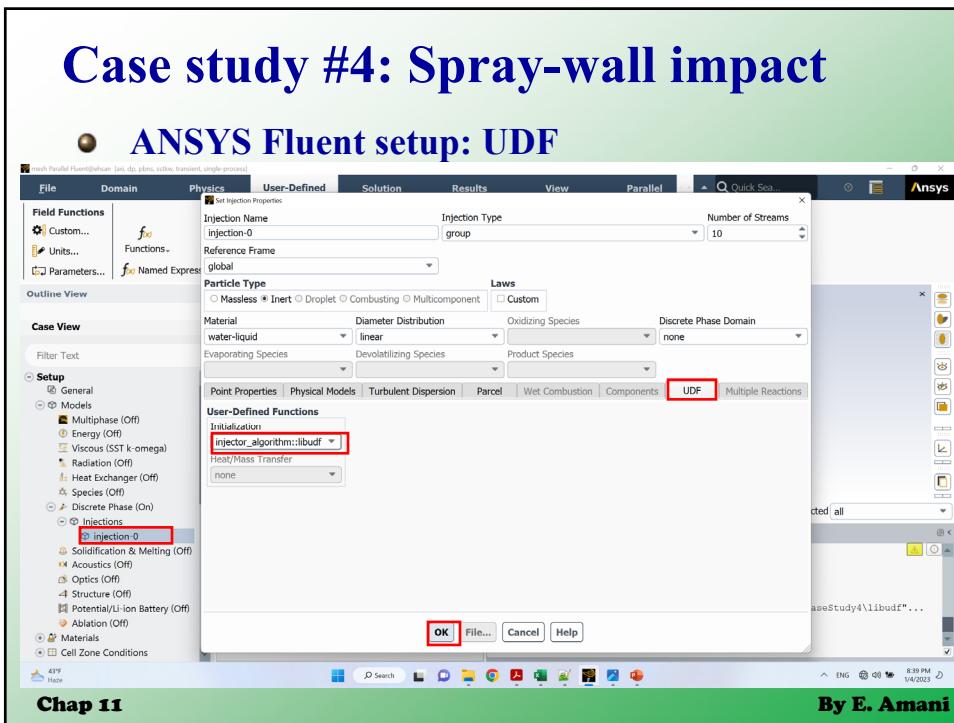
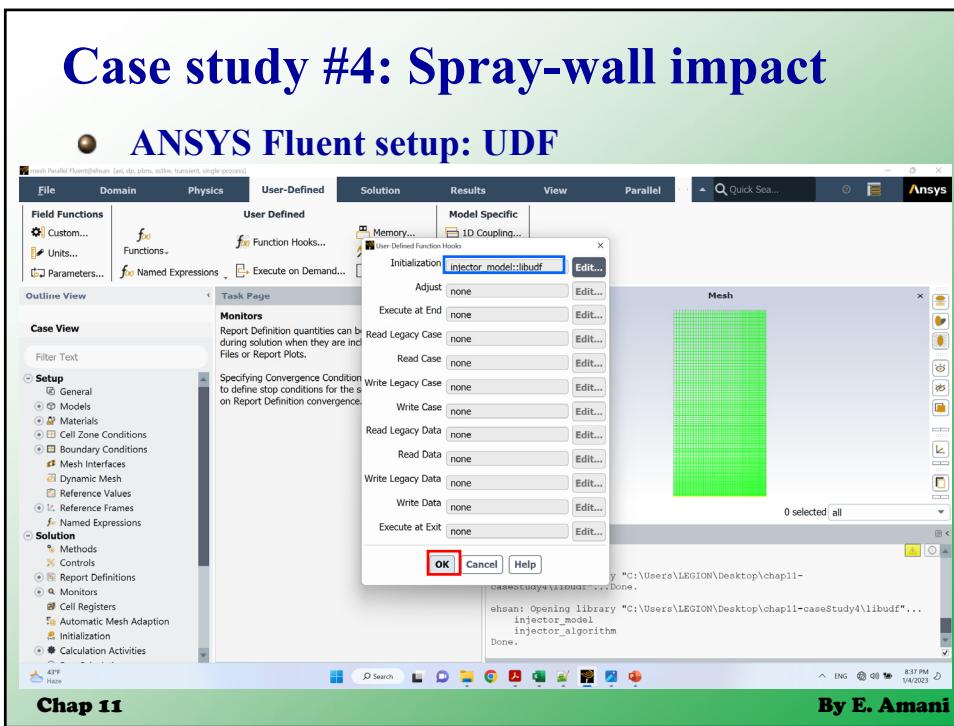
Injection modeling hierarchy

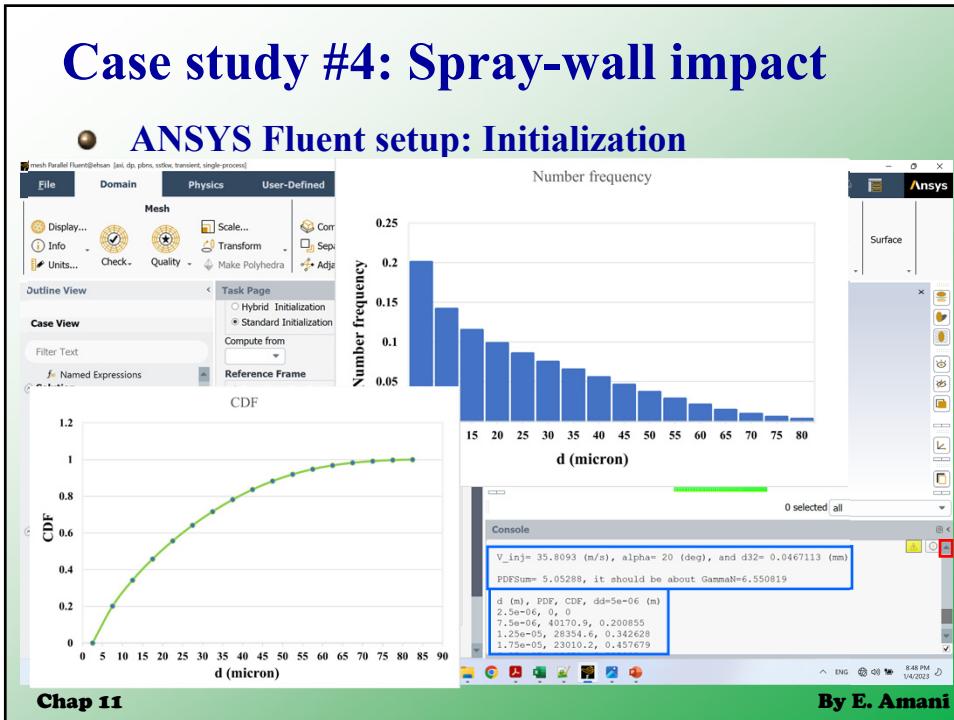
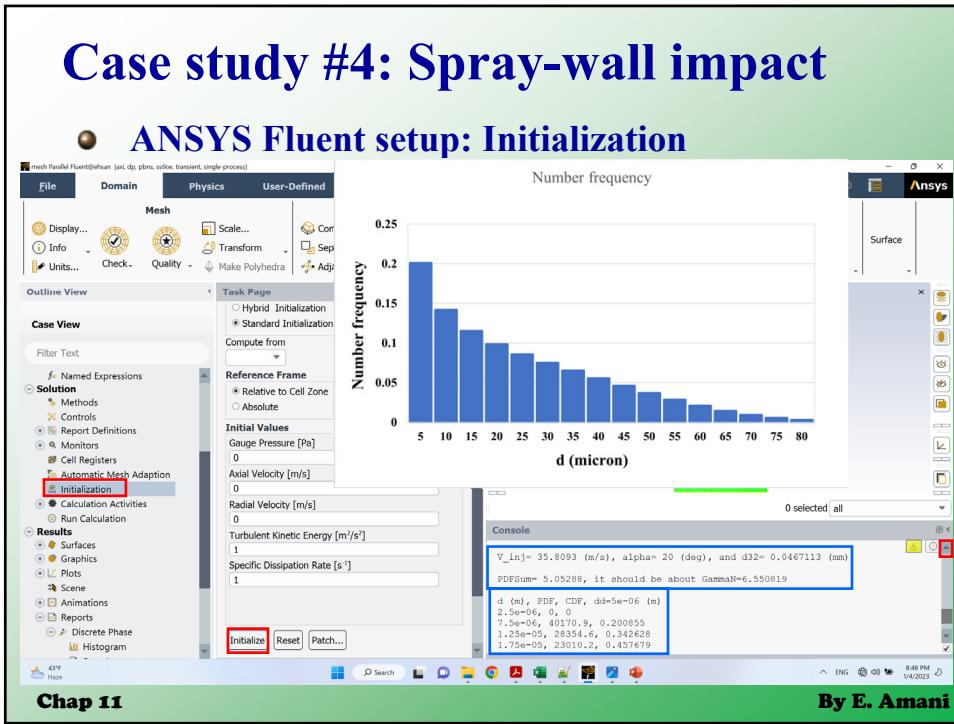


Case study #4: Spray-wall impact

ANSYS Fluent setup: UDF







Case study #4: Spray-wall impact

ANSYS Fluent setup: UDF

```
UDF.c
1  /*************************************************************************/
2  /*
3  * NSMF-chap11-VOF
4  * By Dr. Ehsan Amani
5  * First version: January 2023
6  */
7  /*************************************************************************/
8
9 //Headers
10 #include "udf.h"
11 #include "dpm.h"
12 #include "surf.h"
13 #include "random.h"
14
15
16 //INPUTS
17 #define db 1.e6 // Pa
18 #define Cv 0.8
19 #define Cd 0.6
20 #define sIndex 3.5
21 #define rho_l 998.2 // kg/m3
22 #define rho_g 1.225 // kg/m3
23 #define sigma 0.0728 // N/m
24 #define NData 16
25 #define D 500.e-6 // m
26 #define alphad 20 //deg
27 #define L 5.e-3 // m
28 #define dr_inj 0.005 //
29 #define dmin 2.5e-6 //m
30 #define dmax 82.5e-6 //m
31
32
33 void RR_PDFcalc(double sI, double delta, double dmi, double dma, double d[], double PDF[], double* dd);
34 void CDF_calc(double d[], double PDF[], double dd, double CDF[]);
35 double ITM_sample(void);
36
37 double d[Ndata+1]={..}, PDF[Ndata+1]={..}, CDF[Ndata+1]={..};
38 double dd=0., V_inj=0., alpha=0., mdot_inj=0.;
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
```

Chap 11

By E. Amani

Case study #4: Spray-wall impact

ANSYS Fluent setup: UDF

```
UDF.c
40 ////////////////FLUENT UDF Macros
41 DEFINE_INIT(injector_model, domain)
42 {
43     double A, We, d32, delta;
44     int i;
45
46     A=3.0+0.28*L/D;
47     V_inj=Cv*sqrt(2*dP/rho_l);
48     mdot_inj=Cd*(M_PI/4*D*D)*rho_l*V_inj;
49     We=rho_l*pow(V_inj,2.)*D/8./sigma;
50     //alpha=atan(4*M_PI/A*sqrt(3*ro_g/ro_l)/6);
51     alpha=alphad*M_PI/180;
52     d32=135.0*(D/8.)pow(We,-0.74);
53     delta=1.2726*d32;
54
55     Message0("\n V_inj= %g (m/s), mdot_inj= %g (g/s), alpha= %g (deg), and d32= %g (mm) \n",V_inj,mdot_inj,alpha,d32);
56
57     RR_PDFcalc(sIndex, delta, dmin, dmax, d, PDF, &dd);
58     CDF_calc(d, PDF, dd, CDF);
59     Message0("\n d (m), PDF, CDF, dd=%g (m) ", dd);
60     for (i=0; i<(Ndata+1); i++)
61     {
62         Message0("\n %g, %g, %g ",d[i],PDF[i],CDF[i]);
63     }
64 }
```

Chap 11

By E. Amani

Case study #4: Spray-wall impact

- ANSYS Fluent setup: UDF

```

UDF.c
65 DEFINE_DPM_INJECTION_INIT(injector_algorithm,I)
66 {
67     double thetainj, massSum;
68     Particle *p;
69
70     massSum=0;
71     loop(p,I->p_init)
72     {
73         PP_DIAM(p)=ITM_sample();
74
75         thetainj=cheap_uniform_random()*alpha;
76         PP_VEL(p)[0]=V_inj*cos(thetainj);
77         PP_VEL(p)[1]=V_inj*sin(thetainj);
78
79         PP_RHO(p)=rho_1;
80         PP_MASS(p)=M_P1/6*PP_RHO(p)*pow(PP_DIAM(p),3.0);
81         massSum+=PP_MASS(p);
82     }
83
84     loop(p,I->p_init)
85     {
86         //PP_INIT_N(p) = mdot_inj*dt_inj/massSum;
87         //p->number_in_parcel = mdot_inj*dt_inj/massSum;
88         //TP_N(p) = mdot_inj*dt_inj/massSum;
89         PP_FLOW_RATE(p) = mdot_inj*PP_MASS(p)/massSum;
90         //Message0("\n weight= %g , dt_inj= %g \n",mdot_inj*dt_inj/massSum);
91     }
92 }
93
94 ////////////Functions
95 void RR_PDFcalc(double sI, double delta, double dmi, double dma, double d[], dou
96 {
97     int i;
98     double dbin, PDFSum;

```

Chap 11

By E. Amani

Case study #4: Spray-wall impact

- ANSYS Fluent setup: UDF

Modeling Discrete Phase

simulation, you can clear particles that are currently in the domain by clicking the **Clear Particles** button in the **Discrete Phase Model Dialog Box** (p. 4145).

You can choose the **Parcel Release Method** to determine how Ansys Fluent creates parcels. For an overview of the concept of parcels, see **Parcels** (p. 2387). The methods available are:

standard

injects a single parcel per injection stream per time step. The number of particles in the parcel, NP , is determined as follows:

$$NP = \frac{\dot{m}_p \Delta t_{inj}}{\sum_{p=1}^{N_{inj}} m_p} \quad (20.1)$$

where,

- NP is the number of particles in a parcel
- \dot{m}_p is the mass flow rate of the particle stream
- Δt is the time step size
- m_p is the particle mass

$$\dot{m}_s = \frac{\dot{m}_p m_p}{\sum_{p=1}^{N_{inj}} m_p}$$

This is the default method.

```

UDF.c
65 DEFINE_DPM
66 {
67     double thetainj, massSum;
68     Particle *p;
69
70     massSum=0;
71     loop(p,I->p_init)
72     {
73         PE
74         th
75         PE
76         PE
77         PE
78         PE
79         PE
80         ma
81
82     }
83
84     loop(p,I->p_init)
85     {
86         //p->number_in_parcel = mdot_inj*dt_inj/massSum;
87         //TP_N(p) = mdot_inj*dt_inj/massSum;
88         PP_FLOW_RATE(p) = mdot_inj*PP_MASS(p)/massSum;
89         //Message0("\n weight= %g , dt_inj= %g \n",mdot_inj*dt_inj/massSum);
90     }
91
92 }
93
94 ////////////Functions
95 void RR_PDFcalc(double sI, double delta, double dmi, double dma, double d[], dou
96 {
97     int i;
98     double dbin, PDFSum;

```

Chap 11

By E. Amani

Case study #4: Spray-wall impact

- ANSYS Fluent setup: UDF

```

94 ///////////////Functions
95 void RR_PDFcalc(double sI, double delta, double dmi, double dma, double d[], double PDF[], double* dd)
96 {
97     int i;
98     double dbin, PDFSum;
99
100    d[0]=dmi;
101    PDF[0]=1;
102    *dd=(dma-dmi)/Ndata;
103    PDFSum=0;
104    for (i=1; i<(Ndata+1); i++)
105    {
106        d[i]=d[i-1]+(*dd);
107        dbin=d[i-1]+(*dd)/2;
108        PDF[i]=sI/delta*pow(dbin/delta,sI-4)*exp(-pow(dbin/delta,sI)); //1./GammaN=0.15265
109        PDFSum+=PDF[i]*(*dd);
110    }
111    Message0("\n PDFsum= %g, it should be about GammaN=6.550819 \n",PDFSum);
112    for (i=1; i<(Ndata+1); i++) //normalization
113    {
114        PDF[i]/=PDFSum;
115    }
116}

```

$f(d) = \frac{1}{\Gamma \delta} \left(\frac{d}{\delta}\right)^{s-4} \exp\left[-\left(\frac{d}{\delta}\right)^s\right]$

Chap 11

By E. Amani

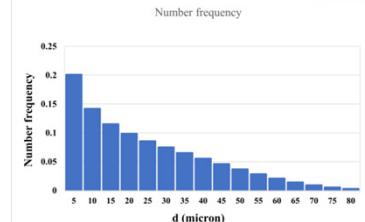
Case study #4: Spray-wall impact

- ANSYS Fluent setup: UDF

```

117 //Inverse transform method
118 void CDF_calc(double d[], double PDF[], double dd, double CDF[])
119 {
120     int i;
121
122     CDF[0]=0;
123     for (i=1; i<(Ndata+1); i++)
124     {
125         CDF[i]=CDF[i-1]+dd*PDF[i]; ——————  $F(d) = \int_0^d f(d') dd'$ 
126     }
127
128 }
129
130 double ITM_sample(void)
131 {
132     double y, w;
133     int i;
134
135     y=cheap_uniform_random();
136     for (i=1; i<(Ndata+1); i++)
137     {
138         if (y<=CDF[i]) break;
139     }
140     w=(CDF[i]-y)/(CDF[i]-CDF[i-1]);
141
142     return w*d[i-1]+(1.-w)*d[i];
143 }

```



Chap 11

By E. Amani

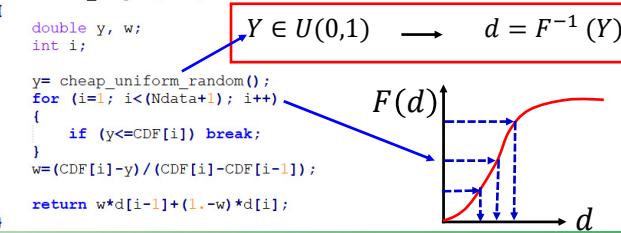
Case study #4: Spray-wall impact

ANSYS Fluent setup: UDF

```

117 //Inverse transform method
118 void CDF_calc(double d[], double PDF[], double dd, double CDF[])
119 {
120     int i;
121
122     CDF[0]=0;
123     for (i=1; i<(Ndata+1); i++)
124     {
125         CDF[i]=CDF[i-1]+dd*PDF[i];
126     }
127
128 }
129
130 double ITM_sample(void)
131 {
132     double y, w;
133     int i;
134
135     y= cheap_uniform_random();
136     for (i=1; i<(Ndata+1); i++)
137     {
138         if (y<=CDF[i]) break;
139     }
140     w=(CDF[i]-y)/(CDF[i]-CDF[i-1]);
141
142     return w*d[i-1]+(1.-w)*d[i];
143 }

```

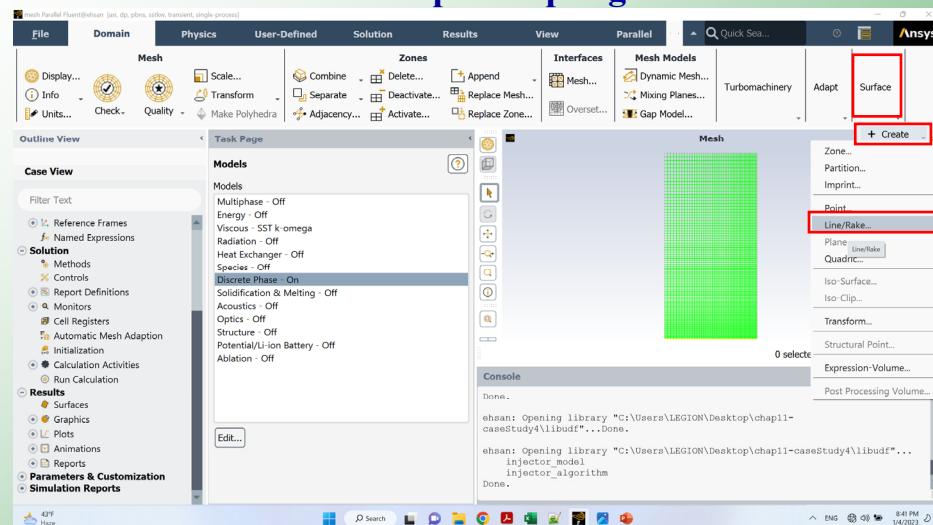


Chap 11

By E. Amani

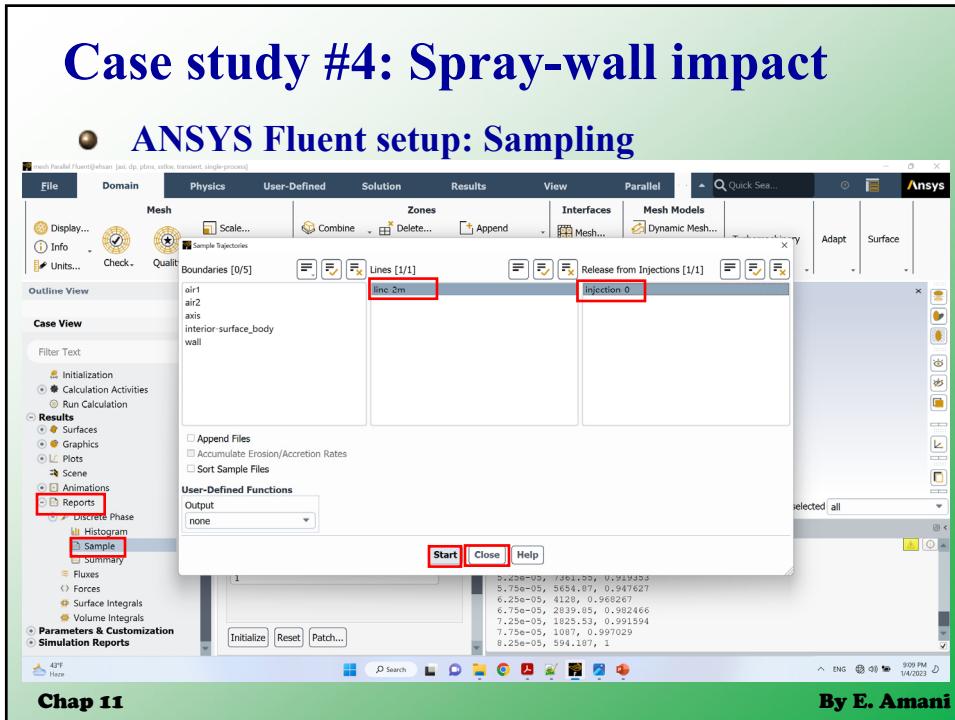
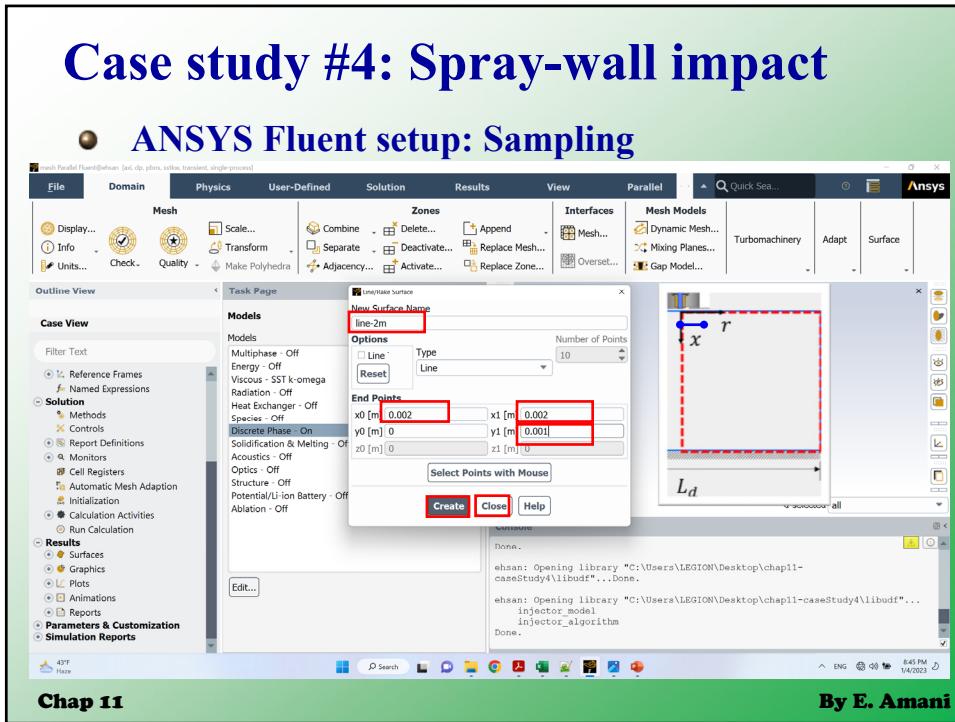
Case study #4: Spray-wall impact

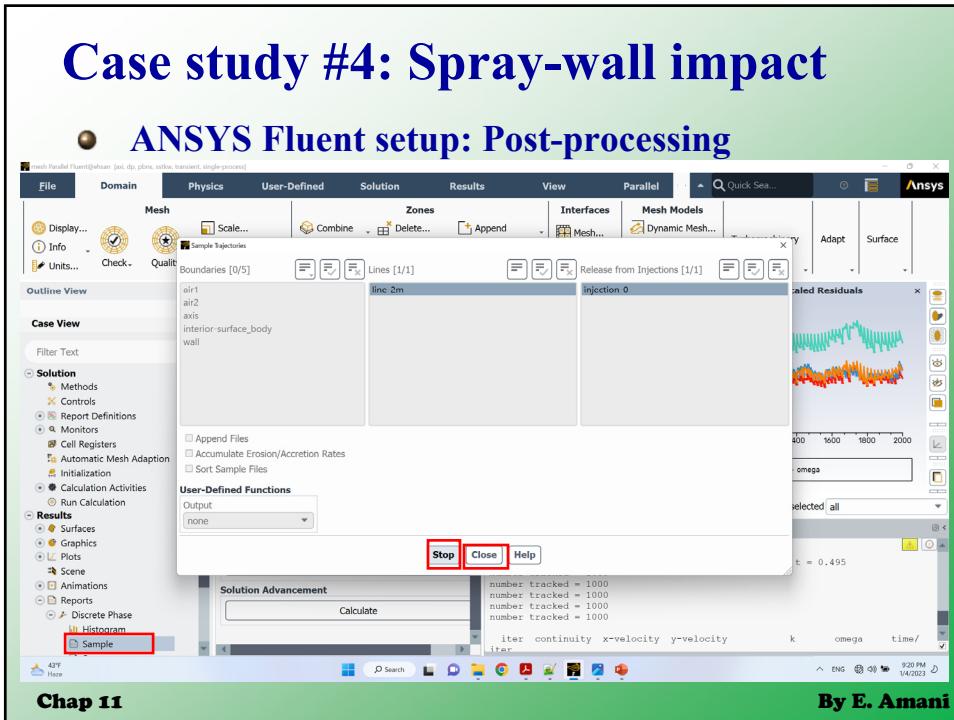
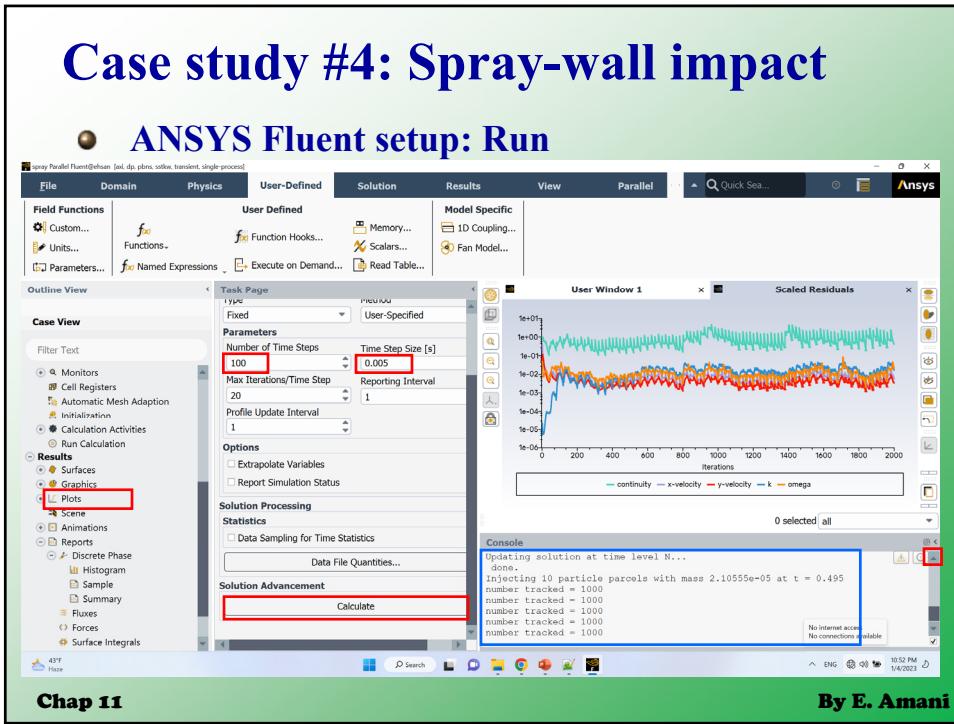
ANSYS Fluent setup: Sampling



Chap 11

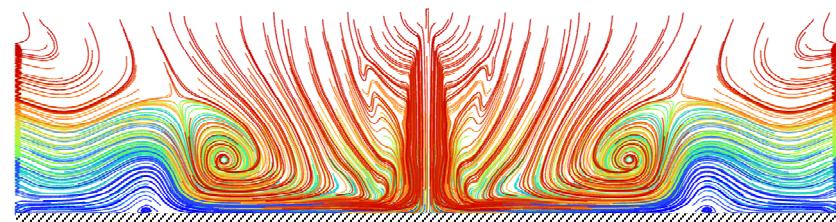
By E. Amani





Case study #4: Spray-wall impact

- ANSYS Fluent setup: Post-processing

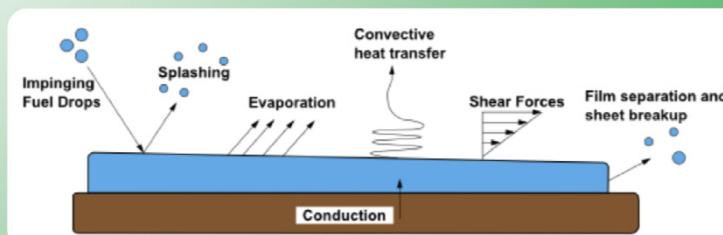


Chap 11

By E. Amani

Boundary conditions

- Droplet-wall impact models
 - Stick
 - Reflect
 - Wall-jet (hot walls with no film)
 - Wall-film, e.g., you can see Ref. [2]
 - ...

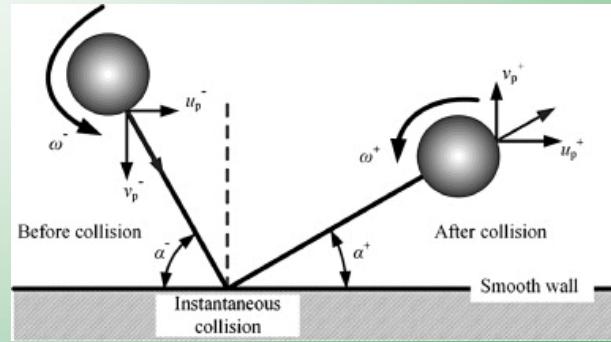


Chap 11

By E. Amani

Boundary conditions

- Solid-particle-wall impact models
 - Regular bouncing



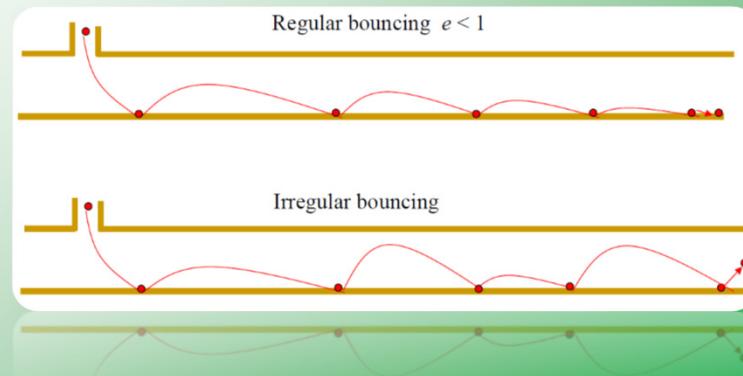
$$m_2 \rightarrow \infty, r_2 \rightarrow \infty$$

Chap 11

By E. Amani

Boundary conditions

- Solid-particle-wall impact models
 - Regular bouncing
 - Irregular bouncing



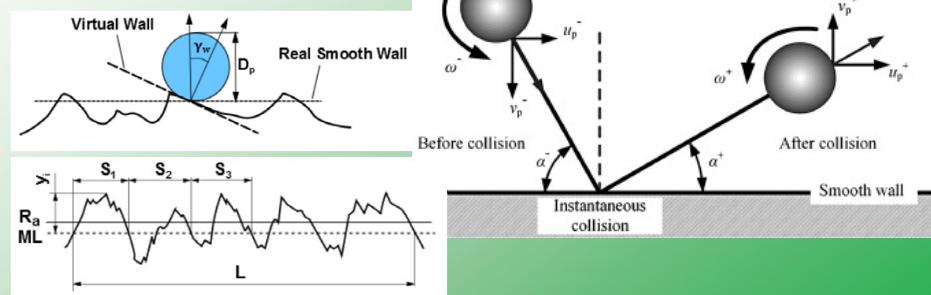
Chap 11

By E. Amani

Boundary conditions

- Solid-particle-wall impact models
 - Regular bouncing
 - Irregular bouncing

$$\alpha_{\text{impact}} = \alpha^- + \alpha_W$$



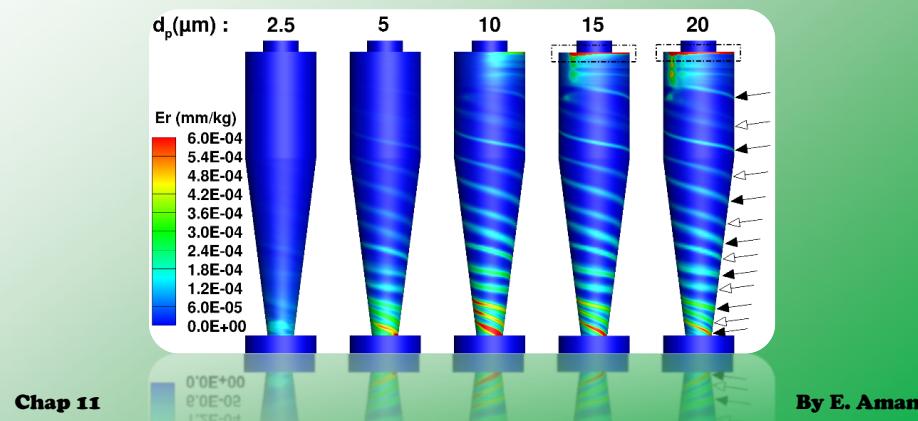
(See HW#11)

Chap 11

By E. Amani

Post-processing

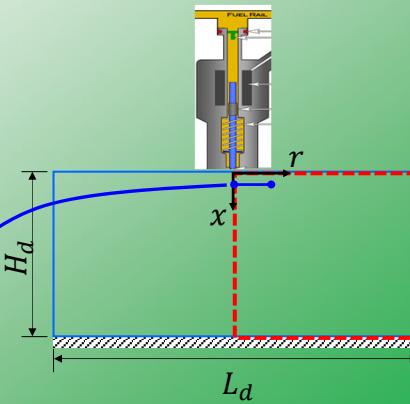
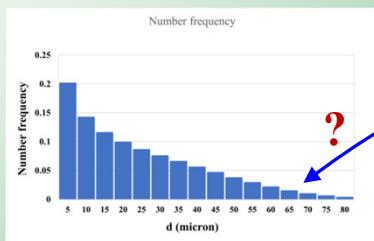
- Erosion
 - Solid particles
 - Irregular bouncing, e.g., you can see Ref. [3]



Hands-on practice

HW#11

- ✓ Processing the samples recorded near the nozzle and check the PDF of injected particles
- ✓ ...



Chap 11

By E. Amani

References

1. Ashgriz, Nasser, ed. *Handbook of atomization and sprays: theory and applications*. Springer Science & Business Media, 2011.
2. Asgari, Behrad, and Ehsan Amani. An improved spray-wall interaction model for Eulerian-Lagrangian simulation of liquid sprays. *International Journal of Multiphase Flow* 134 (2021): 103487.
3. Tofighian, H., E. Amani, and M. Saffar-Aval. "A large eddy simulation study of cyclones: The effect of sub-models on efficiency and erosion prediction." *Powder Technology* 360 (2020): 1237-1252.

Chap 11

By E. Amani

