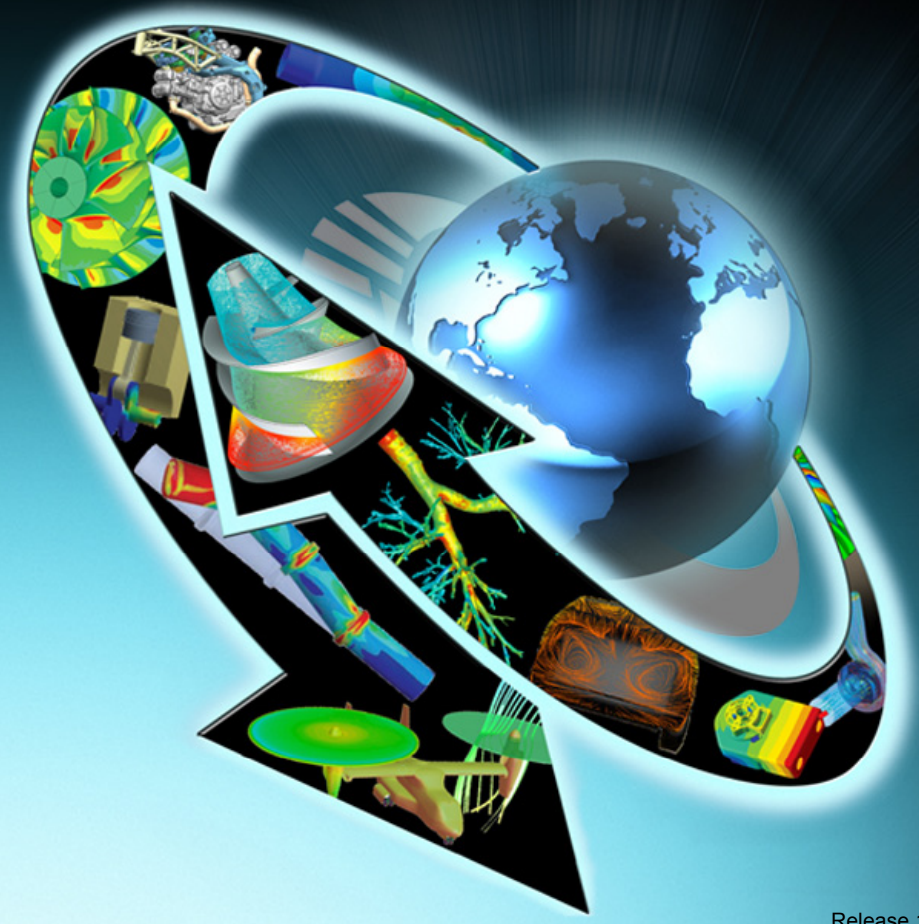
ANSYS®

Chapter 5

UDFs for the Discrete Phase Model



Advanced FLUENT User Defined Functions

Release 13.0
December 2010

DPM Macros (1)

- The most important datatype for DPM model is Tracked_Particle

  Tracked_Particle *p;
- DPM tracks particles by performaing “Lagrangian” tracking
- Particle data at current position
  - P_DIAM(p) Particle diameter
  - P_VEL(p)[I] Particle Velocity
  - P_T(p) Particle Temperature
  - P_RHO(p) Particle density
  - P_MASS(p) Particle mass
  - P_TIME(p) Current time for particle
  - P_DT(p) Particle time step
  - P_LF(p) Particle liquid fraction
  - P_VFF(p) Particle volatile fraction
- Many more particle data macros are available

DPM Macros (2)

- **Values of particle properties at entry to current cell**
  - `P_DIAM0(p)` **Diameter**
  - `P_VEL0(p)[i]` **Velocity**
  - `P_T0(p)` **Temperature**
  - `P_RHO0(p)` **Density**
  - `P_MASS0(p)` **Mass**
  - `P_TIME0(p)` **Time**
  - `P_LF0(p)` **Liquid fraction**
- **Values of particle properties at injection into domain**
  - `P_INIT_DIAM(p)` **Diameter**
  - `P_INIT_MASS(p)` **Mass**
  - `P_INIT_RHO(p)` **Density**
  - `P_INIT_TEMP(p)` **Temperature**
  - `P_INIT_LF(p)` **Liquid fraction**

DPM Macros (3)

– P_EVAP_SPECIES_INDEX(p)	Evaporating species index in mixture
– P_DEVOL_SPECIES_INDEX(p)	Devolatilizing species index in mixture
– P_OXID_SPECIES_INDEX(p)	Oxidizing species index in mixture
– P_PROD_SPECIES_INDEX(p)	Combustion product species index in mixture
– P_CURRENT_LAW(p)	Current law index
– P_NEXT_LAW(p)	Next particle law index

DPM Macros (4)

- **Material Properties for particles**
  - **P_MATERIAL(p)** — **Material pointer for particles**
  - **DPM_SWELLING_COEFFI(p)** — **Swell coefficient for devolatilization**
  - **DPM_EMISSIVITY(p)** — **Particle radiation emissivity**
  - **DPM_SCATT_FACTOR(p)** — **Particle radiation scattering factor**
  - **DPM_EVAPORATION_TEMPERATURE(p)** — **Evaporation temperature**
  - **DPM_BOILING_TEMPERATURE(p)** — **Boiling temperature**
  - **DPM_LATENT_HEAT(p)** — **Latent Heat**
  - **DPM_HEAT_OF_PYROLYSIS(p)** — **Heat of pyrolysis**
  - **DPM_HEAT_OF_REACTION(p)** — **Heat of reaction**
  - **DPM_VOLATILE_FRACTION(p)** — **Volatile fraction**
  - **DPM_CHAR_REACTION(p)** — **Char fraction**
  - **DPM_SPECIFIC_HEAT(p, t)** — **Specific Heat at temperature t**

DPM Functions (1)

- **The following interactions on particles can be customized by UDFs:**
  - **Body force - custom body forces on the particles**
  - **Drag - user defined drag coefficient between particles and fluid**
  - **Source Terms - access particle source terms**
  - **Output - user can modify what is written out to the sampling plane output**
  - **Erosion - called when particle encounters "reflecting" surface**
  - **DPM Law - custom laws for particles**
  - **Scalar Update - allows users to update a scalar every time a particle position is updated**
  - **Switch - change the criteria for switching between laws**

DPM Functions: An Example

- The function below models a custom law
- "p" is a pointer to data structure of type Tracked_Particle

```
#include "udf.h"
#include "dpm.h"
DEFINE_DPM_LAW(Evapor_Swelling_Law, p, ci)
{
  real swelling_coeff = 1.1;
  /* first, call standard evaporation routine to calculate
  mass and heat transfer */
  Vaporization_Law(p);
  /* compute new particle diameter and density */
  P_DIAM(p) = P_INIT_DIAM(p)*(1. + (swelling_coeff - 1.)*
              (P_INIT_MASS(p) P_MASS(p))/
              (DPM_VOLATILE_FRACTION(p)*P_INIT_MASS(p)));
  P_RHO(p)  = P_MASS(p) / (3.14159*P_DIAM(p)
                           *P_DIAM(p)*P_DIAM(p)/6);
  P_RHO(p)  = MAX(0.1, MIN(1e5, P_RHO(p)));
}
```