CON1D Mold Geometry Calibration: “Offset Method”

Inwho Hwang
(BSME Student)

Lance C. Hibbeler
(Ph.D. Student)

Department of Mechanical Science and Engineering
University of Illinois at Urbana-Champaign

Introduction

• The simplified mold geometry in CON1D can be calibrated using analytical techniques and heat transfer FEM models to provide increased accuracy at practically no cost

• This calibration has been automated using a Python script and ABAQUS
CON1D’s 1D Mold Model

- Given
  - Hot face heat flux $q_{hot}$
  - Cold face convection $h_{cold}$, $T_{water}$
  - Thermal conductivity $k$
- Conduction equation gives:
  \[
  T = T_{water} + q_{hot} \left( \frac{1}{h_{cold}} + \frac{L - x}{k_{mold}} \right)
  \]
- $h_{cold}$ is a function of channel geometry, water properties, and water speed

Water Channel Geometry
Width and Depth

- The simulated rectangular channels and the actual water channels must have identical
  - Cross-sectional area: correct amount of water
    \[w_c d_c = A_{c, actual}\]
  - Hydraulic diameter: correct convection behavior
    \[2w_c d_c / (w_c + d_c) = D_{h, actual}\]
- Two equations and two variables, solved:
  \[w_c d_c = A_{c, actual} / D_{h, actual} \pm \sqrt{\left(A_{c, actual} / D_{h, actual}\right)^2 - A_{c, actual}}\]
  Use average $A_c$ and $D_h$ for the mold
Water Channel Geometry

Pitch

- Water channel pitch is determined by plotting cumulative water channel area as a function of distance from the centerline and performing least-squares
  - Slope used to determine pitch

\[ A_{\text{cumulative, simulated}}(x) = \frac{A_{c,\text{actual}}}{p_c} x \]

Calibrating for 3D Effects

- The accuracy of a 3D finite-element model can be given to CON1D by calibrating the mold thickness and thermocouple locations
- Manipulating 1D temperature solution gives
  - Calibrated cold face position
    \[ d'_{\text{cold}} = \frac{k}{q} \left( T_{\text{hot,3D}} - T_{\text{cold,3D}} \right) \]
  - Calibrated thermocouple position
    \[ d'_{TC} = \frac{k}{q} \left( T_{\text{hot,3D}} - T_{TC,3D} \right) \]

Underlined terms taken from 3D FE model; \( k \) and \( q \) must match model
3D Mold Model

- The script exploits symmetry and models a small piece of the bigger casting mold.

4 Mold Types

- Type 1 has circular channels
- Type 2 has rectangular channels and a deep channel
- Type 3 has rectangular channels and a slanted channel
- Type 4 has rectangular channels and a single circular channel.
General Dimensions: Domain size

Script Input:

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>MoldWidth = 77.5</td>
</tr>
<tr>
<td>12</td>
<td>MoldThickness = 40.0</td>
</tr>
<tr>
<td>13</td>
<td>MoldLength = 90.0</td>
</tr>
</tbody>
</table>

Example mold with bolt rib

ChannelDepthFromHotFace + ChannelDepth = MoldThickness makes a flat back mold

Example mold with flat back

Top

Front

Side

MoldWidth

MoldLength

General Dimensions: Are TC and bolt present?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>19</td>
<td>BoltHolePresent = True</td>
</tr>
<tr>
<td>23</td>
<td>TCHolePresent = True</td>
</tr>
</tbody>
</table>

TCHolePresent

True

False

BoltHolePresent

True

False
General Dimensions: bolt & TC

- BoltHoleDiameter/2
- BoltHoleDepth
- TCHoleDiameterUpper/2
- TCHoleDiameterLower/2
- TCHoleTransitionStart*
- TCHoleTransitionLength*
- TCDepthFromHotFace

Starred (*) values can equal zero.

Type 1 Dimensions

- ChannelStart = 20.0
- ChannelPitch = 15.0
- ChannelDepthFromHotFace = 22.0

* Type 1 input

- ChannelDiameter = 5.0
Type 2 Dimensions

Type 3 Dimensions
Type 4 Dimensions

ChannelDepth

ChannelStart

ChannelPitch

ChannelOuterWidth

ChannelInnerWidth

ChannelDepthFromHotFace

BoltChannelDiameter

BoltChannelCenterX

BoltChannelCenterY

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>ChannelStart = 20.0</td>
<td>39</td>
</tr>
<tr>
<td>16</td>
<td>ChannelPitch = 15.0</td>
<td>59</td>
</tr>
<tr>
<td>17</td>
<td>ChannelDepthFromHotFace = 22.0</td>
<td>69</td>
</tr>
<tr>
<td></td>
<td># Type 4 inputs</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>ChannelOuterWidth = 5.0</td>
<td>41</td>
</tr>
<tr>
<td>42</td>
<td>ChannelInnerWidth = 2.0</td>
<td>43</td>
</tr>
<tr>
<td>44</td>
<td>ChannelDepth = 15.0</td>
<td>44</td>
</tr>
</tbody>
</table>

Type 2~4 Channel Dimensions

Square Slot Roots

ChannelOuterWidth

ChannelInnerWidth = ChannelOuterWidth

Round Slot Roots

ChannelOuterWidth

ChannelInnerWidth = 0

General Radiused Slots

ChannelOuterWidth

0 < ChannelInnerWidth < ChannelOuterWidth

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>41</td>
<td># Type 2-4 inputs</td>
<td></td>
</tr>
<tr>
<td>42</td>
<td>ChannelOuterWidth = 5.0</td>
<td>42</td>
</tr>
<tr>
<td>43</td>
<td>ChannelInnerWidth = 2.0</td>
<td>43</td>
</tr>
<tr>
<td></td>
<td># Type 2 input</td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>DeepChannelOuterWidth = 5.0</td>
<td>46</td>
</tr>
<tr>
<td>47</td>
<td>DeepChannelInnerWidth = 2.0</td>
<td>47</td>
</tr>
<tr>
<td></td>
<td># Type 3 input</td>
<td></td>
</tr>
<tr>
<td>55</td>
<td>SlantChannelOuterWidth = 5.0</td>
<td>55</td>
</tr>
<tr>
<td>56</td>
<td>SlantChannelInnerWidth = 2.0</td>
<td>56</td>
</tr>
</tbody>
</table>
Odd vs Even Number of Slots

If there is an odd number of channels and the symmetry line cuts the last channel in half, make HalfChannel = True to model the (last) half channel on the symmetry line.

Input data: echo to screen?

If EchoInput is True, then the script will print out the relevant input for the mold.
Execution: Input data into script

Open the Offset Python File in a text editor. (Notepad++ is free and highly recommended)

Insert the mold geometry by editing the values corresponding to the dimensions shown in the schematics.

Execution: enter data for all molds

Change the above (red data and blue choices), to match your mold geometry.
Execution: mold type specific data

With Type specific parameters, only the values corresponding to the desired type will be considered (other values can be ignored).

E.g. If the mold type was type 2, the only sections that the script will read are the Type 2-4 inputs, and the Type 2 input.

Execution: Run Abaqus

Open Abaqus, and under the File menu click Run Script (left). Then find and load the OffsetScript and click OK (above).
Execution: Output results for CON1D calibration

If all the input values are valid, the script will automatically make a mesh and solve with ABAQUS. If not it will state which dimension is invalid.

When complete, Abaqus will present the final mesh and print out the desired values, including the thermocouple offset value.

Typical mesh = ~200,000 tetrahedrons (for 3% heatflux error target on mesh adaptivity)

```
Average Hotface Temperature = 190.8364
Average Channel Root Temperature = 61.5405
Thermocouple Temperature = 101.9632
```

Type the following into CON1D variables [user length units]

- Cooling water channel depth = 21.9644
- Cooling water channel width = 4.2493
- Channel distance (center to center) = 16.3407
- Mold thickness = 48.8563
- Calibrated TC depth = 18.4856

i.e. Offset the thermocouple 3.5114mm closer to the hot face

Execution: 3D detailed results

Because the OffsetScript creates the mold and analyses the effects of the heat loads, after running the script the user can load the .odb file for further analysis.
Validation Cases

Temperature Values (°C):

<table>
<thead>
<tr>
<th></th>
<th>Abaqus</th>
<th>Calibrated CON1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotface</td>
<td>190.84</td>
<td>191.42</td>
</tr>
<tr>
<td>Coldface</td>
<td>61.55</td>
<td>62.09</td>
</tr>
<tr>
<td>Thermocouple</td>
<td>101.96</td>
<td>102.49</td>
</tr>
</tbody>
</table>

M. Langeneckert MS Thesis Fig 3.16

<table>
<thead>
<tr>
<th></th>
<th>Abaqus</th>
<th>Calibrated CON1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotface</td>
<td>319.63</td>
<td>319.19*</td>
</tr>
<tr>
<td>Coldface</td>
<td>86.78</td>
<td>86.36*</td>
</tr>
<tr>
<td>Thermocouple</td>
<td>161.85</td>
<td>161.37*</td>
</tr>
</tbody>
</table>

M. Langeneckert MS Thesis Fig 3.1

<table>
<thead>
<tr>
<th></th>
<th>Abaqus</th>
<th>Calibrated CON1D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hotface</td>
<td>238.77</td>
<td>238.07</td>
</tr>
<tr>
<td>Coldface</td>
<td>90.06</td>
<td>89.28</td>
</tr>
<tr>
<td>Thermocouple</td>
<td>143.15</td>
<td>142.40</td>
</tr>
</tbody>
</table>

L. Hibbeler MS Thesis Fig 3.9

*With applied convection coefficient on TC in script of 60 kW/m²K

Conclusion

• We have developed a method to calibrate CON1D to have the accuracy of a 3D FEA model

• This is implemented in a user friendly Python Script that will soon be available
Acknowledgements

• Continuous Casting Consortium Members
  (ABB, ArcelorMittal, Baosteel, Tata Steel, Goodrich,
  Magnesita Refractories, Nucor Steel, Nippon Steel,
  Postech/ Posco, SSAB, ANSYS-Fluent)

• Ron O’Malley, Junya Iwasaki, Melody Langeneckert

• Dassault Systemes (ABADUS parent company)

• More information:
  L.C. Hibbeler, M.M. Langeneckert, J. Iwasaki, I. Hwang, R.J. O’Malley, and B.G. Thomas,