**Raphael modeling in MSPM**

The Fin Enhanced Surface HX issue:

* MSPM HX fix:
* Add Lf value in matrix.m
* Sara: Still does not work, cannot change HX temperatures in UI.

Meeting Oct 08:

* Look at nodes network and compare to EP1 / Beta model to reduce runtime
* DCH HX annulus dimensions: height 6.6mm, inner radius 35mm, outer radius 88.2mm. Const. temp. body with hot source temp.

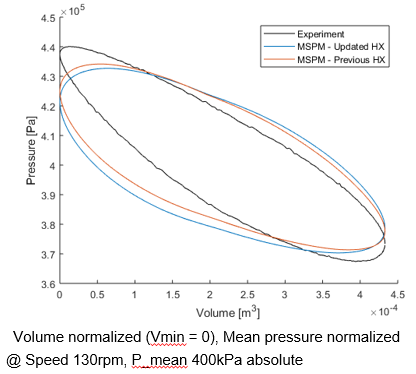
Oct14:

* Comparing MPSM to Exp in terms of indicated work shows:
  + Consistent trend of decreasing W\_ind with increasing speed
  + Good agreement in abs. values for Pmean~480kPa
  + MSPM predicts smaller W\_ind than Exp for lower Pmean
  + **🡪 MSPM underestimates heat flow capacity of HX?**
  + Little change of W\_ind over Pmean in Exp 🡪 likely close to optimum Pmean for HX
  + **🡪 MSPM does not predict HX performance reaching their max in this range of Pmean**

Nov 02:

* In MSPM, HX source channel cover entire length of HX. In reality, 36mm of the 96mm **(3/8)** length are not covered by water jacket. Also varying wall thickness for another 2x5mm.
* Model will overestimate heat flow capacity.
* **Model HX as separate bodies** with the ones not covered by source as room temperature

Nov 05:

* Results with changed HX geometry (including water jacket):
  + 
  + A) **Overall Delta-P under-estimated** by model, but B) **same-volume delta-P** (PV shape height) **over-estimated**
  + Could mean: A**) heat exchanged underestimated**, B) **flow friction / pressure drop underestimated**

Nov 09:

* Maximum deviation between PP volume data from exp and mspm, normalized to Vmin=0:  
  **29.7 cm^3.**

Nov 23:

* Temperature Sensors in Model:
  + No difference between Tge\_exp and Tge\_exp\_center\_test (same body, one node)
  + Max difference **11C between Tgh\_reg and Tgh\_reg\_TopOfReg\_test! (**heater bottom/ reg top)
  + Max difference **3C between TgPP and TgPP\_PPtop\_test!** (bottom conn. Pipe/ top PP space)
  + **Tgc and TgPP are equal!**

Nov26:

* Add materials: DONE
  + Plastic insulation parts (plates and reg housing) should be PEI (Polyetherimide) (current: Polyethylene)
  + Reg material should be polyester (current: Polyethylene)

Jan 13:

* Current Model computing time: ~10min per setpoint

Jan 14:

Gas temps (p200):

* + - **Hot: Model much lower (15-20C!)**
    - Reg: **Model much lower (~10C).** only calculated Exp temps (log mean).
    - Cold:
      * For all temps except below: **Model much lower, (6-12C)**
      * TgPP: model higher at low speed (up to 4C), model lower at high speed (up to 2C)
      * TgCC: model higher (~5C)

Feb 15

* Diagram

  Description automatically generated with low confidence
* Model with Fin Enhanced Surface HX gives much too high heat flow, W\_ind results.
* Suspected problem: No heat transfer resistance from convection between source and wall is present in model. Only conduction from const. temp. body through wall.
* 🡪 implement an optional resistance to input into a face between two nodes, or add insulating body around source body to mimic the convection resistance
* Resistance, conduction:
* Resistance, oil convection: (h estimated to 79.3 W/m^2 K)
* Resistance, water/glycol convection: (h estimated to 383.2 W/m^2 K)
* (applies to all of the fin area! Is about 16x the area without fins)
* (Area on liquid side about 16x higher with fins than without.)
* (with full ), (with /16)
* Conduction is <0.5% of total resistance (with /16)
* With new liquid fins, Conduction is 2.2% of total resistance (with /16 and /16)
* k(air) = 30 mW/m K, h(air) = 130 W/m^2 K
* **🡪 this should change results significantly**
* Surround source with solid ‘SIL180 oil’ material (k=0.1)
* 🡪 conduction length equivalent to convection resistance:
  + **Oil:** 0.00126 m
  + **Water/glycol**: 0.000261 m
* Chart, histogram

  Description automatically generated Runtime: 1h 20min!
* Suspecting this may be due to the very thin node that is the cooler insulation. 🡪 use higher conductivity material that will lead to insulation that fills nearly the entire source body.
* Made k customizable for oil material in MSPM
* Make insulation thickness equal to 1/3 of source body thickness, which is 0.0016m = L\_cond
* Desired = 0.127 (oil) / 0.613 (water-glycol)
* 🡪 Result: Not much difference, runtime still upwards of 4000s.

Feb18:

* Instead of insulating bodies, try to implement a **custom convection resistance** into code that can be added to the faces between a const. temp. body and its neighbor bodies.
* Face.m:
  + When face is created, property ‘U’ is assigned. Comparing equations for ‘U’ to Steven’s thesis, ch. 2.2.2 about faces, ‘U’ seems to be the inverse of heat transfer resistance ‘R’, which is the conductance.
  + Graphical user interface, text, application

    Description automatically generated  
    (this function is used for ‘simple’ faces such as in matrix. (Function calcData)
  + For other faces, the face constructor function is called with inputs that represent NodeContacts:   
    Text

    Description automatically generated
  + Gas-Solid (Mixed) faces have a property ‘R’ (resistance):  
    
  + Solid-solid and environment-solid faces have ‘U’:  
    
  + Goal is to modify conductance. 🡪 look into ‘getConductance’ function.
* NodeContact.getConductance
  + Calculates conductance ‘U’ for face between NodeContacts NC1 and NC2 according to thesis equations 28, 29
* Goal: add custom heat transfer resistance to properties of a solid body that will be applied to conductance of each of its faces with other sold bodies. (i.e. solid faces of nodes of this body)

Feb 22:

* NodeContact.getConductance
  + Added code that for static solid faces includes am additional heat transfer coefficient ‘h\_custom’ (works analogous to convective coefficient ‘h’) into the conductance calculation if a parent body has such coefficient, located in ‘Body.h\_custom’.
* Need to add code to display ‘h\_custom’ in dropdown box and to change it.
  + In ‘Body.m’, add ‘get’ and ‘set’ functionality for ‘h\_custom’
  + In ‘ListObj.m’, function ‘getObjs’: Add entry for ‘h\_custom’ to be displayed as a ListObj for all Bodies. Use ListObj of type ‘Editnum’. Add condition to function ‘on\_click’ for ‘Editnum’ that allows NaN value in case of ‘h\_custom’.

Feb 24:

* Investigating the cause of the long runtimes with the models that have HX insulation or custom heat transfer coefficient representing source channel convection
* Is the ‘convergence acceleration’ (steven thesis ch. 5.1) working in all runs?
  + Simulation.Run: The solid temperature acceleration operations (thesis eq. 101,102) seem to be active when ‘ME.continuetoSS’ is false.
  + ‘ME.continuetoSS’ is true in the final cycle in which the acceleration is not supposed to be used.

Feb 25:

* Investigating change in model result after considered steady state by model (ca. 6 cycles for p200, A22) and running it longer (ca. 40 cycles)
* Power changed from 7.1891 to 7.4791
* W\_ind from 5.0756 to 5.1854 (experiment: 10.6534)
* Change in W\_ind amounts to 2.8% of deviation from experiment. 🡪 insignificant compered to model deviation at this stage

**May 27: Fixing the application of Source/Sink temps to Nodes. Also traced and checked Source/Sink heat flow calculations.**

* PROBLEM: Source and Sink temperatures from RunConditions are applied to the wrong Nodes. ME.Sources and ME.Sinks is assigned incorrectly.
  + (For model ‘Raphael\_2022-05-26\_FinEnhSurf\_Custom\_h\_Default’)
  + Current: ME.Sources=103, ME.Sinks=62
  + Actual: Source body = 34, Node.index 103. Sink body = 42, node.index 62.
  + BUT: Node 103 has index 1026. Node 62 has index 1067. Indexes are descending in order while node numbers ascending.
  + 🡪 indexes of Nodes match with ME.Sources/Sinks, but not with Node position in Node array!
  + Source Node is position 1069, Sink is position 1110
  + Last Node is Env node, index 51.
  + Node indices, from back of array, start with env node (51), then 52 to 317, 1 to 16, 318 to 323, 17 to 22, 324 to 371, 23 to 40, 372, 41 to 43, 373 to 1092, 44 to 49, 1093 to 1109, 50, 1110 to 1121 (position 1)
  + Are they sorted by the bodies they belong to?
    - Discretization is Body by Body, but indexes are not all in order.
* 🡪 Where are ME.Source and Sink defined? In Model.discretize.
* **DO we just need to change the assignment of SourceTemp and SinkTemp to look for indexes?** A screenshot of a computer

  Description automatically generated with medium confidence
* Values of assignment arrays ME.Regions, Sources, Sinks refer to indexes and NOT to positions in array!
* What is ME.BoundaryNodes? all nodes on the boundary of bodies , that have mixed faces.
* After Groups and all other elements have discretized, all nodes are collected to form the Simulation.Nodes array. Staring with environment node, then groups, backwards from end of array to position 1, thus env node is always last in array.
  + Each group, when discretizing, discretizes each body (in order of this.Bodies). Then all Nodes are collected into this.Nodes (this being the group) in ascending order, same order as bodies.
* 🡪 Nodes (and connections) can be in any random order in array. Identification should be done using the ‘index’ property.
* isSourceOrSink follows Node.index (is true at 62 and 103)
* SubjectNodes = [62 103]
* Equally, SubjectFaces contains ‘index’ property of faces connecting to Source/Sink
* IsSource is true at 103
* Sources = 103, Sinks = 62
* **🡪 Need to change assignment of SourceTemp and SinkTemp as proposed above.**
  + Actually, the values in ME.T at the source and sink index positions appear to be source and sink temps already, but they don’t change if changed in the model through the UI. 🡪 These are prob. The values from the snapshot that are loaded as initial conditions.
  + ME.T is sorted by Node ‘index’ and initial source/sink temp assignment is correct.
  + Problem is in the code for solid temp acceleration:
    - Text

      Description automatically generated
    - ‘var’ contains the original source/sink temps from the model. Here the previously assigned ones from the RunConditions are overwritten. Vector ‘b’ also contains the old temps. So does ME.bCond. where does this one come from?
    - ‘b’ and ‘bCond’ contain constant temperatures for nodes that don’t change temperature.
    - bCond assigned in Model.discretize: Graphical user interface, application, Word

      Description automatically generated
    - For all nodes with dT\_du<1 (const temp and insulator), bCond assigned from **Node.data.T**
    - ‘crun’ can have fields ‘SourceTemp’ and ‘SinkTemp’, or not have them (manual run).

**Checking correct assignment of Heat flows:**

* Sim.ToSource and ToSink contain face ‘index’
* In Simulation.implicitSolve:
  + Face ‘index’ values from ToSource/Sink become facesSr/facesSi
  + Then heat flows are added up like:
  + Text

    Description automatically generated
  + 🡪 is ‘Qfc’ sorted by face position or ‘index’? Would be correct if sorted by ‘index’
  + Largest heat flows are at positions 36 to 370 in ‘Qfc’. The Faces with these indexes are all mixed faces between gas and solid nodes of the regenerator. It makes sense that the highest heat flows would be inside the regenerator. This indicates that ‘Qfc’ is sorted by face ‘index’.
  + **🡪 Source and Sink heat flows are most likely calculated correctly.**
  + Where is Me.dt\_max calculated? Text

    Description automatically generated

# ‘Wall\_smart\_Discretize’ function

Solid bodies:

* Discretizes based on
  + Minimum engine speed (from ‘Body.Group.Model.engineSpeed’)
  + Mesher settings: oscillation\_depth\_N, maximum\_thickness, maximum\_growth
* Uses the “5% amplitude condition” for oscillation depth:  
  xdepth = 3\*sqrt(2\**diffusivity*/*min\_engine\_speed*);

Gas bodies:

* Based on mesher settings: Gas\_Entrance\_Exit\_N, maximum\_growth, Gas\_Maximum\_Size, Gas\_Minimum\_Size

**Different engine speeds lead to different mesh in solid bodies!**

* Can disable this and keep mesh constant based on speed set in model file by commenting out this line in ‘Model.discretize’ (ca. line 510):   
  this.engineSpeed = run.rpm;
* HIGHER speed will lead to finer mesh, MORE nodes (because osc. Depth smaller)!

# ‘RunConditions’ fields

* can be used to modify run parameters
* List in Steven’s thesis (ch. 6.1.7.2) is complete, lists all fields found in code

## In ‘Model.Run’:

'Uniform\_Scale'

* scales entire geometry physically in size by a factor.

## In ‘Model.discretize’:

‘NodeFactor’

* Multiplies number of nodes
* Is applied in initial part of ‘Model.discretize’
* There, see code section “%% IMPORTANT for 'NodeFactor'”:
* The node factor by default only applies to:
  + All bodies without Discretization Function enabled in model (gas nodes can only be discretized in one dimension, so the factor only applies to the direction that has ‘Divisions’ > 1)
  + Gas nodes discretized with Discretization Function
* By default does NOT apply to Solid nodes with Discretization Function. To do this, node factor has to be applied to ‘oscillation\_depth\_N’ and ‘maximum\_thickness’ (currently commented out)

'HX\_Convection' (default = 1)

* is applied as factor to all Nusselt number equations for all solid nodes in a gas body with a matrix component that has 'SourceTemperature' property (i.e. all HX matrix solid nodes)
* scales convection coefficient h of all HXs

'Regen\_Convection'

* Same as 'HX\_Convection' but for gas bodies with matrix without source (i.e. Regenerator)

'Outside\_Matrix\_Convection'

* Same as 'HX\_Convection' but for gas bodies without matrix (i.e. all other convection that is not in a matrix)
* CODE NEEDS FIXING before use! Has ‘HX\_Convection’ in it, probably error

‘Friction’

* Factor for friction factor equations for all gas and MatrixTransition faces

'Solid\_Conduction'

* Factor for conductivity of all solid and mixed faces

'Axial\_Mixing\_Coefficient'

* Factor for axial mixing coefficient (NkFunc) equations of all gas and MatrixTransition faces

‘HX\_C1’ … ‘HX\_C4’ and ‘HX\_SA\_V’

* are used if there is a heat exchanger matrix of type ‘Custom HX’

'Regen\_C1' … 'Regen\_C4', 'Regen\_Porosity’, ‘Regen\_SA\_V’

* are used if there is a matrix of type ‘CustomRegen’

'h\_custom\_Factor' (Added by Matthias)

* Scales the custom convection coefficient ‘h\_sustom’ which imitates a fixed convection heat transfer resistance between bodies

## For Optimization:

***PressureBounds*** and ***SpeedBounds***

# limit the search area for gradient ascent algorithm

# May 26: Sensitivity Study – Potential Parameters

1. Conduction (to env. And between HXs)
   * Different levels of conduction enabled by making bodies insulators
2. Mesh
   * Have certain bodies discretized vs. not
   * HX node density
   * Overall mesh refinement from baseline mesh using ‘NodeFactor’
3. Source/Sink temperatures
4. h\_custom
   * Linda’s analytical and CFD results! 🡪 What is realistic range of h?
5. Regenerator
   * Porosity
   * Wire diameter
6. Appendix gap
   * Have vs. not
7. PP seal
   * Leak: need to measure leak mass flow rate vs. dP. Could calculate combined leak mass flow rate of PP seal and DP rod bushing from leak test data
   * Friction: would probably need to be implemented in mechanism

# Sensitivity Results

1. **Conduction**

W\_ind:

* Overall: All Wind from model are ~50% smaller than exp
* **Major effect:**
  + **No conduction through HX side wall (-40% Wind).**
  + **No conduction through base / displacer mount: -10%**
* Small effect:
  + No axial HX conduction through Regenerator PEI parts: +2.5%
  + No DCH conduction (all 2 plates): +3.8%
* No effect (<0.6%):
  + DP conduction
  + Adding PP and CC conduction

Heat Flows:

* Exp: Heater and Cooler both ~850W. All model data is ~30% below exp. But exp data has high uncertainty in heater heat flow.
* No Axial HX Conduction:
  + Small (-2%) decrease in source, sink, env heat flow. Equal increase in Wind
* No DP conduction:
  + NO effect except very small (<0.6%) decrease in Wind, Eff)
* No HX Side conduction:
  + Large decrease Wind (-42%), Source (-33%), Sink (-24%), Env (-46%).
  + Decrease in Eff (-14%), i.e. power output decreased more than heat input.
  + Somehow the HX insulation leads to much lower temps on hot side and higher on cold side. Does Perfect insulator body act as heat source/sink?
* No top (DCH) conduction:
  + Small decrease Source (-2%), increase sink (1%). Large decrease Env (-60%)
  + Results in 6% boost to Eff, 4% to Wind
* Added PP CC conduction:
  + 6% increase to Env. Otherwise negligible.
* No Base conduction (displacer mount):
  + Decrease to Sink (-4.6%), small decrease Source (-2%).
  + Increase Env (10%). Looks like the default model TAKES heat from environment through DM, thus this increase when DM conduction disabled.
  + -10% decrease to Eff, -11% to Wind.

Gas Temps:

* Are much too low in model! 30C on hot side, 10C on cold side.

# H\_custom for liquid convection

**Matthias, from correlations:**

* Resistance, oil convection: (h estimated to 79.3 W/m^2 K)
* Resistance, water/glycol convection: (h estimated to 383.2 W/m^2 K)
* Ratio 4.83

Calculations:

* Assume jacket is a rectangular channel flow
* W\_channel = 4.8mm
* H\_channel (MSPM) = 55mm
* H\_channel (area equivalent) = 55.8mm
* D\_h = 8.84mm
* Rho\_sink = 1054 kg/m3 (0 degC)
* rho\_SIL180 = 932 kg/m3 (at 21.5 C)
* dyn\_visc\_SIL180 = 2 cSt = 2 mm2/s = 2E-6 m2/s (at 200 C)
* Mu\_sink = 1.7 centipoise (26.7 degC) = 1.7\*0.01\*0.1 Pa s = 0.0017 Pa s
* m\_dot\_Cooler = 0.0235576 kg/s
* m\_dot\_heater = 0.054598534 kg/s (at 150C)
* U\_cooler = 0.08346 m/s
* U\_heater = 0.2187 m/s
* **Re\_cooler = 457**
* **Re\_heater = 967**
* Nu(ts=const) = aprx. 7
* k\_SIL180 = 0.1 W/m K
* **h\_source = 79.3 W/m2 K**
* k\_sink = 0.484 (to produce below h\_sink)
* **h\_sink = 383.2**

**Linda, from CFD results:**

* = 0.0679 K/W 🡪 = 281.3 W/m2 K
* = 52324 mm2
* = 0.026 K/W 🡪 = 735.065 W/m2 K
* Ratio 2.62
* Wind(exp) / Wind(default model) = 2.14
* According to sensitivity study, changing to Linda’s value would increase by factor 2.63.
* Also, changing to Linda’s value would increase by factor 1.23.
* BUT, is not expected to increase linearly over such wide range of h. 🡪 result could be close to experimental .