DTU

Bulk flow energy equation

Table of Contents

Refer to separate notes for development. Below is dimensionless, steady form for incompressible fluid.

$$
\begin{aligned}
c_p \frac{\partial (\rho h v_x T)}{\partial x} + c_p \frac{\partial (\rho h v_y T)}{\partial y} = {} & \frac{R}{C}\left[h_s (T_s - T) + h_r (T_r - T)\right] \\
& + [E_c] \frac{R\Omega}{u_*} \frac{h}{2} \frac{\partial p}{\partial y} + \left[\frac{R}{C} E_c\right] \left\{ \frac{R\Omega}{u_*} \frac{a\rho}{4} v_y^2 f_s \right. \\
& - \frac{R\Omega}{u_*} \frac{a\rho}{4} \left(v_y - \frac{R\Omega}{u_*} \right)^2 f_r + 0.5\rho v_x^2 f_r v_r \\
& + 0.5\rho v_x^2 f_s v_s + 0.5\rho v_y^2 f_r v_r + 0.5\rho v_y^2 f_s v_s \\
& \left. - 0.5\rho v_y \frac{R\Omega}{u_*} f_r v_r \right\}
\end{aligned}
\tag{1}
$$

Table of Contents

Required changes/additions to code

1. Formation of linear system
   - Formulation of coefficient matrix parallels momentum transport, i.e. mass flux carries enthalpy ($c_p T$) across cell faces
   - Source terms considered constant over CVs.
2. Solution of energy equation after momentum and pressure correction
3. Dirichlet boundary condition at inflow, outflow is zero gradient
4. Additions to input file

Coefficient matrix

`_setup_zeroth_energy()` method added to seal class

```
for i in range(self.Nfint):
  p = owner[i]
  nb = neighbor[i]
  # mixed upwind/linear
  fluxp = phi[i] * ((1. - self.gamma) * 0.5 * (np.sign(phi[i]) + 1.) + self.gamma * gf[i])
  fluxnb = phi[i] * ((-1. + self.gamma) * 0.5 * (np.sign(phi[i]) - 1.) + self.gamma * (1. - gf[
      i]))
  # owner
  self.At[p, p] += fluxp
  self.At[p, nb] += fluxnb
  # neighbor
  self.At[nb, p] += -fluxp
  self.At[nb, nb] += -fluxnb
# convective fluxes at inflow and outflow
idx = 0
idx2 = 0
for i in range(self.Nfstart[0], self.Nf):
  p = owner[i]
  if self.bc_type[idx] == 1:  # inflow
    self.bt[p] += - phi[i] * self.tbc[idx]   # convective flux
  if self.bc_type[idx] == 2:  # outflow
    self.At[p, p] +=  phi[i]
  if self.bc_type[idx] == 0:  # solid wall
    pass
  if self.bc_type[idx] == 4:  # cyclic 2
    nb = cyclic[idx2, 2]
    # convective
```

Source terms

again, within `_setup_zeroth_energy()` method

```
U_r, U_s, f_r, f_s = self._compute_friction(self.u, self.v)

m = self.rotor_m

# source terms
for i in range(self.Nc):
  self.bt[i] += 0.5 * self.Ec * self.R * self.v_r * self.hc[i] * self.grad_p[i,1] * self.cell[i
      , 2]
  self.bt[i] += (self.R / self.C) * self.Ec * self.rho[i] * (0.25 * self.v_r  * self.v[i] ** 2.
      * f_s[i] -\
          0.25 * self.v_r  * (self.v[i] - self.v_r)**2 * f_r[i] + \
          0.5 * self.u[i] ** 2. * f_r[i] * U_r[i] + \
          0.5 * self.u[i] ** 2. * f_s[i] * U_s[i] + \
          0.5 * self.v[i] ** 2. * f_r[i] * U_r[i] + \
          0.5 * self.v[i] ** 2. * f_s[i] * U_s[i] - \
          0.5 * self.v[i] * self.v_r * f_r[i] * U_r[i] ) * self.cell[i, 2]
  # convective ht with rotor and stator surfaces
  if self.ht_flag:
    self.bt[i] +=   (self.R / self.C) * ( self.h_s * (self.T_stator - self.t[i]) +\
          self.h_r * (self.T_rotor - self.t[i]) ) * self.cell[i, 2]
```

Solve energy equation

added to `solve_zeroth()` method after momentum and pressure correction

```
# implicit under-relaxation
if self.relax_mode == 'implicit':
  self.u = self.u_star - self.Dp * self.grad_p_corr[:,0]
  self.v = self.v_star - self.Dp * self.grad_p_corr[:,1]
# explicit under-relaxation
elif self.relax_mode == 'explicit':
  self.u = ( 1. - self.relax_uv ) * self.u + self.relax_uv * ( self.u_star - self.Dp * self.
      grad_p_corr[:,0] )
  self.v = ( 1. - self.relax_uv ) * self.v + self.relax_uv * ( self.v_star - self.Dp * self.
      grad_p_corr[:,1] )

self.press = self.press + self.relax_p * self.p_corr
#print(self.grad_p_corr[:,0])

self._correct_phi(self.phi, self.rho, self.rhobc, self.p_corr, self.p_corr_bc)

self._update_zeroth_bcs()

#energy
self._setup_zeroth_energy()
self.t = spsolve(self.At.tocsr(), self.bt)
```

BCs

```python
def _init_zeroth_bcs(self):
  '''
  '''
  idx = 0
  # outflow (right)
  for i in range(self.Nfstart[1], self.Nfstart[2]):
    self.ubc[idx] = 0.0
    self.vbc[idx] = 0.0
    self.pbc[idx] = self.p_e
    self.rhobc[idx] = self.rho_init
    self.tbc[idx] = self.T_o
    idx += 1
  # inflow (left)
  for i in range(self.Nfstart[3], self.Nf):
    self.ubc[idx] = self.u_i
    self.vbc[idx] = self.v_i
    self.pbc[idx] = self.p_i
    self.rhobc[idx] = self.rho_init
    self.tbc[idx] = self.T_i
    idx += 1
```

input file

See `test09.py`

```
# heat transfer stuff
energy_mode : 1              # 0 : no energy, 1 : debug,
                             # 2 : incomp. w/o ht, 2 : incomp. w/ ht
T_s : 300.0                  # temperature scale [K]
T_i : 300.0                  # inflow temperature [K]
T_o : 300.0                  # outflow temperature [K]
c_p : 4180.0                 # specific heat at constant pressure (isobar
h_r : 30000.0                # rotor convective ht coefficient, [W/(m^2-K
h_s : 30000.0                # rotor convective ht coefficient, [W/(m^2-K
T_rotor : 1000.0             # rotor temperature [K]
T_stator : 1000.0            # stator temperature [K]
e_tol : 1.0e-7               # tolerance threshold on energy equation
```

Table of Contents

Kanki and Kawakami "seal 1" geometry

Geometry used for testing passive energy equation implementation

- $C/R = 0.005$
- $\Omega = 2000$ [rpm]
- **"Long", $\mathbf{L/R = 2.0}$**
- Turbulent flow, $Re_a = 16,707$, $Re_\Omega = 11,890$
- Water (incompressible flow)
- Non-isothermal, constant specific heat $c_p = 4180$ [J/(kg-K)]
- Blasius friction factor using $n = 0.079$ and $m = -0.25$
- $\omega/\Omega = [0.0, 0.12, 0.23, 0.36, 0.48, 0.60]$ (nominal)
- $\xi_i = 0.2$ inlet loss, $\beta = 0.2$ inlet swirl ratio
- Static eccentricity ratio $\varepsilon_x = 0.5$

Baseline temperature profiles, passive transport

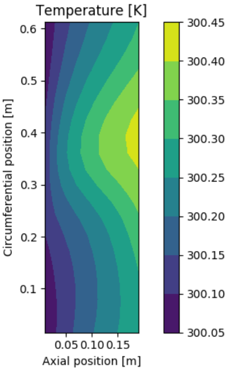

Figure: Convective heat transfer inactive

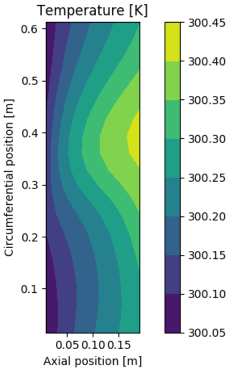

Figure: Convective heat transfer active, but rotor and stator wall temps. same as inlet temp. (Baseline)

Effect of cold vs. hot rotor/stator walls

$h_r = h_s = 30,000$ [W/(m^2-K)]

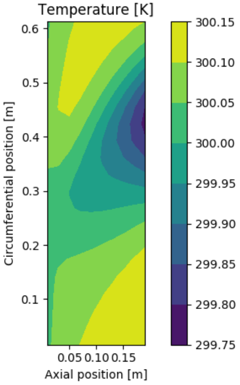

Figure: Cold, 100K

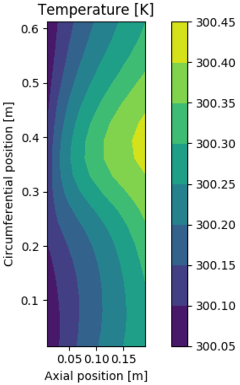

Figure: Baseline

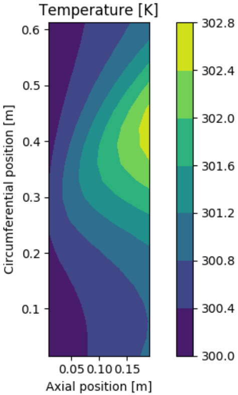

Figure: Hot, 1000K

Comments

- Temperature profiles appear qualitatively correct
- Correct trends observed for hot vs. cold wall settings
- Overall, convective heat transfer to rotor and stator walls for this seal is minimal. It requires very high convective heat transfer coefficient and/or very high/low wall temperatures to initiate appreciable change in the fluid temperature.

Table of Contents

In-progress

Additions for compressible flow zeroth-order solution

- Compressibility terms in energy equation
- Density update (function of temperature and pressure) at start of PISO loop
- Density corrections added to pressure correction equation. Results in mixed elliptic/hyperbolic character of pressure correction equation.
- BCs for different flow scenarios (subsonic, supersonic)

Also, addition of density perturbations to continuity and momentum equations to solve for first-order problem considering compressible flow. Energy equation not perturbed.