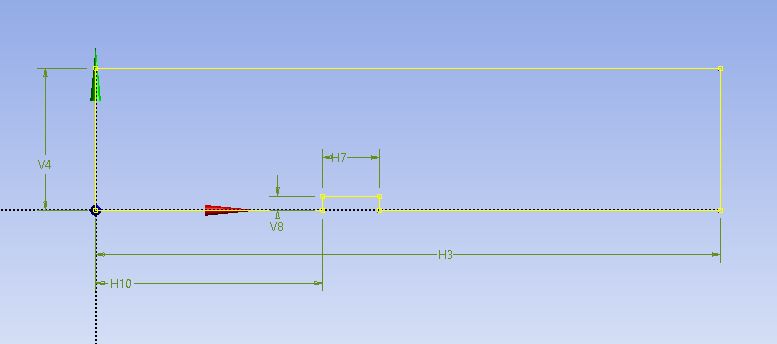
**Modelling and simulation of Combustion of Agricultural Residues Pellet:**

**Objective of simulation: -**

Firstly, we want to simulate drying of the agricultural residue pellet with moisture **(M=0.25% wt.),** having diameter and length, 10 mm, and 20 mm respectively. For this, we have drawn a 2-D **“axisymmetric computational domain”,** which contains a rectangular shape having length 220 mm and 50 mm width shown in fig. 1 (Fluid domain). We have drawn another sketch for pellet which is a rectangular shape with width of 5 mm and length of 20mm (making it also fluid as we consider it porous medium) which is 80 mm from the origin. Further, we supply heating air (500 K) at atmospheric pressure with air velocity (0.05 m/s) in such a way that laminar flow is maintained near the pellet. We would like to simulate the drying of the pellet under the influence of the hot air stream.

Modelling geometry**: -**

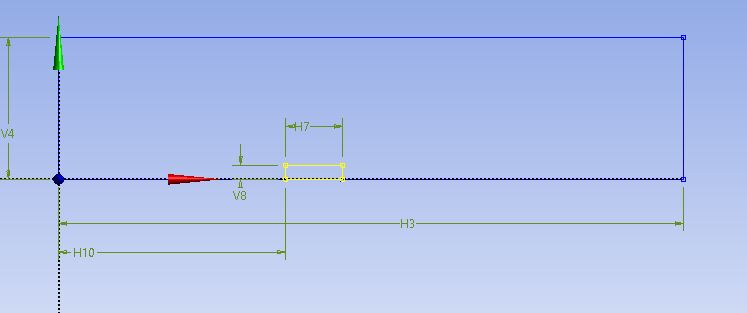


Axis

inlet

Outlet

**Fig.1 Fluid (Gas) domain**



Fluid domain

Pellet with half geometry

**Fig.2 Fluid domain for pellet (porous)**

Problems facing:

1. In order to allow the air stream to dry the pellet, we made 3 species in the mixture (%wt.) i.e water vapour, air and solid water (L) to apply both for fluid domain as well as for porous domain. But we have taken solid water (L) in wt. (%) is zero throughout the gas domain( Fluid). For this we have written user define function (UDF) for initial humidity of air, moisture update, moisture source term, pellet source term and heat source term to simulate our problem (axis-symmety). After defining the number of memory taken in UDF we went for compile the udf (.c file). When we went for initialisation the below mentioned error came and abnormal exit happen (whole fluent window shut-off automatically). Kindly advise on what we are doing wrong.

UDF:

#include "udf.h"

#include "unsteady.h"

#define Mb 0.25 /\* MASS FRACTION OF BOUND WATER IN SOLID\*/

#define M 0.01 /\*Mass fraction of water vapour @500K\*/

DEFINE\_INIT(initial\_humidity,domain)

{

real Ph2o,Psat,RH,A,B,C,Pnew,Xw,Xa,Tnew; /\* My= mole fraction of bound water in wood\*/

Thread \*t;

cell\_t c;

thread\_loop\_c(t,domain)

{

begin\_c\_loop(c,t)

{

Tnew=C\_T(c,t);

Pnew=C\_P(c,t);

C\_UDMI(c, t, 0) = M;

if(Tnew>=300.0)

{

A=3.55; /\* Temperature range 379-573K Liu and Lindsay, 1970\*/

B=643.748;

C=-198.043;

Psat = pow(10, (A - B / (Tnew + C)));

}

else if(Tnew<300.0)

{

A=5.40; /\* Temperature range 273-303K Bridgeman and Aldrich, 1964\*/

B=1838.675;

C=-31.737;

Psat = pow(10, (A - B / (Tnew + C)));

}

Xw= 2\*3.14\*M\* (29.0 / 18.0); /\* Mole fraction of water vapour\*/

Xa = 1 - Xw; /\* Mole fraction of air\*/

Ph2o=2\*3.14\*(Xa\*Pnew/100000.0);

RH=2\*3.14\*(Ph2o/Psat);

C\_UDMI(c,t,1)=RH;

}

end\_c\_loop(c,t)

}

}

DEFINE\_ADJUST(Moisture\_Update, domain)

{

real n\_ts, time\_step, time, t\_hour, last\_timestep, Tabs, RH, A, K, K1, K2;

real Me, k, n, dMdt, Mnew, My;

Thread \*t;

cell\_t c;

n\_ts = RP\_Get\_Integer("time\_step");

time\_step = RP\_Get\_Real("physical\_time\_step");

time = RP\_Get\_Real("flow-time");

t\_hour = time / 3600;

if (last\_timestep != n\_ts)

{

last\_timestep = n\_ts;

thread\_loop\_c(t, domain)

{

begin\_c\_loop(c, t)

{

Tabs = C\_T(c, t);

My= C\_UDMI(c, t, 0);

RH = C\_UDMI(c, t, 1);

A = 349.0 + 1.29 \* (Tabs - 273.15) + 0.0135 \* pow((Tabs - 273.15), 2);

K = 0.805 + 0.000736 \* (Tabs - 273.15) - 0.00000273 \* pow((Tabs - 273.15), 2);

K1 = 6.27 - 0.00938 \* (Tabs - 273.15) - 0.000303 \* pow((Tabs - 273.15), 2);

K2 = 1.91 + 0.0407 \* (Tabs - 273.15) - 0.000293 \* pow((Tabs - 273.15), 2);

Me = 2\*3.14\*((1800.0 / A) \* ((K \* RH) / (1 - K \* RH) + (K \* K1 \* RH + 2.0 \* K1 \* K2 \* pow(K \* RH, 2)) / (1 + K \* K1 \* RH + K1 \* K2 \* pow(K \* RH, 2))));

k = 0.02958 - (0.4456 \* RH) + (0.01215 \* (Tabs - 273.15));

n = 0.13365 + (1.93653 \* RH) - (1.77431 \* pow(RH, 2)) + (0.009468 \* (Tabs - 273.15));

if (Tabs <= 473.15)

{

dMdt = 2\*3.14\*((Mb - Me) \* (-k \* n \* pow(t\_hour, (n - 1))) \* pow(3600, -1));

}

else if (Tabs > 473.15)

{

dMdt = 2\*3.14\*(Mb \* (-k \* n \* pow(t\_hour, (n - 1))) \* pow(3600, -1));

}

Mnew = (time\_step \* dMdt) + My; /\*updated water vapour in gas domain\*/

C\_UDMI(c, t, 2) = Mnew;

C\_UDMI(c, t, 3) = dMdt;

}

end\_c\_loop(c, t)

}

}

}

DEFINE\_SOURCE(Pellet\_source, c, t, dS, eqn)

{

real Sw = 0.0;

real dMdt, Rho;

real e = 0.4;

dMdt = C\_UDMI(c, t, 3);

Rho = C\_R(c, t);

Sw = -2\*3.14\*dMdt \* (1 - e) \* Rho;

dS[eqn] = 0.0;

C\_UDMI(c, t, 4) = Sw;

return Sw;

}

DEFINE\_SOURCE(Moisture\_source, c, t, dS, eqn)

{

real Sm = 0.0;

real e = 0.4;

real dMdt, Rho;

dMdt = C\_UDMI(c, t, 3);

Rho = C\_R(c, t);

Sm = 2\*3.14\*dMdt \* (1 - e) \* Rho;

dS[eqn] = 0.0;

C\_UDMI(c, t, 5) = Sm;

return Sm;

}

DEFINE\_SOURCE(Heat\_source, c, t, dS, eqn)

{

Domain\* d = Get\_Domain(1);

int zone\_id = 5;

real Sh = 0.0;

real dMdt, RHO, Rho, Hfg, Tabs, e, Mnew;

Mnew = C\_UDMI(c, t, 2);

e = (0.623 - 0.25 \* Mnew);

Rho = C\_R(c, t);

Tabs = C\_T(c, t);

RHO = Rho \* (1 - (0.623 - 0.25 \* Mnew));

dMdt = C\_UDMI(c, t, 3);

Hfg = -2386.0 \* (Tabs - 273.15) + 2503000.0;

Sh = 2\*3.14\*dMdt \* (1 - e) \* RHO \* Hfg;

dS[eqn] = 0.0;

C\_UDMI(c, t, 6) = Sh;

return Sh;

}

Error:

