Goals

determine what metrics have to be calculated for both

- Reynolds number
- Nusselt number

Show values for both of them.

Find how to make them equal

Dynamic similarity is the requirement that if the two objects are geometrically similar, then they also have similar flow patterns, i.e., the velocities and velocity gradients, fluid forces and streamlines all scale with the geometry.

For heat transfer problems, we will also require that the temperatures, temperature gradients and heat fluxes scale with the geometry.

Ansys introduction to heat transfer

- As in fluid dynamics, formal methods can be used to determine the relevant dimensionless groups for convection heat transfer.
  - The Nusselt number is a non-dimensional representation of heat transfer.
  - The Prandtl number represents the relative sizes of the thermal and velocity gradients.
  - The Reynolds Number is a non-dimensional representation of the fluid dynamics (ratio of inertial to viscous forces).
- Could a relationship between the Nusselt number and Reynolds and Prandtl numbers be found? That is:

$$Nu = F(Re, Pr)$$

- It turns out, the results for a given fluid (i.e., fixed Pr) fall close to a straight line on a $\log - \log$ scale, and the expression for a global Nusselt number can be represented by an empirical correlation:

$$\overline{Nu}_L = C Re^m Pr^n$$

- Specific values of C , m and n often are independent of the fluid, but they depend on the geometry of the surface and flow type.

- For the case of forced convection, two geometrically similar systems are dynamically similar if their Nusselt, Reynolds and Prandtl Numbers are the same.
  - Therefore, if an experiment is run with a model geometry at given Reynolds and Prandtl numbers to determine a Nusselt number, then the Nusselt number results can be applied to a scaled-up version of the model provided the Reynolds and Prandtl numbers are the same.

- The Reynolds Number $Re = \frac{\rho V L}{\mu}$

- The Prandtl Number $Pr = \frac{C_p \mu}{k} = \frac{\nu}{\alpha}$

```
 dynViscosity = py.CoolProp.CoolProp.PropsSI('V', 'T', 296.15, 'P|not_imposed',
101325, 'air')
```

```
dynViscosity = 1.8351e-05
```

```
    densityAir = py.CoolProp.CoolProp.PropsSI('D', 'T', 296.15, 'P|not_imposed',
101325, 'air')
```

```
densityAir = 1.1923
```

```
    thermCond = py.CoolProp.CoolProp.PropsSI('L', 'T', 296.15, 'P|not_imposed',
101325, 'air')
```

```
thermCond = 0.0261
```

```
    ConstPNum = py.CoolProp.CoolProp.PropsSI('C', 'T', 296.15, 'P|not_imposed',
101325, 'air')
```

```
ConstPNum = 1.0062e+03
```

```
    thermMassAir = py.CoolProp.CoolProp.PropsSI('C', 'T', 296.15, 'P|not_imposed',
101325, 'air')
```

```
thermMassAir = 1.0062e+03
```

```
    KinViscosity = dynViscosity/densityAir;

 dynViscosity = py.CoolProp.CoolProp.PropsSI('V', 'T', 299.15, 'P|not_imposed',
101325, 'air')
```

```
dynViscosity = 1.8496e-05
```

```
    densityAir = py.CoolProp.CoolProp.PropsSI('D', 'T', 299.15, 'P|not_imposed',
101325, 'air')
```

```
densityAir = 1.1803
```

```
    thermCond = py.CoolProp.CoolProp.PropsSI('L', 'T', 299.15, 'P|not_imposed',
101325, 'air')
```

```
thermCond = 0.0263
```

```
    ConstPNum = py.CoolProp.CoolProp.PropsSI('C', 'T', 299.15, 'P|not_imposed',
101325, 'air')
```

```
ConstPNum = 1.0063e+03
```

```
    thermMassAir = py.CoolProp.CoolProp.PropsSI('C', 'T', 299.15, 'P|not_imposed', 101325, 'air')
```

```
thermMassAir = 1.0063e+03
```

```
    KinViscosity = dynViscosity/densityAir;

  speed1 = 1;
  length1 = 1;
  length2 = 1.05;
  speed2 = 1/length2;


  Re1 = densityAir*speed1*length1
```

```
Re1 = 1.1803
```

```
  Re2 = densityAir*speed2*length2
```

```
Re2 = 1.1803
```

```
  Pr1 = ConstPNum * dynViscosity*thermCond
```

```
Pr1 = 4.8994e-04
```

V	Dynamic viscosity [Pa-s
L	Thermal conductivity [kW/m/K]
D	Density [kg/m3]
C	Specific heat at constant pressure [kJ/kg/K]

PRANDTL, Prandtl		O	False	Prandtl number