Mathematical modeling of a spray-dryer

in Matlab and OpenFoam.

Somma Andrea

Chemical plants and process operations management

Politecnico di Milano - 2022

Mathematical Formulation: Balances on an evaporating particle and gas phase

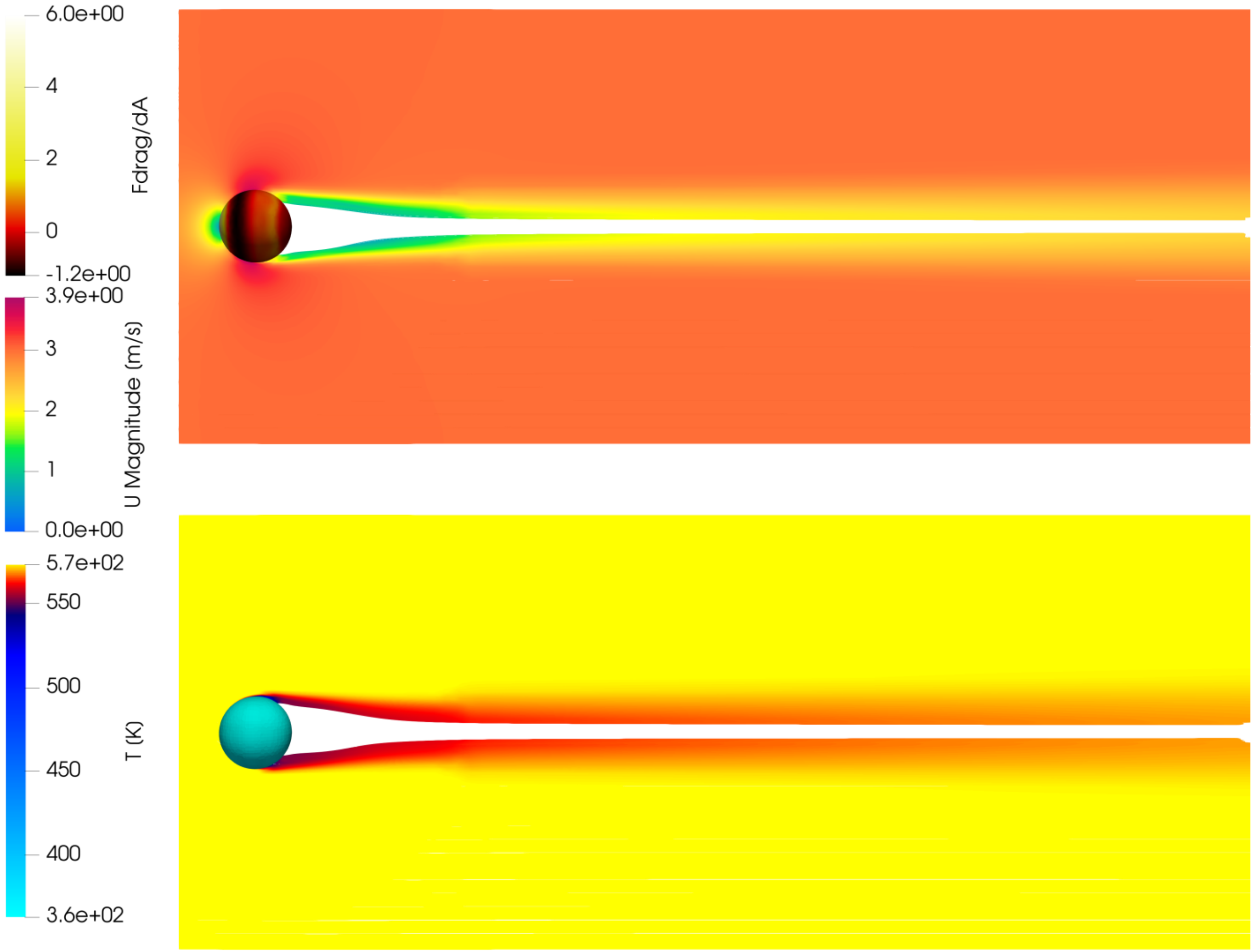
- ***MomB*** *on the particle;* › $m_p \frac{dv_s}{dt} = (\rho_p - \rho_g)V_p g - 3f_D \mu \pi D_p v_s - v_s K_p S_p \left(P_w - P^0(T_g)\right)$
- ***MassB*** *on the particle* › $\frac{dm_p}{dt} = K_p S_p \left(P_w - P^0(T_g)\right)$
- ***MassB*** *on the gas phase;* › $\frac{dG_i}{dt} = K_p S_p \left(P_w - P^0(T_g)\right) \eta A v_p$
- ***EnergyB*** *on the particle;* › $m_p c_{pL} \frac{dT_p}{dt} = hS(T_g - T_p) + K_p S_p \left(P_w - P^0(T_g)\right) \Delta H_{ev}$
- ***EnergyB*** *on the gas phase;* › $G c_p \frac{dT_g}{dt} = h S_p (T_p - T_g) \eta A v_p$
- ***Velocity to axial coordinate*** conversion › $\frac{dz}{dt} = v_p$

Closure Terms:
Evaluation of Nu,Sh,Cd

Chosen relations:

- $Nu = 2+0.4Re^{0.5}Pr^{0.33}$
- $Sh = 2+0.4Re^{0.5}Sc^{0.33}$
- $Cd = 24/Re(1+0.14Re^{0.7})$

*Simulation of a still, **cold** particle in a **hot** stream; drag coefficient, velocity field and temperature field are presented.*



Code Overview:
Everything is evaluated inside the integration loops

```
function spraydryer = solver(t,x)
    ...
    x_H2O = (mp - mp0*fat) / mp;
    rho_avg = x_H2O * rhoL + (1 - x_H2O) * rho_milk;
    V = mp / rho_avg;
    dp = (6 * V / pi)^(1/3);
    Dmin = 83.87e-6;
    dp = max(Dmin, dp);
    ...
 end
```

```
function spraydryer = solver(t,x)
    ...
    Re = abs(rhoG * vs * dp / muG);
    Pr = muG * CpG / kG;
    Sc = muG / rhoG / Diff;
    Nu = 2 + 0.4 * Re^0.5 * Pr^(1/3);
    Sh = 2 + 0.4 * Re^0.5 * Sc^(1/3);
    f = (1 + 0.14*Re^0.7);
    h = Nu * kG / dp;
    Kc = Sh * Diff/ dp;
    Kpw = Kc / R / Tg * MWwater;
    ...
 end
```

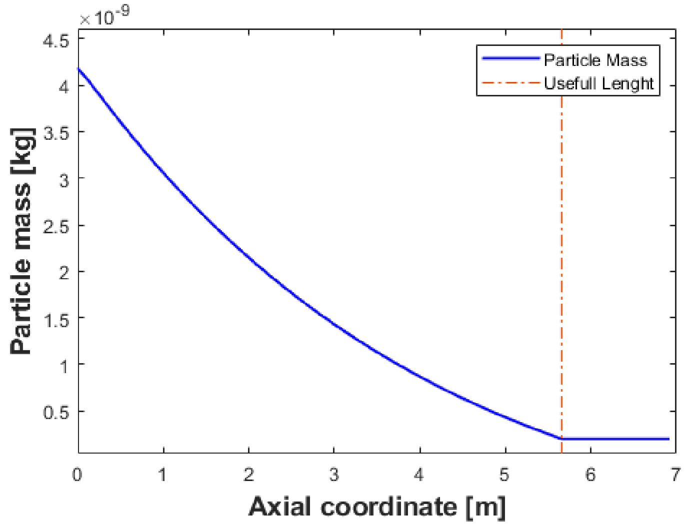
```
function spraydryer = solver(t,x)
    ...
    mmin = mp0 * fat;
    mp = max(mmin, mp);
    ...
    if mp > mmin + mmin/10000
        spraydryer(1) = Kpw * Sp * (Pw - P0);
    else
        spraydryer(1) = 0;
    end
    ...
 end
```

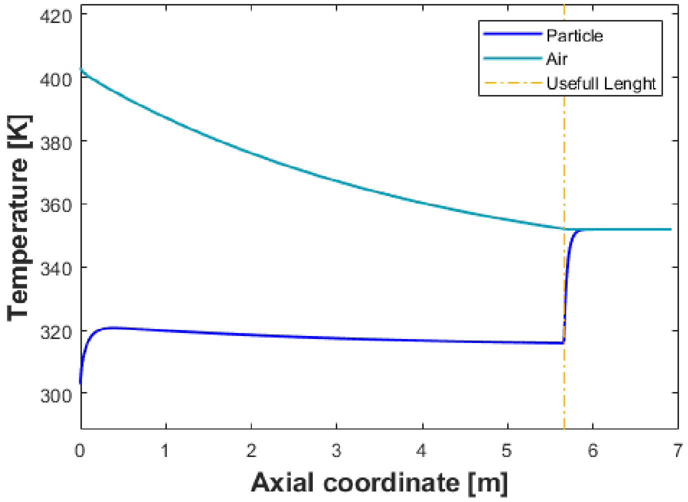
***Adimensional numbers** and **coefficients** are calculated inside the integration function, furthermore to ensure **physical results** some considerations have been done.*

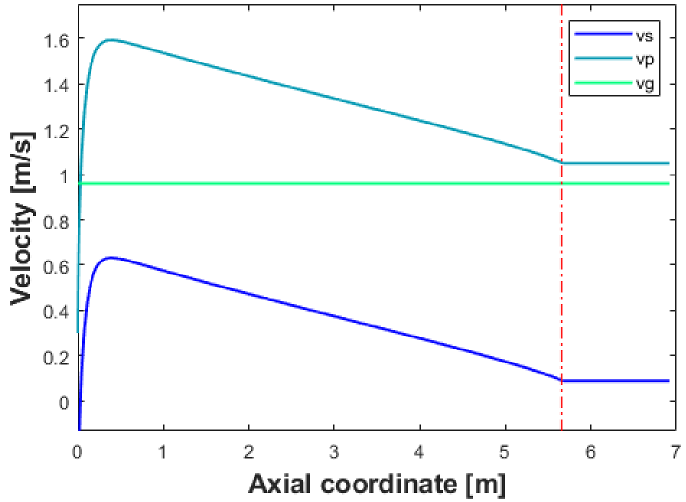
Results: Realistic trends obtained

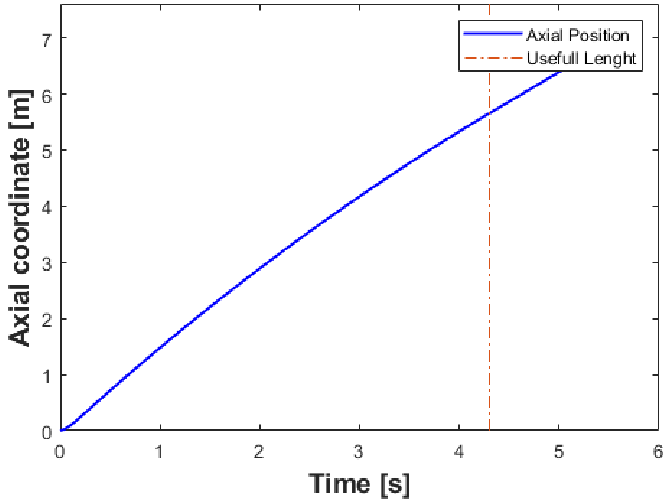
*Proposed results present **realistic trends** for the calculated fields:*

- *Particle mass reaching a **minimum**;*
- *p-velocity almost reaching **G-velocity**;*
- ***Temperature** sharply increasing after the consuption of water.*

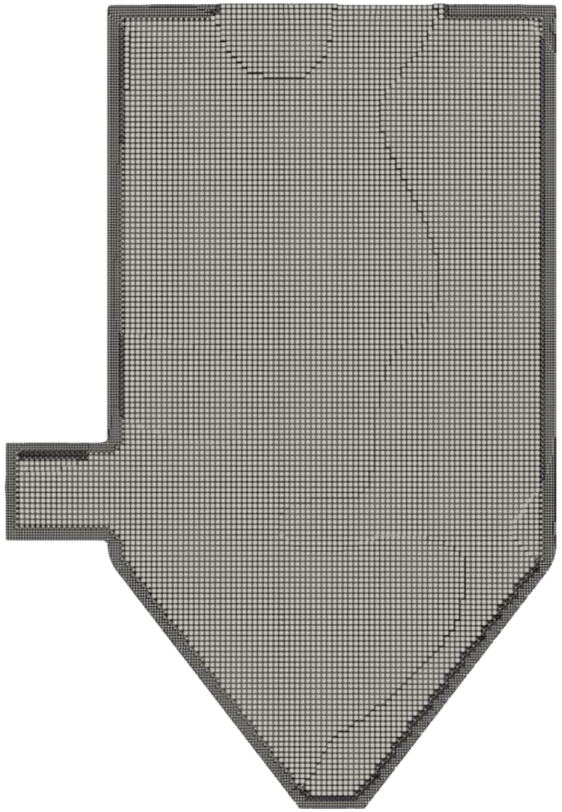




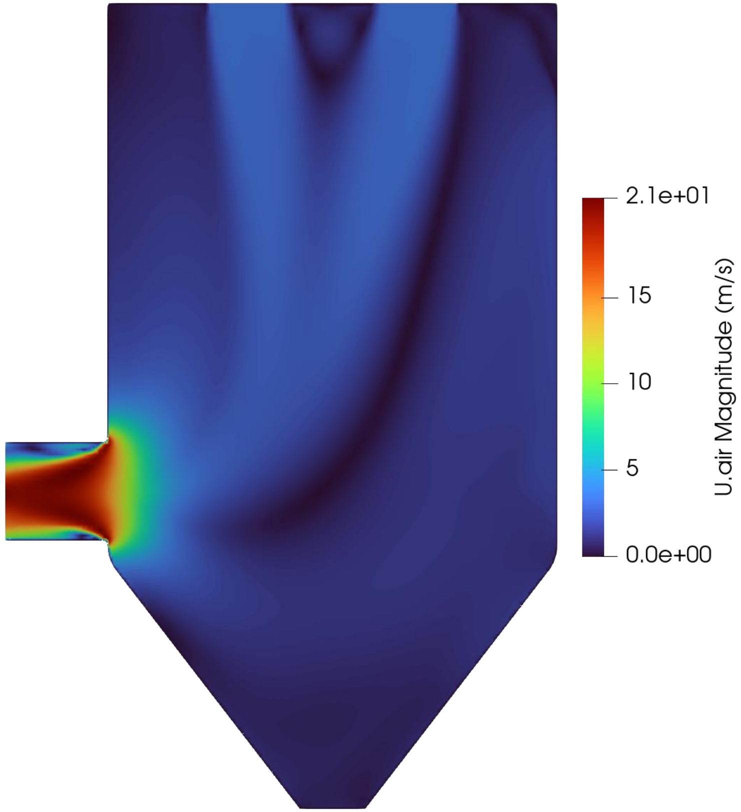


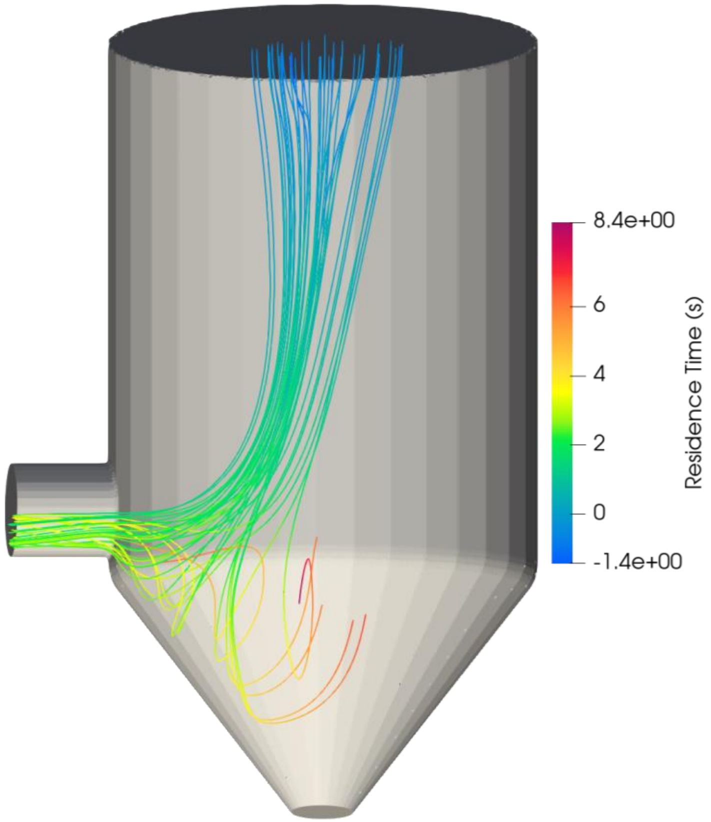


OpenFoam setup:
Ras-model,
K-omega-SST



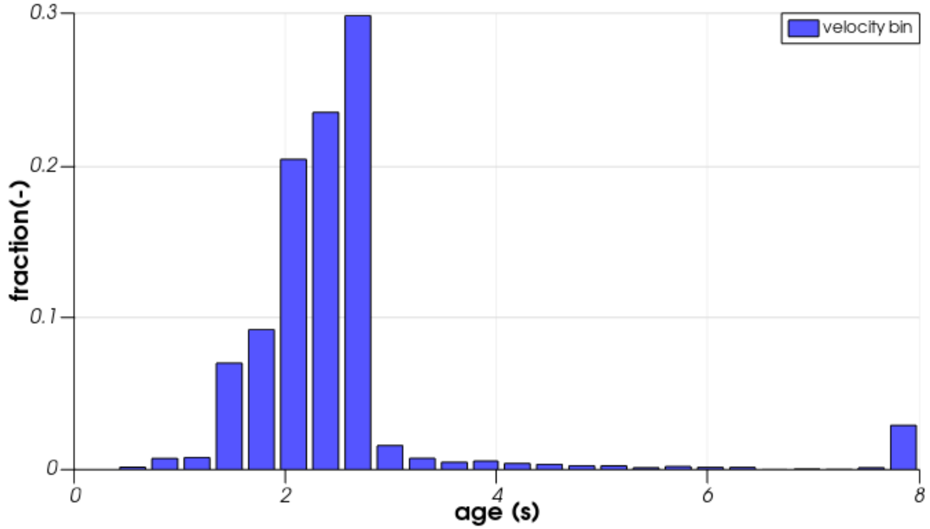
3D-case hex-dominant mesh 2Mln cells



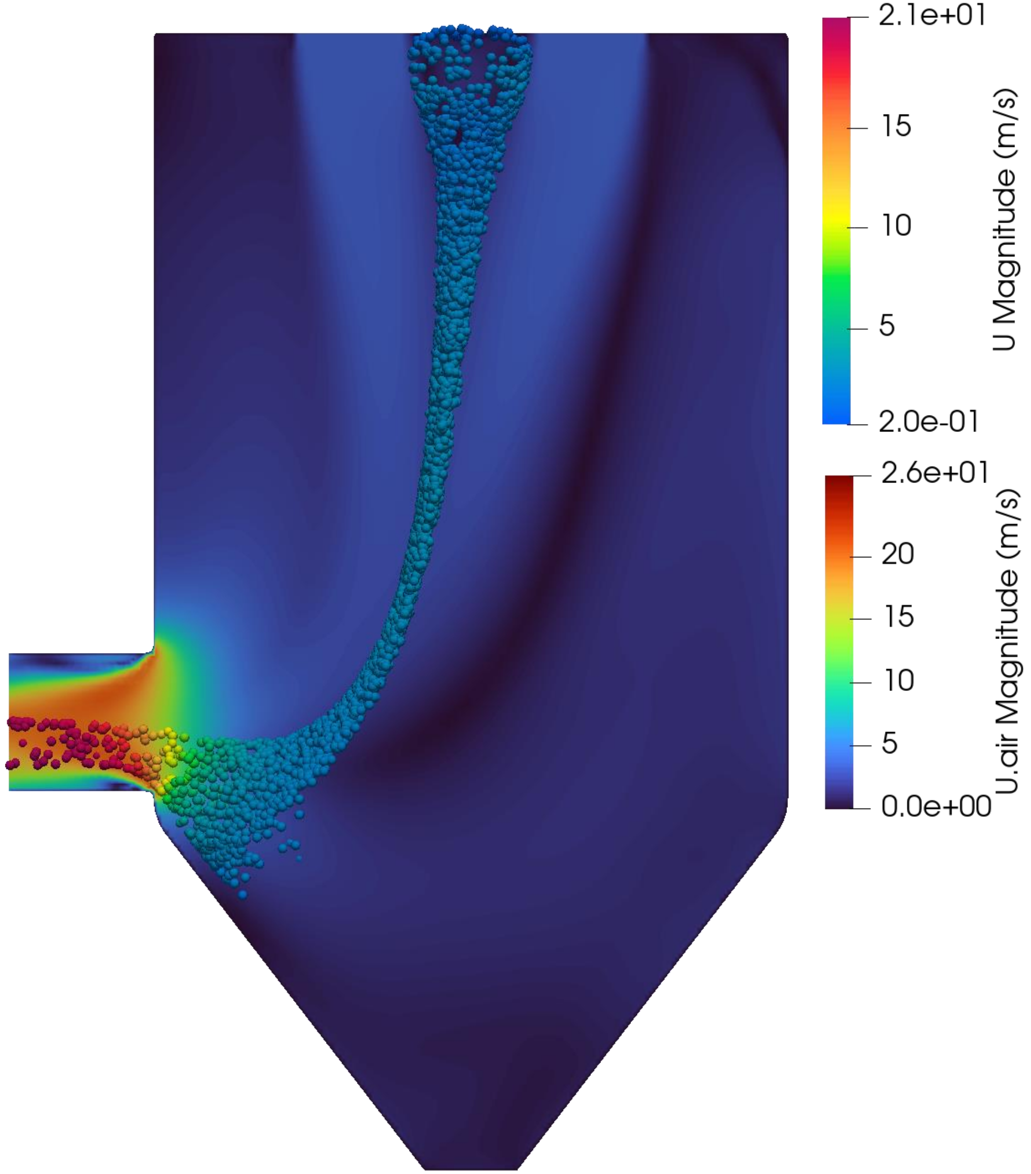


Velocity field *and streamline* ***residence time****, in SI units. Recirculations and velocity gradients are* ***not*** *negligible.*

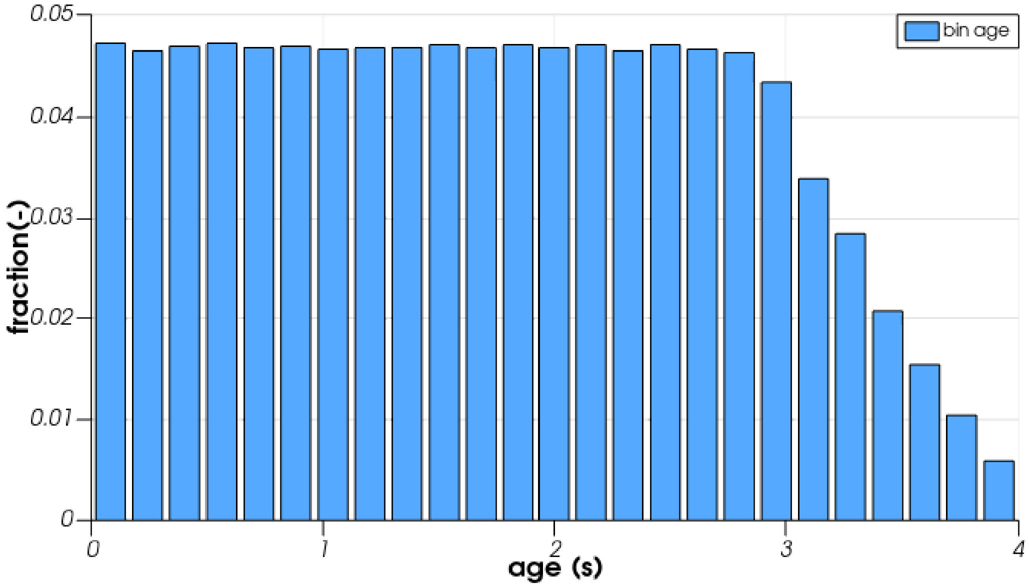
OpenFoam setup:
Ras-model,
denseParticleFoam solver



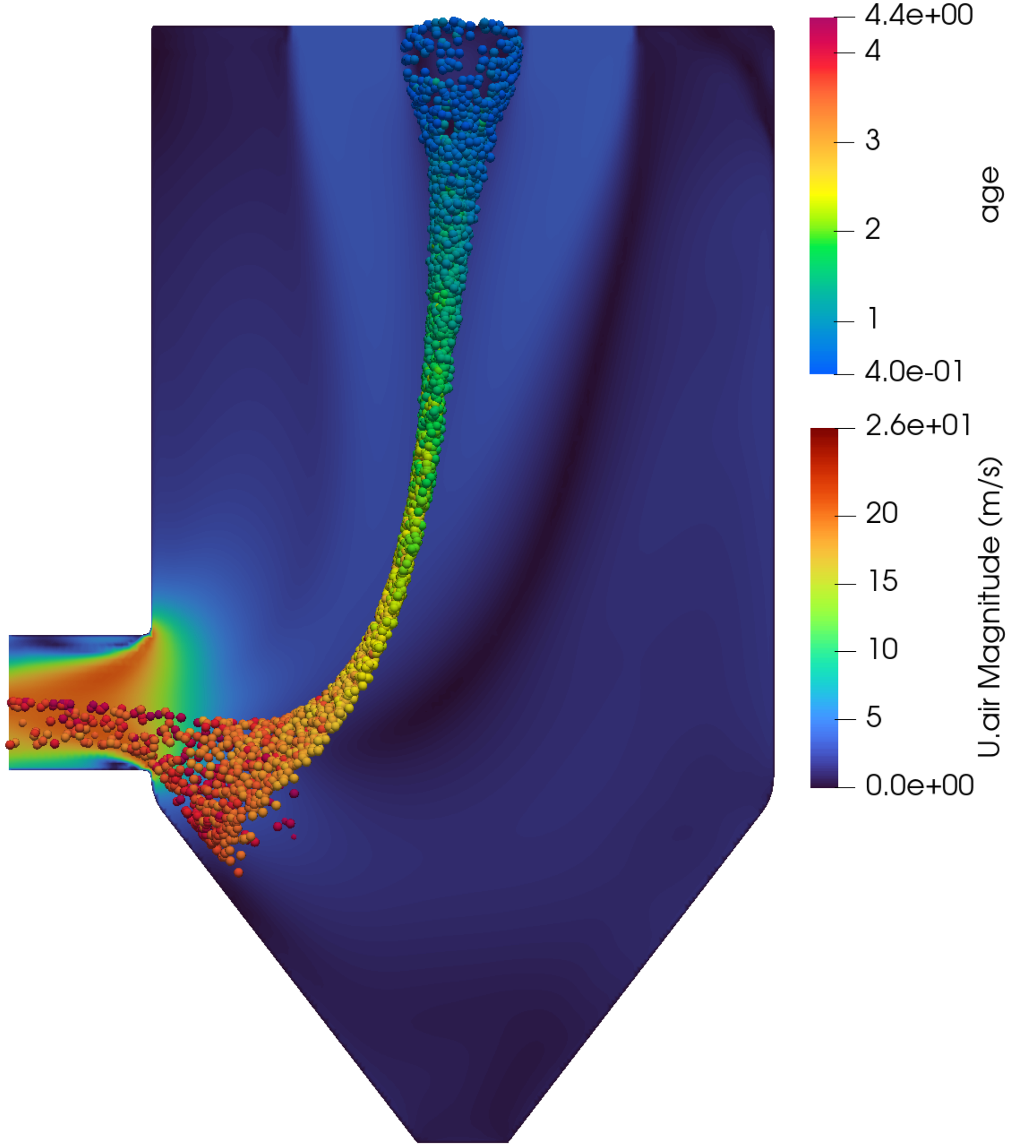
Particles tend to ***strictly follow*** *the air stream, given the almost negligible mass (very mass-diluted sistem). The displayed graph is at 4s after the first injection.*



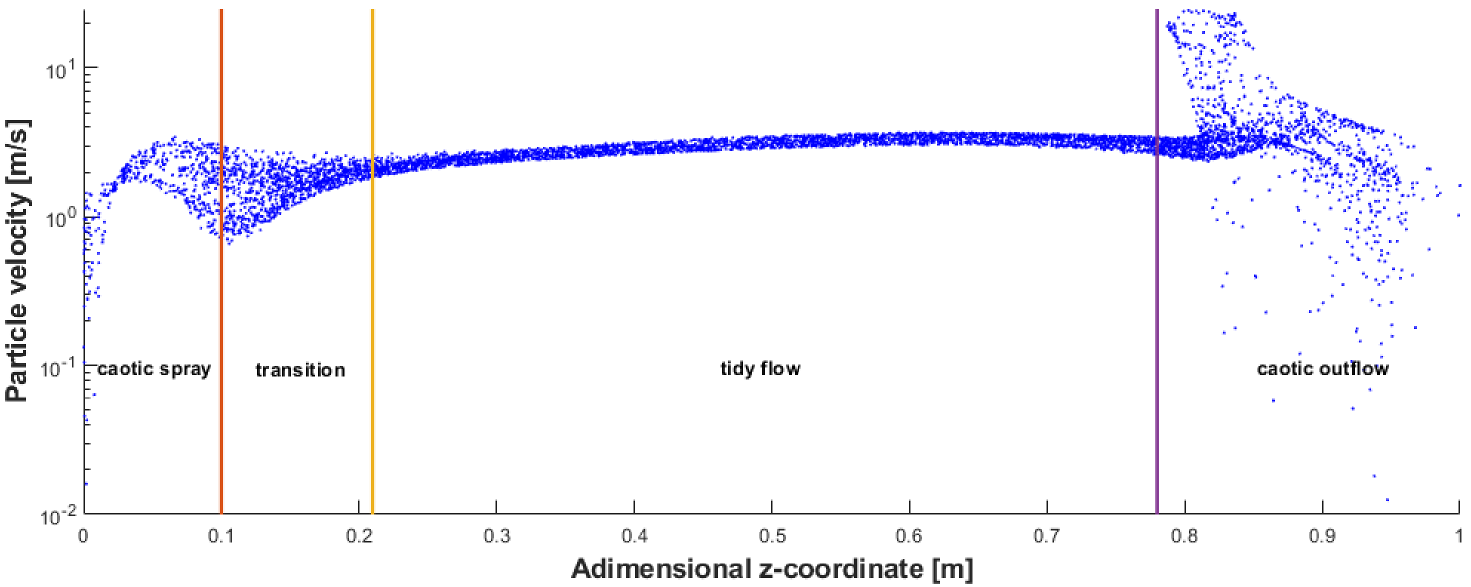
OpenFoam setup:
Ras-model,
denseParticleFoam solver



*Particles **have not** the required age, due to the non uniform velocity field. The displayed graph is at 4s after the fisrt particle injection.*



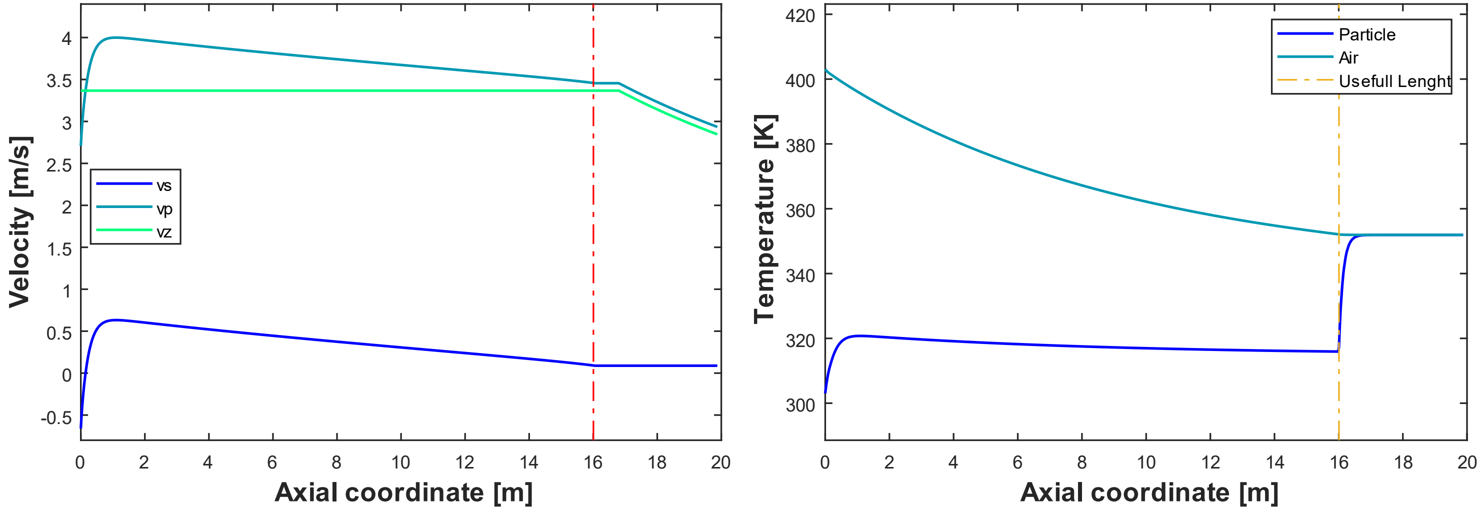
Jet correlation: Riyiaz and Ahmad 2014



$$v_z = \begin{cases} U_{exit}, & \boldsymbol{if}\ 0 \leq z \leq x \\ 6.11 D_{outlet} U_{exit} \frac{1}{z}, & \boldsymbol{if}(\ z > x\ \&\ v_z > v_{dev}) \\ v_{dev}, & \boldsymbol{otherwise} \end{cases}$$

A new correlation is nedeed to ensure that the particle is affected by ***a realistic velocity***, with this a new corrected 1D-model can be developed.

D ratio = 2, lenght = 16.0072m



Workflow Overview:
Mesh, Boundary conditions, Solve!

```
.
└── mesh/
    ├── constant/
    │   ├── extendedFeatureEdgeMesh
    │   ├── polyMesh
    │   └── triSurface
    ├── processor*/
    │   ├── 0
    │   └── constant
    └── system/
        ├── blockMeshDict
        ├── controlDict
        ├── decomposeParDict
        ├── snappyHexMeshDict
        └── surfaceFeaturesDict
```

*Tipical folder tree of a **mesh** in OpenFoam, with SnappyHexMesh.*

```
.
└── Spray-dryer/
    ├── 0/
    │   ├── k.air
    │   ├── nut.air
    │   ├── omega.air
    │   ├── p
    │   └── U.air
    ├── constant/
    │   ├── polyMesh
    │   ├── cloudProperties
    │   ├── g
    │   ├── momentumTransport.air
    │   └── physicalProperties
    ├── processor*/
    │   ├── 0
    │   ├── constant
    │   └── times/
    │       ├── 1
    │       ├── 2
    │       └── ...
    └── system/
        ├── controlDict
        ├── decomposeParDict
        ├── fvSchemes
        └── fvSoltion
```

*Tipical folder tree of a **densePartilceFoam** case in OpenFoam, cloudProperties is the file where all paticle properties/models are defined.*

Thank you for the attention.

github.com/sommaa