```
model WedgeMassFlowHopper
  extends Icons.MassFlowHopper;
  //Geometric Parameters
  parameter Integer n = 100;
  parameter Types.Volume V_bin = 1;
  parameter Types.Area A_bin = 1;
  parameter Types.Length H_bin = V_bin / A_bin;
  parameter Real dx = H_bin / (n - 2);
  parameter Types.Temperature T_s_0 = 20 + 273.15;
  //Particle Variables
  Types.MassFlowRate m_dot_s_in;
  Types.MassFlowRate m_dot_s_out;
  Types.Temperature T_s_in;
  Types.Temperature T_s[n](each start = T_s_0);
  Types.Temperature T_s_out;
  Types.Enthalpy h_s_in;
  Types.Enthalpy h_s_out;
  Types.Velocity v_s;
  Types.Density rho_s;
  Types.SpecificHeatCapacity cp_s;
  parameter Real phi_s = 0.6;
  FallingParticleReceiverSystem.Interfaces.ParticleFlow
ParticleOutlet annotation(
    Placement(visible = true, transformation(origin = {0, -100},
extent = {{-10, -10}, {10, 10}}, rotation = 0),
iconTransformation(origin = {0, -100}, extent = {{-10, -10},
{10, 10}}, rotation = 0)));
  FallingParticleReceiverSystem.Interfaces.ParticleFlow
ParticleInlet annotation(
    Placement(visible = true, transformation(origin = {0, 98},
extent = {{-10, -10}, {10, 10}}, rotation = 0),
iconTransformation(origin = {0, 100}, extent = {{-10, -10}, {10,
10}}, rotation = 0)));
equation
//Connection
  m_dot_s_in = ParticleInlet.m_dot;
  m_dot_s_out = -ParticleOutlet.m_dot;
  T_s_out = ParticleOutlet.T;
  T_s_in = ParticleInlet.T;
//Mass Balance
  m_dot_s_in = m_dot_s_out;
  m_dot_s_in = rho_s * phi_s * v_s * A_bin;
//Energy Balance
  T_s[1] = T_s_in;
  for i in 2:n loop
    cp_s * rho_s * phi_s * der(T_s[i]) = -v_s * rho_s * phi_s *
cp_s * (T_s[i] - T_s[i - 1]) / dx;
```

```
  end for;
  T_s[n] = T_s_out;
//Property Evaluation
  h_s_in = Media.Particle.Enthalpy(T_s_in);
  h_s_out = Media.Particle.Enthalpy(T_s_out);
  cp_s = Media.Particle.SpecificHeat(T_s_out);
  rho_s = Media.Particle.Density();
end WedgeMassFlowHopper;
```