



**Metal Process Simulation Laboratory
Department of Mechanical Science and Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801**



CON1D Users Manual Version 10.10.01

Brian G. Thomas

Continuous Casting Consortium

Report

Oct 01, 2010

CON1D User's Manual

Version 10.10.01

Brian G. Thomas

University of Illinois at Urbana-Champaign
Department of Mechanical Science and Engineering
1206 West Green Street,
Urbana, IL 61801

Acknowledgments

The CON1D program has been developed by many different graduate students and researchers working since 1987 in the Metals Process Simulation Laboratory at the University of Illinois under the direction of Professor Brian G. Thomas. Major code developers include Brian G. Thomas, Bryant Ho, Guowei Li, David Stone, and Ya Meng. Other contributors include Nick Youssef, Avijit Moitra, Ying Shang, Huan Li and Xiaoxu Zhou. This project has been sponsored by the Continuous Casting Consortium at UIUC, and the National Science Foundation (Grants MSS-8957195 and DMI9800274). Ongoing model calibration and validation has been made possible by experimental data and measurements on operating casters, which have been provided by several different steel companies, notably including LTV (ISG Cleveland, Inc) and Armco (AK Steel).

Oct 01, 2010.

Part I Introduction

Welcome to CON1D! CON1D is a Fortran program which models heat transfer and solidification in the mold region of a continuous caster. The model simulates one-dimensional transient heat transfer-solidification in the steel shell coupled with 2-D steady-state heat conduction in the mold. Hence, the model is most easily applied to regions away from the corners of the cross-section. It is intended for the study of steel slab casters, but can also be applied to other processes.

The heat flux extracted from the solidified shell surface can either be supplied as a specified function of distance below the meniscus, or can be calculated using the interfacial model included in the program. The superheat can be treated in three different ways in the program: 1) calculating temperature in the liquid steel; 2) supplying a superheat flux profile as a stepwise linear function of distance below the meniscus; or 3) letting the program calculate the heat flux added to shell surface, based on previous 3-D turbulent flow calculations. The program can simulate wide/narrow face, outer/inner face of molds (with or without curvature). It is also capable of calculating heat transfer as the strand passes by each roll in the spray zones beneath the mold.

A large quantity of information can be obtained using CON1D, which runs in only a few seconds on a personal computer. The output results include the following variables (as a function of distance below the meniscus):

- (1) Temperatures: mold hot face, cold face, shell surface and shell interior, cooling water
- (2) Shell thickness (including positions of liquidus, solidus, and shell isotherms);
- (3) Heat flux leaving the shell (across the interfacial mold / shell gap);
- (4) Ideal mold taper (based on 1D shrinkage calculations);
- (5) Thickness and velocity of solid and liquid flux layers in the mold/shell interfacial gap;

In addition, the model derives important constants: (solidus temperature, liquidus temperature, mean heat flux in the mold, heat balance at mold exit, maximum temperature in the water channel, negative strip time, oscillation mark pitch, etc.).

The model warns when there may be boiling in the water channels, if the mold cold-face temperature exceeds the water boiling point at that pressure. It also indicates when there may be excessive mold friction, if the lubricating liquid flux layer completely solidifies before mold exit.

Finally, the model outputs the temperatures of mold thermocouples, whose positions are input by the user. This allows easy calibration of the model with known data, by rerunning CON1D until the predicted mold temperatures at these positions match the measured ones.

The model features user-friendly input of the following comprehensive list of variables affecting mold heat transfer, (reasonable defaults are provided for many of these):

complete mold geometry including

- water slots and bolts
- mold curvature
- mold platings: variable thickness distribution and conductivity down mold
- scale buildup on water channels

mold flux properties and adjustable parameters

- temperature-dependent viscosity and conductivity
- variable slag rim thickness down mold
- powder consumption rate
- solid flux velocity

temperature and composition-dependent properties of steel, copper, and water

- thermal conductivity
- specific heat
- density
- steel phase (austenite or ferrite), latent heat, solidus and liquidus temperatures
- water inlet temperature, velocity, and pressure

oscillation mark size, frequency and stroke

- effect on heat flow
- effect on mold flux consumption

effect of fluid flow on superheat delivery to the shell

- enhanced thermal conductivity in the liquid
- specified superheat flux to the solidifying interface
- model predicted superheat flux (slab casters)

spray zone variables

- heat transfer coefficient Model
- roll space, roll radius, contact angle
- nozzle spray width and length, water flow rate

Part II How to run CON1D

(1) How to run CON1D on IBM-PC

(after copying all contents of floppy disk to a single folder on your PC hard drive)

- a) Copy sample input data file to 'xxxx.inp' (where xxxx is any 4-character identifier you want). Edit this file to change the data as desired to match your conditions of interest (See Part III)
- b) Run con1d by typing "con1d" (from a DOS window) or double clicking on the con1d.exe file
- c) Answer the questions asked interactively
- d) Examine output data contained in following output files:

- xxxx.ech (echo file containing interactive input data)
- xxxx.ext (conditions at mold exit)
- xxxx.frc (phase fraction of shell surface and under certain depth below surface)
- xxxx.fxt (flux temperature distribution)
- xxxx.gpt (gap heat transfer coefficients and film thicknesses)
- xxxx.gpv (consumption check)
- xxxx.mld (summary of mold temperatures and heat fluxes)
- xxxx.prf (temperature distribution and heat balance at mold exit)
- xxxx.pf2 (temperature distribution and heat balance at user interested point)
- xxxx.prp (steel properties)
- xxxx.shl (shell temperatures and taper histories)
- xxxx.shr (gap shear stress distribution)
- xxxx.spr (heat transfer coefficients in secondary cooling zones)
- xxxx.sst (steel shell temperature below surface)
- xxxx.tpr (taper history)

- xxxx.liq (liquid concentration at surface)
- xxxx.sol (solid concentration at shell surface)
- xxxx.lqi (liquid concentration at some distance under shell surface)
- xxxx.sli (solid concentration at some distance under shell surface)
- xxxx.seg (solidification time, CR, SDAS, Tsol from shell surface to inside)

- e) Extract data from these files and create graphics using your favorite Personal Computer or Macintosh graphics programs (eg. Kaleidograph or Excel) OR:

Make plots in a window on your monitor by running the program wgnuplot and executing the 3 macro-routines provided by typing the following at the prompt in wgnuplot:

Call 'g.mld' 'xxxx' 'yyyy' (compare mold temperature, shell temperature, and heat flux results from two runs xxxx.mld and yyyy.mld)

Call 'g.shl' 'xxxx' 'yyyy' (shell thickness results)

Call 'g.gpt' 'xxxx' 'yyyy' (interfacial flux thickness results)

After viewing each plot, type carriage return for next plot.

(2) How to run CON1D on workstation

- a) Transfer files in sub directory src.unix to workstation
(Warning: check last line to make sure the file transferred correctly)
- b) Type "make" to compile and get the executable file 'con1d' on workstation
- c) Edit input data file and run CON1D as in (1) a)-d)
- d) View data using the 3 gnuplot macros: plotg.mld, plotg.shl or plotg.gpt by typing at the unix prompt: plotg.mld xxxx yyyy
plotg.shl xxxx yyyy or plotg.gpt xxxx yyyy
Use ^C to get next plot. Alternatively, copy results files to another computer and postprocess with Kaleidograph or Excel etc..

Compiling con1d:

Con1d10.10.01 is compiled with double precision on all variables, which avoids errors in some situations.

(3) How to calibrate CON1D:

- a) Verify that program works OK by running example input file
- b) Copy example input file to a new file xxxx.inp (where xxxx is any 4-character identifier) and edit to match the caster and conditions of interest
- c) Run the model until the following match:
 1. Mean heat flux in the mold should match measured mold heat flux (based on performing a heat balance on the cooling water) within about 3%.
(The heat balance calculates the average heat extraction rate (kW/m^2) from the top of the mold to the specified distance, usually mold exit)
Total power extracted per wide face (kW) = $q_{\text{tot}} (\text{kW/m}^2) * z_{\text{mold}} * W$
 2. Mold water ΔT for individual water cooling channels (slots) should match just as well as mean heat flux.
Note that the total water channel cross section area must be input to obtain the predicted temperature increase compared with measured data.

Change input parameters which are uncertain, and rerun until a match is obtained. Typical variables to change include: mold flux properties, (conductivity, emissivity) air gap thickness profile down mold, and speed ratio of the solid flux.

- d) After getting mold water / heat balance to agree, compare predicted and measured mold temperatures at thermocouple locations. Model temperatures should be slightly low, if the thermocouples are positioned close to either the water or to hot bolts (where a slot is missing). This geometric dependency should be checked with results from a FEA analysis of the mold in order to quantify the offset distance needed to match the model predictions with results at the thermocouple location for a known heat flux.

If model temperatures are still low, there may be scale build-up on the mold cold face. Add a scale layer and rerun the model to obtain a match. (The heat flux should not change much)
If model temperatures are very high, either a) the thermocouples may have a contact resistance and not be measuring the actual mold temperature, or b) there is incipient boiling increasing the effective heat transfer coefficient.

- e) Shell thickness profile down mold should match that obtained from breakout shells to the extent that the input conditions match the breakout conditions. Time-dependent casting speed and precise determination of the time – distance relationship for the breakout shell is needed when making this comparison.
- f) Run the calibrated model for other input conditions, as desired.

3a) CON1D Subprograms

con1d	main program initialize variables, handle input and output files, control simulation
ckseg	use Clyne-Kurz simple analysis segregation Model to calculate liquidus and solidus (optional)
bound	function to determine heat flux leaving shell (incorporates model of mold/shell interface) (calls slagthick and check etc. subroutines for calculation)
fluxtemp	calculate temperature distribution in mold flux vs. time & distance
hbal	perform approximate heat balance on shell at specified distance (usually mold exit) output temperature distribution and heat balance compare heat loss from shell surface with heat content in shell: 1) superheat or heat flux to shell inside 2) latent heat 3) sensible heat
input	input and "echo" write xxxx.ech file (using SI units) (uses functions doutrf and dinnrf to calculate curved mold thickness)
ma	2-D mold temperature calculation (uses function fyy in calculating y direction)
moldtemp	function to calculate mold hot face temperatures (based on 1-D, 2-D, or data points)
once	define heat flux to inside of shell (based on user-specified data points)
onceff	define default superheat flux to inside of shell (based on previous 3-D turbulent flow results)
phase1	determine fraction of α , γ , δ and liquid based on phase diagram (uses function ae3c to determine Ae3 temperature with carbon)
t1e, rk, cp	unctions to define temperature dependent properties (TLE, k, Cp) based on stainless steel or pure iron (props1.f)
para, fracti	composition-dependent property calculation (phase2.f) define temperature, carbon- and phase- dependent properties for low-carbon steels (para calculates coefficients for phase diagram change lines fracti defines phase fractions for TLE, Cp, k)
print	interpolate between nodal temperatures to locate solidus and liquidus; calculate ideal taper from simple shell shrinkage (using subroutine tpr) output results at specified time interval
tapering	calculate taper according to Dipenaar's method
shear	calculate shear stress distribution vs.time
simul	calculate temperatures (shell) at next time step
usertle, userrk, usercp, usertemp	define TLE, k, Cp, liquidus, solidus by user
waterh	calculate heat transfer coefficients between water and mold cold face

These *.for files are compiled into *.obj files and linked using any FORTRAN compiler (eg. Microsoft 8087) on IBM-PC or standard FORTRAN-77 on UNIX workstations.

(3b) List of Unix files contained in CON1D

```

con1d.f:    program con1d
bound.f:    real function bound(ts,zmm,zmoldmm,tmold,hwater,ispay,inaro,vctmms,
+          tcold1,ibound,iz,twater,dzcm,tsol,freqz,qcold,thotc,zpp,ratio)
          subroutine rcalc(rhoflux,rindex,wcao,wsio2,wmgo,wna2o,
+          wk2o,wfeo,wfe2o3,wnio,wmno,wcr2o3,wal2o3,wtio2)
          subroutine slagthick(mu0,expn,rhoflux,eslag,esteel, rindex,acoeff,ts,
+          tcrystal,tmold,tkliquid,tksolid,qosc,qcons,vsolid,
+          vctms,dliquid,dsolid,deff,rcontact)
          subroutine balanc (a,n,np)
          subroutine zrhqr (a,m,rtr,rti)
          subroutine hqr(a,n,np,wr,wi)
          subroutine check zm,dosc,vctms,t,vsolid,dsolid,dliquid,rhoflux,
+          expn,mu0,vosc,freqz)
          function liqthick(tmpdliq,mu0,expn,rhoflux,vctms,qcons,qosc)
          function oscstuck(pitch,dliquid,dsolid,tkeff,zm,ioscflag)
ckseg.f:    subroutine ckseg(tckliq,tcksol,tckperi,SDAS,CR)
          subroutine segoutput(time,temp,iFileHandle1,iFileHandle2, zmm1,zmm2)
fluxtemp.f: subroutine fluxtemp_time(dzcm,vctmms,ratio)
          subroutine fluxtemp(zmm,tn)
hbal.f:    subroutine hbal(tnext,qtin,qts,n,iout1)
input.f:    subroutine input(ibatch,nconsf,ibound,i2d,pwater,nflux,nthc)
          real function doutrf(dmld1a,amrad,zm,zmold,dmen,dwater)
          real function dinnrf(dmld1a,amrada,zm,zmold,dmen,dwater)
ma.f:      subroutine ma(z1mm,nz,nthc1,nz0,dmen,xmold)
          function fyy(x,y,zmm,c1,fb1,xmold,zmold)
moldtemp.f: real function moldtemp(qtot,hwater,zm,zmold,tcold,inaro,twater,thotc,
+          dtcoat,dtcupp,number)
once.f:    subroutine once(qfill,max,nflux)
onceff.f:  subroutine onceff(qfill,max)
phase1.f:  subroutine phase1(tliq,tsol,tperi,carbeq,pc,ae1,ae3)
          function ae3c(carbon)
phase2.f:  subroutine para
          subroutine fracti(tc,pc)
print.f:   subroutine print(n,t,printflag)
          subroutine tpr(tsurf,def,n,t)
props1.f:  function tle(tc,pc)
          real function rk(tc,dkdt)
          real function cp(tc)
shear.f:   subroutine shearstress(vctmms,ratio)
simul.f:   subroutine simul (qfill,t,tnext,q,n,maxq)
tapering.f subroutine tapering(t,tnext,shrink,zmm,dzcm,zmold,tsol,tliq,pc,n)
user.f:    subroutine userrk(tc,dkdt,rk)
          subroutine usercp (tc,cp)
          subroutine usertemp (tliq,tsol,tperi)
          subroutine usertle (tc,pc,tle)
waterh.f:  subroutine waterh(zm,zpp,hwater,tcold,twater,tkmold)

```

Part III Theory

III-1.1 Heat Conduction in Solidifying Steel Shell

Temperature in the solidifying steel shell is governed by the 1D transient heat conduction equation :

$$\rho_{steel} C_{p,steel}^* \frac{\partial T}{\partial t} = k_{steel} \frac{\partial^2 T}{\partial x^2} + \frac{\partial k_{steel}}{\partial T} \left(\frac{\partial T}{\partial x} \right)^2 \quad [3.1]$$

where C_p^* , the effective specific heat for the solidifying steel, is defined as:

$$C_p^* = \frac{dH}{dT} = C_p - L_f \frac{df_s}{dT} \quad [3.2]$$

The simulation domain, a slice through the liquid steel and solid shell, together with the boundary conditions, is presented in Figure 3.1.

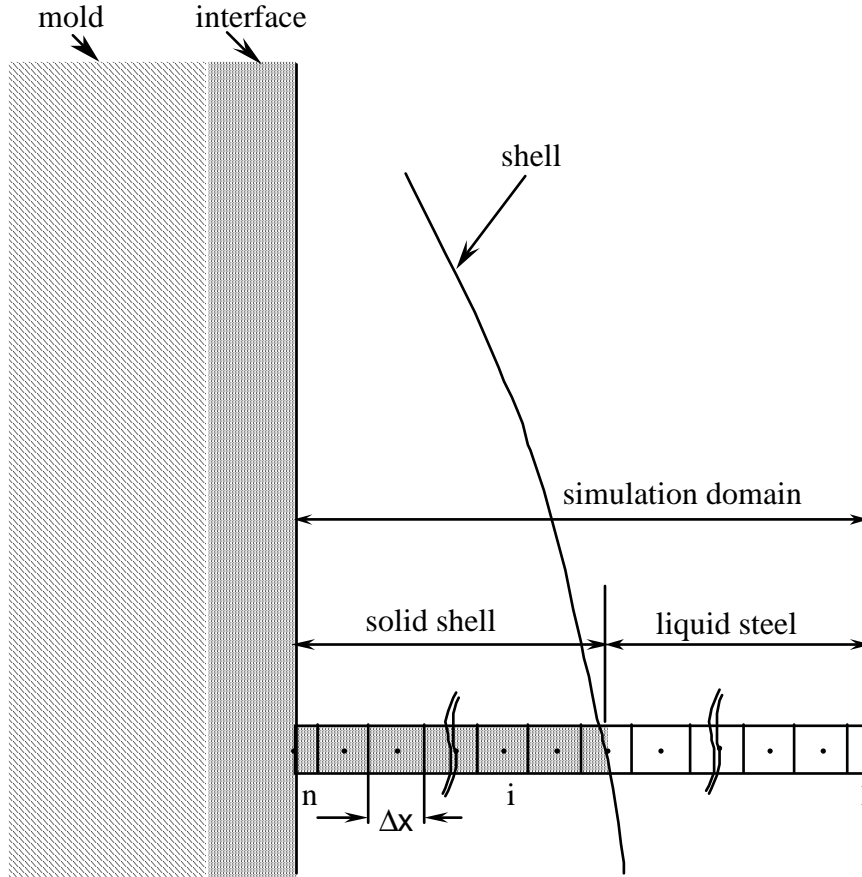


Fig. 3.1 The simulation domain in steel

Equation [3.1] is solved at each time step using the following explicit finite-difference discretization (a central difference scheme):

(1) Interior nodes:

$$T_i^{new} = T_i + \frac{\Delta t \cdot k}{\Delta x^2 \rho C p^*} (T_{i-1} - 2T_i + T_{i+1}) + \frac{\Delta t}{4\Delta x^2 \rho C p^*} \frac{\partial k}{\partial T} (T_{i+1} - T_{i-1})^2 \quad [3.4]$$

(2) Liquid boundary node (adiabatic boundary condition):

$$T_1^{new} = T_1 + \frac{2\Delta t \cdot k}{\Delta x^2 \rho C p^*} (T_2 - T_1) \quad [3.5a]$$

(3) Shell surface node (with heat flux boundary condition):

$$T_n^{new} = T_n + \frac{2\Delta t \cdot k}{\Delta x^2 \rho C p^*} (T_{n-1} - T_n) + \frac{\Delta t}{\rho C p^*} \frac{\partial k}{\partial T} \left(\frac{q_{int}}{k} \right)^2 - \frac{2\Delta t \cdot q_{int}}{\Delta x \rho C p^*} \quad [3.5b]$$

The shell surface temperature is denoted as T_s :

$$T_s = T_n^{new}$$

where $q_{int} = h_{gap} (T_s - T_{hot})$, and T_{hot} is the mold hot face temperature, which can be obtained together with q_{int} and h_{gap} through iteration. The details are presented in section III.

(4) Steel thermal properties (T_{liq} , T_{sol} , ρ , C_p , k and L_f):

T_{liq} and T_{sol} , are calculated by the program as a function of steel composition, based on the phase diagram for low-alloy steel. The properties, ρ , C_p , k and L_f can be treated in three ways. First, the carbon content and temperature dependent properties can be calculated based on the phase diagram for low carbon steel by the program (select option $icp = 1000$ in input data file, see Part IV). Alternatively, temperature dependent properties can be found from a set of empirical formula for stainless steel. Finally, ρ , C_p , k and L_f can be input as constants.

III-1.2 Superheat Flux

Before it solidifies, the steel must first cool from its initial pour temperature to the liquidus temperature. Due to turbulent convection in the liquid pool, the “superheat” contained in this liquid is not distributed uniformly. A small database of results from a 3-D fluid flow model³ is used to determine the heat flux delivered to the solid / liquid interface due to the superheat dissipation, as a function of distance below the meniscus, q_{sh} . Examples of this function are included in Fig. 3.2, which represents results for a typical bifurcated, downward-directing nozzle. The initial condition on the liquid steel at the meniscus is then simply the liquidus temperature.

This superheat function incorporates the variation in superheat flux according to the superheat temperature difference, ΔT_s , casting speed, V_c , and nozzle configuration. The influence of this function is insignificant to shell growth on most of the wide face, where superheat flux is small and contact with the mold is good.

For the first node, where temperature is lower than liquidus temperature, the effect of superheat on node temperature is:

$$T_i = T_i + \frac{\Delta t}{\rho C_p^* dx} q_{sh} \quad [3.6]$$

where $dx = \Delta x$ for interior nodes, and $dx = \Delta x/2$ for boundary nodes.

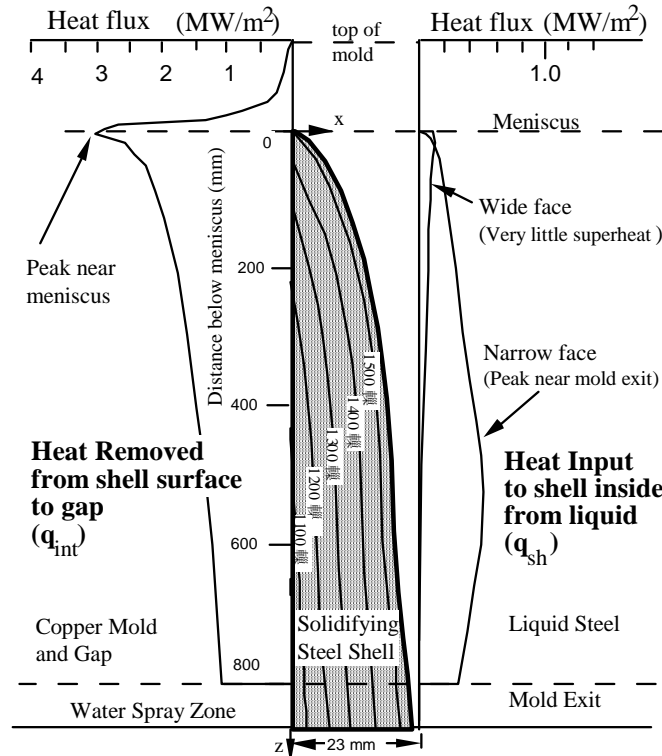


Fig. 3.2. Model of solidifying steel shell showing typical isotherms and heat flux conditions

III-1.3 Slag Rim

The slag rim and the location of the peak plays an important role in calculating the temperatures of the hot and cold face of the mold, as well as the behavior of the slag and its phases in the simulation. The slag rim affects the heat flux peak, and can be used in the tuning the CON1D model.

An extra slag rim (in addition to the solidified slag layer) may extend above the heat flux peak. As shown in Fig. 3.3, its dimensions can be specified in the input file by piece-wise linear interpolation between the slag rim thickness at the top-surface meniscus level, (D_{RIM1mm}) and the thickness at the point of maximum heat flux (D_{RIM2mm}) that is located z_{peak} mm below the top surface.

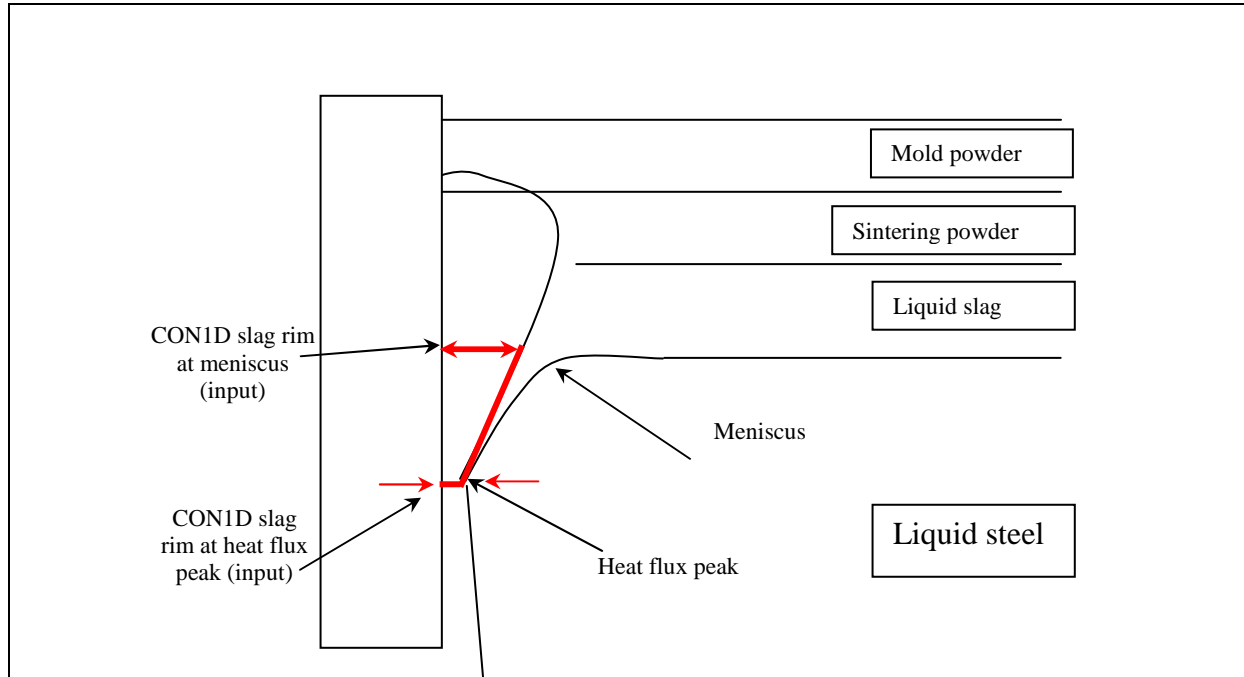


Fig 3.3 Schematic representation of CON1D slag rim

The liquid pool exerts a tiny pressure in the flux channel, which depends on the thickness of the liquid flux layer on the top surface, H_{RIMmm} (mm).

III-2 Heat Conduction in the Mold

Temperatures within the mold, including in particular the hot and cold face temperatures, are calculated knowing the heat fluxes, q_{cold} and q_{int} , and the effective heat transfer coefficient to the water, h_{water} .

Two dimensional, steady state temperature distribution within a rectangular section through the mold is calculated in the upper portion of the mold. It is found that the temperature is nearly linear in the mold thickness direction about 50 mm or so below the meniscus. Thus, a 1D assumption is adopted in the model below the distance z_{ana} below the meniscus (Fig. 3.4A).

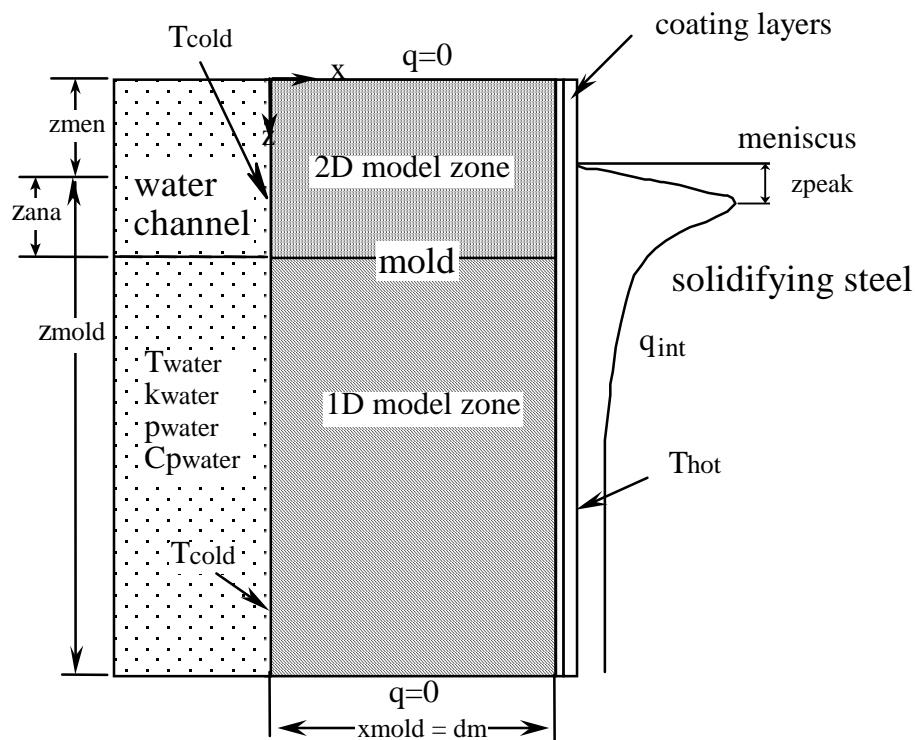


Fig. 3.4A Mold Temperature Calculation

III-2.1 Effective Heat Transfer Coefficient at Mold Cold Face:

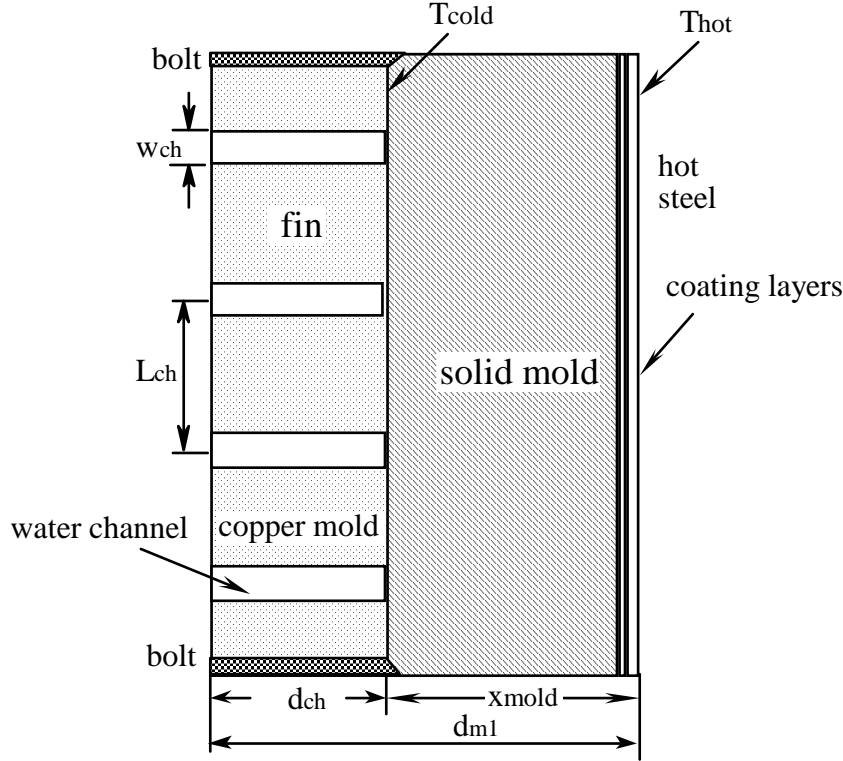


Fig. 3.4B Water Channel in the mold

The effective heat transfer coefficient between the cooling water and the cold face (“water-side”) of the mold copper, h_{water} , is calculated using the following formula, which includes a possible resistance due to scale build-up:

$$h_{water} = 1 / \left(\frac{d_{scale}}{k_{scale}} + \frac{1}{h_{fin}} \right) \quad [3.7]$$

To account for the complex nature of heat flow in the undiscretized width direction, the heat transfer coefficient between the mold cold face and cooling water, h_{fin} , is obtained using the following formula which treats the sides of the water channels as heat-transfer fins.

$$h_{fin} = \frac{h_w w_{ch}}{L_{ch}} + \frac{\sqrt{2h_w k_{mold} (L_{ch} - w_{ch})}}{L_{ch}} \tanh \sqrt{\frac{2h_w d_{ch}^2}{k_{mold} (L_{ch} - w_{ch})}} \quad [3.8]$$

Where, L_{ch} , w_{ch} , d_{m1} , d_{ch} are geometry parameters shown in Fig. 3.4B and k_m is the mold (copper) thermal conductivity. The presence of the water slots can either enhance or diminish the heat transfer relative to a mold with constant thickness (equal to the minimum distance between the root of the water

channel and the hot face). Deep, closely-spaced slots augment the heat transfer coefficient, (h_{fin} larger than h_w) while shallow, widely-spaced slots inhibit heat transfer. In most molds, h_{fin} and h_w are very close.

In Eq. 3.8, the heat transfer coefficient between the water and the sides of the water channels, h_w , is calculated assuming turbulent flow through a pipe: [C. A. Sleicher and M. W. Rouse, Int. J. Heat Mass Transf. V. 18, pp. 677-683, 1975]

$$h_w = \frac{k_{water}}{D} \left(5 + 0.015 \text{Re}_{waterf}^{c_1} \text{Pr}_{waterw}^{c_2} \right) \quad [3.9]$$

Here D is the equivalent diameter of the water channel, c_1 and c_2 are the empirical constants.

$$D = \frac{2w_{ch}d_{ch}}{w_{ch} + d_{ch}} \quad [3.10]$$

$$c_1 = 0.88 - 0.24 / (4 + \text{Pr}_{waterw}) \quad [3.11]$$

$$c_2 = 0.333 + 0.5e^{-0.6\text{Pr}_{waterw}} \quad [3.12]$$

$$\text{Re}_f = \frac{\rho_{water} v_{water} D}{\mu_{waterf}} \quad [3.13]$$

$$\text{Pr}_w = \frac{\mu_{waterw} C p_{water}}{k_{waterw}} \quad [3.14]$$

The properties of the water, needed in the above equations, can be treated as either constants or temperature dependent variables evaluated at the film temperature (half-way between the water and mold wall temperature), according to the selection in the input data file made by the user.

III-2.2 1D Steady State Temperature Model of Mold:

Hot face temperatures at or near the mold surface are calculated from the thickness of the copper, d_m , the water side heat transfer coefficient, and the interfacial heat flux, explained in a later section.

$$T_{hotc} = T_{water} + q_{int} \left(\frac{1}{h_{water}} + \frac{d_m}{k_m} \right) \quad [3.15a]$$

Further hot face temperatures are calculated by incorporating the resistances of the various thin coating layers, (Ni, Cr, air gap etc.), which vary with distance down the mold according to the input file:

$$T_{hot} = T_{water} + q_{int} \left(\frac{1}{h_{water}} + \frac{d_m}{k_m} + \frac{d_{ni}}{k_{ni}} + \frac{d_{cr}}{k_{cr}} + \frac{d_{poly}}{k_{poly}} \right) \quad [3.15b]$$

$$T_{mprime} = T_{water} + q_{int} \left(\frac{1}{h_{water}} + \frac{d_m}{k_m} + \frac{d_{ni}}{k_{ni}} + \frac{d_{cr}}{k_{cr}} + \frac{d_{poly}}{k_{poly}} + \frac{d_{air}}{k_{air}} \right) \quad [3.15c]$$

The hot face temperatures include the surface of the copper, T_{hotc} , the surface of the outermost mold plating layer, T_{hot} and the temperature at the interface between the air gap and the solid mold flux layer, T_{mprime} , including any contact resistance that might be present. The output file includes several temperatures:

$$T_{hot} = (q + h_{mold1} * T_{water}) / h_{mold1}$$

$$T_{hotc} = T_{hot} - dT_{coat}$$

$$T_{cold} = T_{hotc} - q * d_{mold} / tk_{mold}$$

In these equations, the copper thickness, d_m , varies with distance down the mold, according to the mold curvature:

Outer radius:

$$d_{mold}^{outer} = d_{moldo}^{outer} + \sqrt{R_o^2 - \frac{1}{4}(Z_{mold_total})^2} - \sqrt{R_o^2 - \frac{1}{4}(Z_{mold_total})^2 - Z_{mold_total}^2} \quad [3.16]$$

Inner radius

$$d_{mold}^{inner} = d_{moldo}^{inner} - \sqrt{R_I^2 - \frac{1}{4}(Z_{mold_total})^2} + \sqrt{R_I^2 - \frac{1}{4}(Z_{mold_total})^2 - Z_{mold_total}^2} \quad [3.17]$$

where d_{moldo} is the mold thickness at the top of the mold, Z_{mold_total} is the total mold length (sum of working mold length Z_{mold} and distance of meniscus from top of the mold Z_{men}) and R_O , R_I are mold outer and inner radius of curvature respectively.

T_{cold} is the temperature of the root of the water channel, at the interface between the mold copper and the scale layer, if present:

$$T_{cold} = T_{water} + \frac{q_{int}}{h_{water}} \quad [3.18]$$

III-2.3 2D Steady-State Temperature Model of Mold:

By assuming constant thermal conductivity in the upper mold and constant heat transfer coefficient between the mold cold face and the water channel along the casting direction, the two dimensional steady state heat conduction equation for mold temperature modification in meniscus region is the following Laplace equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial z^2} = 0 \quad [3.19]$$

The analytical solution to this equation with the boundary conditions shown in Fig. 3.2 is a cosine series:

$$T(x, z) = T_{water} + c_0 \left(\frac{k_{mold}}{h_{water}} + x \right) + \sum_{n=1}^{\infty} \left(c_{2n} \cos(\lambda z) (c_1 e^{-\lambda x} + e^{-\lambda x}) \right) \quad \lambda = \frac{n\pi}{Z_{2D}} \quad \lambda = \frac{n\pi}{Z_{2D}} \quad [3.20]$$

where x is distance through the thickness of the mold, measured from the root of the water slot. The constants λ , c_0 , c_1 and c_{2n} depend on the heat flux calculated to enter the mold hot face q_{int} , thermal conductivity k_m , the effective heat transfer coefficient h_{fin} and the mold geometry.

$$\lambda = \frac{n\pi}{Z_{2D}} \quad [3.21]$$

$$c_0 = \frac{2 \cdot \Delta z}{Z_{2D}} \left(a_j \frac{z_{j+1} + z_j}{2} + b_j \right) \quad [3.22]$$

$$c_1 = \frac{k_{mold} \lambda - h_{water}}{k_{mold} \lambda + h_{water}} \quad [3.23]$$

$$c_{2n} = B \frac{\frac{2}{Z_{2D}} \left(\frac{1}{\lambda} \right)^2}{e^{\lambda x} (1 - c_1 e^{-2\lambda x})} \quad [3.24]$$

$$B = \sum_j \left(\left(a_j z_{j+1} + b_j \right) \sin(\lambda z_{j+1}) - \left(a_j z_j + b_j \right) \sin(\lambda z_j) \right) + \frac{a_j}{\lambda} \left(\cos(\lambda z_{j+1}) - \cos(\lambda z_j) \right)$$

where a_j and b_j are the linear interpolation coefficients of the interface heat flux in zone j :

from $q_{aj} = k_m (a_j z_{aj} + b_j)$ and $q_{bj} = k_m (a_j z_{bj} + b_j)$, the a_j and b_j can be obtained:

$$\begin{aligned}
a_j &= \frac{q_{\text{int } j+1} - q_{\text{int } j}}{k_{\text{mold}} (z_{j+1} - z_j)} \\
b_j &= \frac{q_{\text{int } j} z_{j+1} - q_{\text{int } j+1} z_j}{k_{\text{mold}} (z_{j+1} - z_j)}
\end{aligned}
\tag{3.25}$$

The actual hot face temperature of the mold is adjusted to account for the possible presence of mold coatings and air gaps:

$$T_{\text{mold}} = T(d_{\text{mold}}, z) + q_{\text{int}} \left(\frac{d_{ni}}{k_{ni}} + \frac{d_{cr}}{k_{cr}} + \frac{d_{poly}}{k_{poly}} + \frac{d_{air}}{k_{air}} \right)
\tag{3.26}$$

III-3.1 Heat Flux Across the Interfacial Gap

Heat flux extraction from the steel is governed primarily by heat conduction across the interfacial gap, whose thermal resistance is determined by the thermal properties and thicknesses of the solid and liquid powder layers, in addition to the contact resistances at the flux / shell and flux / mold interfaces and powder porosity, which are incorporated together into a single equivalent air gap, d_{air} . Non-uniformities in the flatness of the shell surface, represented by the oscillation marks, have an important effect on the local thermal resistance, and are incorporated into the model through the depth and width of the oscillation marks. This is used to calculate an effective average depth of the marks relative to heat flow, d_{eff} . The oscillation marks can be filled with either flux or air, depending on the local shell temperature. When the gap is large, significant heat is transferred by radiation across the semi-transparent flux layer. This model for gap heat conduction is illustrated in Fig. 3.5 and 3.6 and given by the following equation:

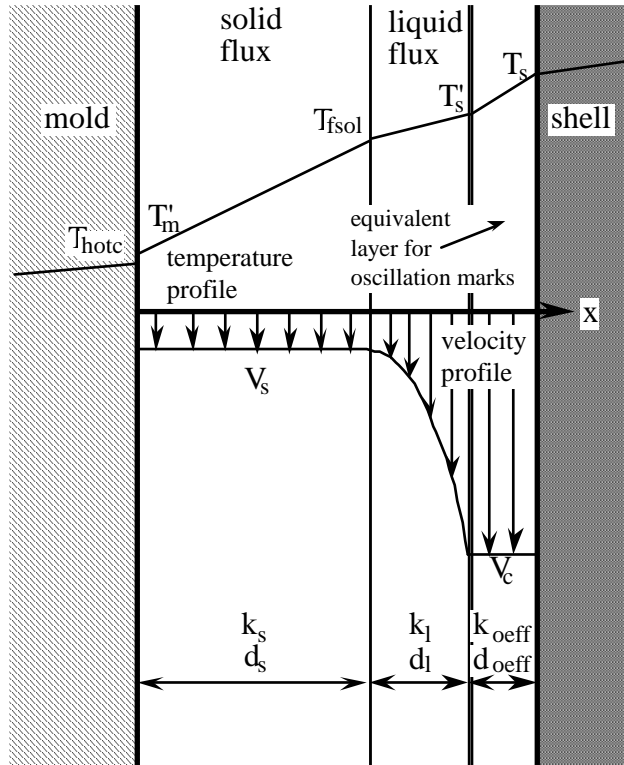


Fig. 3.5 Velocity and temperature profiles assumed across interfacial gap

$$q_{\text{int}} = h_{\text{gap}} (T_s - T_{\text{mold}}) \quad [3.27]$$

$$h_{\text{gap}} = \frac{1}{\left(r_{\text{contact}} + \frac{d_{\text{air}}}{k_{\text{air}}} + \frac{d_{\text{solid}}}{k_{\text{solid}}} \right) + \left(\frac{1}{\frac{d_{\text{liquid}}}{k_{\text{liquid}}} + \frac{d_{\text{eff}}}{k_{\text{eff}}}} + h_{\text{rad}} \right)} \quad [3.28]$$

$$h_{\text{rad}} = \begin{cases} \frac{m^2 \sigma (T_s^2 + T_{\text{mold}}^2)(T_s + T_{\text{mold}})}{0.75a(d_{\text{liquid}} + d_{\text{eff}}) + \frac{1}{e_{\text{mold}}} + \frac{1}{e_{\text{steel}}} - 1} & (T_{\text{mprime}} \geq T_{\text{crystal}}) \\ \frac{m^2 \sigma (T_s^2 + T_{\text{crystal}}^2)(T_s + T_{\text{crystal}})}{0.75a(d_{\text{liquid}} + d_{\text{eff}}) + \frac{1}{e_{\text{slag}}} + \frac{1}{e_{\text{steel}}} - 1} & (T_{\text{crystal}} < T_{\text{mprime}}) \end{cases}$$

where

q_{int} = heat flux transferred across gap (Wm^{-2})

h_{gap} = effective heat transfer coefficient across the gap ($\text{Wm}^{-2}\text{K}^{-1}$)

h_{rad} = radiation effective h ($\text{Wm}^{-2}\text{K}^{-1}$)

T_s = surface temperature of the steel shell ($^{\circ}\text{C}$)

T_{mprime} = mold temperature+mold/slag contact resistance delta T ($^{\circ}\text{C}$)

T_{mold} = surface temperature of the mold (outermost coating layer) ($^{\circ}\text{C}$)

T_{crystal} = mold flux crystallization temperature ($^{\circ}\text{C}$)

r_{contact} = flux/mold contact resistance($\text{m}^2\text{K}/\text{W}$)

d_{air} , d_{solid} , d_{liquid} , d_{eff} = thickness of the air gap, solid, liquid flux, and oscillation mark layers (mm)

k_{air} , k_{solid} , k_{liquid} , k_{eff} = conductivity of the air gap, solid, liquid flux, and oscillation mark layers (W/mK)

m = flux refractive index

σ = Stefan Boltzman constant ($\text{Wm}^{-2}\text{K}^{-4}$)

a = flux absorption coefficient (m^{-1})

ϵ_s , ϵ_m = steel, mold surface emmisitivities

The calculation of some of the above variables is explained in the next section. Other variables are input data, defined in the Nomenclature section.

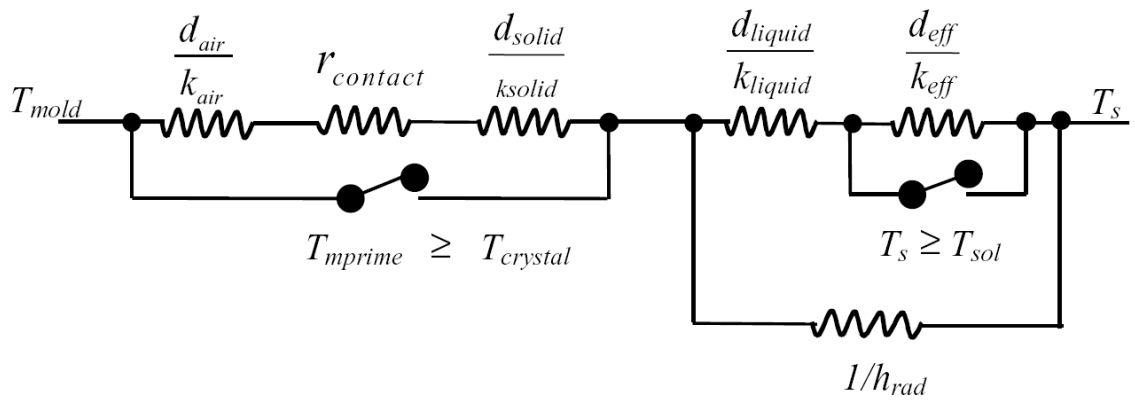


Fig. 3.6 Thermal resistances used in the interface model

III-3.2 Mass and Momentum Balances on the flux within the interfacial gap

Flux is assumed to flow down the gap as two distinct layers: solid and liquid. The solid layer is assumed to move at a constant velocity, V_s , which is always greater than zero and less than the casting speed according to the input factor, f_v .

$$V_s = f_v \cdot V_c \quad (0 < f < 1) \quad [3.29]$$

The casting speed is imposed at the point of contact between the shell and the liquid layer, which is assumed to flow in a laminar manner, owing to its high viscosity. Flow in the liquid layer is given by the Navier-Stokes equation:

$$\frac{\partial \tau_{xz}}{\partial x} = (\rho_{steel} - \rho_{slag}) g \quad [3.30]$$

where x = in the direction across the gap

ρ_{slag} = average density of flux

g = gravity acceleration

This formulation ignores axial pressure gradients in the flux channel, which may be generated in the central regions of the wide face and could be accounted for by replacing $-\rho_{flux}$ with $(\rho_{Fe} - \rho_{flux})$ in Eq. 3.30. The effect was found to be negligible, however, except in the vicinity at the meniscus, where oscillation plays an even more important role.

The tangential shear stress, τ_{xz} , is related to the viscosity of the molten flux, μ , by:

$$\tau_{xz} = \mu \frac{\partial V_z}{\partial x} \quad [3.31]$$

The viscosity, μ , is assumed to vary exponentially with distance across the gap, according to the temperature:

$$\mu = \mu_o \left(\frac{T_o - T_{fsol}}{T - T_{fsol}} \right)^n \quad [3.32]$$

where T_{fsol} is the solidification temperature of the flux, μ_o is the viscosity measured at T_{1300} , and n is an empirical constant chosen to fit all of the measured data. Thus the viscosity evaluated at surface temperature T_s is:

$$\mu_s = \mu_o \left(\frac{T_{1300} - T_{fsol}}{T - T_{fsol}} \right)^n \quad [3.33]$$

Mass balance was imposed to express the fact that the known powder consumption must equal the total powder flow rate past every location down the interfacial gap. The following equation expresses this condition as a balance on the total consumption, Q_f (kg m^{-2}) as the density was assumed to be constant:

$$\frac{Q_f \times V_c}{\rho_{\text{flux}}} = V_{\text{solid}} d_{\text{solid}} + \bar{V}_{\text{liquid}} d_{\text{liquid}} + V_c d_{\text{osc}} \quad [3.34]$$

where the average depth of the oscillation marks (relative to their volume to carry flux), d_{osc} (Fig. 3.7), is calculated from:

$$d_{\text{osc}} = \frac{0.5 L_{\text{mark}} d_{\text{mark}}}{L_{\text{pitch}}} \quad [3.35]$$

$$L_{\text{pitch}} = \frac{V_c}{\text{freq}} \quad [3.36]$$

Equations [3.29], [3.30], [3.31] and [3.33] yield a velocity distribution across the flux layers, which is illustrated in Fig. 3.5. Integrating across the liquid region yields an average velocity for the liquid layer, V_l :

$$V_l = \frac{(\rho_{\text{flux}} - \rho_{\text{steel}}) g d_{\text{liquid}}^2}{\mu_s (n+2)^2 (n+3)} + \frac{V_c + V_s (n+1)}{(n+2)} \quad [3.37]$$

where V_l = avg. velocity of liquid flux (m s^{-1})
 V_c = casting speed (m s^{-1})
 V_s = velocity of solid flux (m s^{-1})
 n = flux viscosity exponent
 ρ_{Fe} = steel density (kg m^{-3})
 ρ_{flux} = flux density (kg m^{-3})
 g = gravity (9.81 m s^{-2})
 d_l = thickness of liquid flux layer (m)
 μ = flux viscosity (Pa-s)
 T_s' = steel outside surface temp. ($^{\circ}\text{C}$)

Equation [3.34] and [3.37] were solved simultaneously for d_s and d_l . Special cases arise when $d_s = 0$ (zone I) or $Q_f < V_c d_{\text{osc}}$ (zone III). In the former case, (zone I), the solid rim thickness, (via z_{rim1} and z_{rim2}) must be input, as it is unaffected by the flux consumption rate. In the latter case, (zone III), the oscillation mark volume is replaced with air.

The average effective thickness of the oscillation marks, relative to its effect on heat transfer, is calculated from:

$$d_{\text{eff}} = \frac{0.5 L_{\text{mark}} d_{\text{mark}}}{(L_{\text{pitch}} - L_{\text{mark}}) \left(1 + 0.5 \frac{d_{\text{mark}}}{d_{\text{liquid}} + d_{\text{solid}}} \frac{k_{\text{gap}}}{k_{\text{mark}}} \right) + L_{\text{mark}}} \quad [3.38]$$

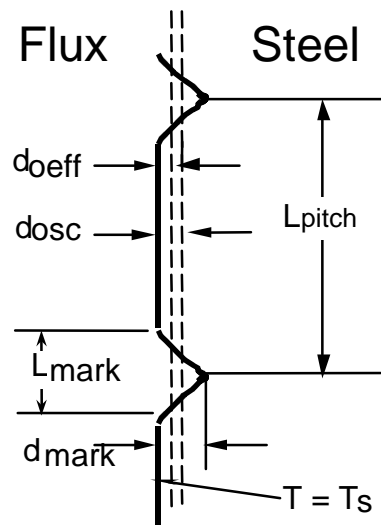


Fig. 3.7 Model treatment of oscillation marks

III-4 Mold cooling water temperature rise:

A heat balance in the mold calculates the total heat extracted, q_{tin} , based on the increments of heat flux found in the interfacial heat flow calculation, q_{int} (W/m^2):

$$q_{tin} (kJ/m^2) = \sum_{mold} q_{int} * \Delta t \quad [3.39]$$

The mean heat flux in the mold is calculated at the specified distance below the meniscus (usually at mold exit)

$$q_{ttot} (kW/m^2) = \frac{q_{tin} V_c}{z} \quad [3.40]$$

The temperature rise of the cooling water is determined by:

$$\Delta T_{cooling\ water} = \sum_{mold} \frac{q_{int} V_c \Delta t L_{ch}}{\rho_{water} C_{pwater} V_{water} (m/s) w_{ch} d_{ch}} \quad [3.41]$$

This relation assumes that the cooling water slots have uniform dimensions, w_{ch} and d_{ch} , and spacing, L_{ch} . Heat entering the hot face (between two water channels) is assumed to pass entirely through the mold to heat the water flowing through the cooling channels. This calculation must be modified to account for missing slots due to bolts or water slots which are beyond the slab width, so do not participate in heat extraction. So the modified cooling water temperature rise is:

$$\Delta T_{modified\ cooling\ water} = \Delta T_{cooling\ water} * \frac{w_{ch} d_{ch}}{L_{ch}} * \frac{slab\ width}{totcharea} \quad [3.42]$$

This cooling water temperature rise prediction is useful for calibration of the model with an operating caster.

III-5 Mold Taper calculation

Previous work involving coupled thermal-stress calculations has investigated the shrinkage profile of the shell down the mold. This calculation decomposed the total strain into the sum of three components: the thermal contraction, the plastic strain, and the elastic strain. The first of these has been found to dominate. Fortuitously, the thermal strain at the surface of the shell has been found to match quite closely with the average total strain across the thickness of the shell, which controls its shrinkage. This is because this outer layer of steel solidifies first and shrinks relatively stress free, while the later, inner steel to solidify against it is weaker and accommodating. Thus, the entire, complex mechanical behavior of the shell can be approximated quite reasonably by the following simple calculation for the shrinkage of the shell, especially if that layer is austenite (eg. %C>0.1%):

$$\text{Old Model (def):} \quad \Delta W = (TLE(T_{sol}) - TLE(T_s)) \frac{W}{2} \quad [3.43]$$

Shrinkage does not start until the shell is fully solid. Deflection on each side of the mold is found by multiplying by half of the domain width (ie wide face width for narrow face taper).

A new model to calculate the shrinkage is based on Chandra's method^[1]. The thermal strain is computed according to the average thermal linear expansion of the solid shell between the two consecutive time step. This method is better for low-carbon steel containing delta-ferrite.

$$\text{New Model (def2):} \quad \Delta W = \left(\frac{1}{i} \right) \sum_{i=\text{solid nodes}} (TLE(T_{\text{next}(i)}) - TLE(T(i))) \frac{W}{2} \quad [3.44]$$

$$\text{cumulative taper (\% per m)} = \frac{2 \Delta W}{Z W} \quad [3.45]$$

$$\text{cumulative taper (\% per mold)} = \frac{2 \Delta W}{W} \quad [3.46]$$

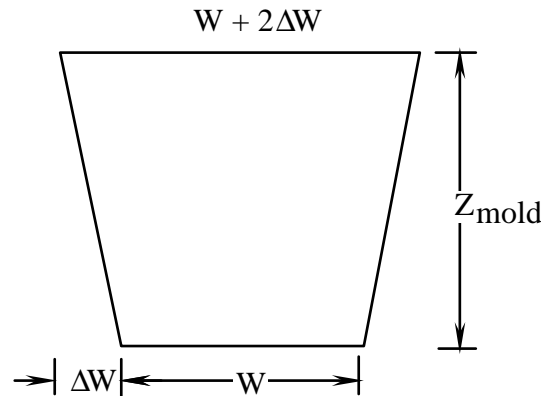


Fig. 3.8 Mold Taper Calculation

Ideal taper at mold exit:

$$\text{ideal taper (\% per m)} = \frac{2 \Delta W}{z_{\text{mold}} W} \quad [3.47]$$

$$\text{ideal taper (\% per mold)} = \frac{2 \Delta W \text{ at mold exit}}{W} \quad [3.48]$$

where:

- ρ = density (kg/m³)
- ρ_0 = density at reference temperature, T_o (kg/m³)
- T_{sol} = steel solidus temperature
- W = strand width between narrow faces (for a wide-face run to find narrow-face taper)
- ΔW = change in steel shell width (mm)
- z_{mold} = working mold length
- Z = $z + z_{\text{men}}$

TLE is the thermal linear expansion function for the given steel grade,

$$\text{defined by TLE (T)} = \left(\frac{\rho_0}{\rho(T)} \right)^{1/3} \quad [3.49]$$

TLE is calculated from weighted averages of the phases present,

$$\text{TLE (T)} = \% \delta \text{ TLE } \delta(T, \%C) + \% \gamma \text{ TLE } \gamma(T, \%C) \quad [3.50]$$

It depends on temperature and carbon content:

Ideal Taper Calculation:

In addition to shell shrinkage (above), ideal **narrow-face** taper in a slab (or funnel) mold should also consider: wideface expansion, narrow face distortion, funnel effect, and flux layer:

$$\Delta W \text{ (mm)} = \Delta W_{\text{shell}} - \Delta W_{\text{NF-D}} - \Delta W_{\text{WF-E}} - \Delta W_{\text{FUNNEL}} - \Delta W_{\text{FLUX}}$$

ΔW_{shell} is the decrease in width of the wideface due to shrinkage of the shell (mm), calculated with CON2D.

$\Delta W_{\text{NF-D}}$ is the narrow face distortion down the mold minus the distortion in the meniscus,

$\Delta W_{\text{WF-E}}$ is the wide face expansion at the meniscus minus the wide face expansion down the mold,

$\Delta W_{\text{WF-D}}$ is the wide face distortion,

ΔW_{FUNNEL} is decrease in wideface perimeter due to the change in mold cavity dimensions down a funnel mold,

ΔW_{FLUX} is the flux layer thickness down the mold minus that at the meniscus.

See details in: Brian G. Thomas and Claudio Ojeda, 2003 ISSTech Steelmaking Conference, Indianapolis, IN, USA, April 27-30, 2003, ISS-AIME, Warrendale, PA, 2003, pp. 295-308.

narrow face distortion for a wide-face run (mldistor)

$$\Delta W_{NF-D} = \frac{3\alpha_{hot} \left(\bar{T}_{hot} - \bar{T}_{cold} \right) (t_{hot} + t_{cold})}{t_{cold}^2 K_1} [Lx - x^2]$$

$$\text{where: } K_1 = 4 + 6 \frac{t_{hot}}{t_{cold}} + 4 \left(\frac{t_{hot}}{t_{cold}} \right)^2 + \frac{E_{hot}}{E_{cold}} \left(\frac{t_{hot}}{t_{cold}} \right)^3 + \frac{E_{cold}}{E_{hot}} \frac{t_{cold}}{t_{hot}}$$

wide face expansion for a wide-face run (mldexpn)

$$\Delta W_{WF-E} (\text{mm}) = \alpha_{mold} \left(\frac{W}{2} \right) \left(T_{ref} - \frac{T_{cold} + T_{hot}}{2} \right)$$

Thermal distortion from the water box is calculated from the equivalent water box thickness. (which is the thickness of steel plate that would have the same rigidity in preventing thermal distortion of the copper mold as the actual water box)

Computing wide-face taper: (narrow face run)

$$\Delta W (\text{mm}) = \Delta W_{shell} - \Delta W_{WF-D} - \Delta W_{NF-E} - \Delta W_{FLUX}$$

Needs ΔW_{WF-D} = wide-face distortion and ΔW_{NF-E} = narrow face expansion

funnel effect (extralength):

$$\Delta W_{FUNNEL} = \left\{ \left[\frac{a^2 + b^2}{2b} \sin^{-1} \left(\frac{2ab}{a^2 + b^2} \right) \right] - a \right\}$$

Billet Mold Distortion:

$$\Delta W_{B-dist} (\text{mm}) = -\alpha_{mold} \left(\frac{W}{2} \right) \left(\frac{T_{cold} + T_{hot}}{2} \right)$$

Variables in Output file xxxx.tpr

Wideface run on slab mold (needs i2d=2, for better taper calculations*)

```

zmm,          distance down mold (mm)
def,          shell shrinkage (mm) (based on shell surface temp)
              in print.f subroutine tpr
def2,         shell shrinkage (mm) (based on average shell temp)
              (Chandra - Dipannar method) in tapering.f
moldexpn,     wide face expansion (mm)
mldistor+moldexpn, total mold distortion (mm) (wf expansion and nf distortion)
mldistor,     narrow face distortion (mm)
extralength,  funnel shape (mm)
idealtprdef,  nf mold shape (deflection) in mm
idealtprinst, instantaneous taper (%/m)
idealtpr      cumulative taper (%/m) based on def2 (as of Aug'07)

```

* The parabolic shape of distorted slab mold (both distortion and expansion) needs temp distribution down the mold, which we only get after iterating until convergence. Then, one more iteration gives us the parabolic shape. With i2d=1, we just use the billet distortion equation (which depends on local temp at that distance down the mold)

III-6 Heat Transfer in the spray cooling zone:

Heat transfer in the spray cooling zones tracks the steel shell as it moves past each individual roll. Heat transfer is governed mainly by the spray heat transfer coefficient, which is calculated from the average spray water flow rate from the nozzle (Q_w [L/m²s]) at each spray zone, using a general equation of the following form:

$$h_{\text{spray}} = A \cdot c \cdot Q_w^n (1 - bT_0) \quad [3.51]$$

Where, T_0 is water and ambient temperature in spray zone. In the model of Nozaki et al.^[2], $A \cdot c = 0.3925$, $n = 0.55$, $b = 0.0075$ and h_{spray} has a unit of kW/ m²K. Users can choose other models^[3] by setting coefficients A , n and b . Heat transfer is enhanced by nucleate film boiling if the steel shell surface temperature drops below 550°C.

Due to the non-uniform distribution of the water flux along the length of the nozzle spray, the heat transfer coefficient beneath the spray water is divided into four length regions and assigned a piece-wise linear profile according to 3 different multiplying factors to the heat transfer coefficient (h_{spray}) input at points B, C, and D (Fig 3.9). The figure below (Fig 3.9) shows a typical heat transfer coefficient curve along the spray length, compared with the original value, h_{spray} (multiplying factor=1).

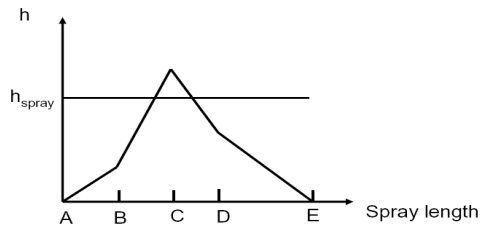


Fig 3.9 Typical heat transfer coefficient curve of the nozzle spraying

The heat transfer coefficient at point A is $h_A = 0$, point B is $h_B = h_1 \cdot h_{\text{spray}}$, point C is $h_C = h_2 \cdot h_{\text{spray}}$, point D is $h_D = h_3 \cdot h_{\text{spray}}$ and point E is $h_E = 0$, where 0, h_1 , h_2 , h_3 and 0 in front of h_{spray} are multiplying factors. The determination of z_1 , z_2 , z_3 , h_1 , h_2 and h_3 should be based on experimental results of nozzle spray characterization (water flux profile in the spray region) or other relevant nozzle spray experiments. Ideally, values of 0, h_1 , h_2 , h_3 , and 0 should produce heat transfer coefficients that match the values obtained by equation 3.51 using the water flux at the specified positions (A, B, C, D, E), after accounting for heat removed by the rolls.

Heat extraction is a maximum directly beneath the spray nozzle (assumed centered between the rolls) and at the roll / shell contact region (Fig. 4.0). The relative size of these maxima is governed by the fraction of heat specified to leave via the rolls relative to that removed by the spray cooling water. The output including the variation in heat transfer coefficient is given in xxxx.spr.

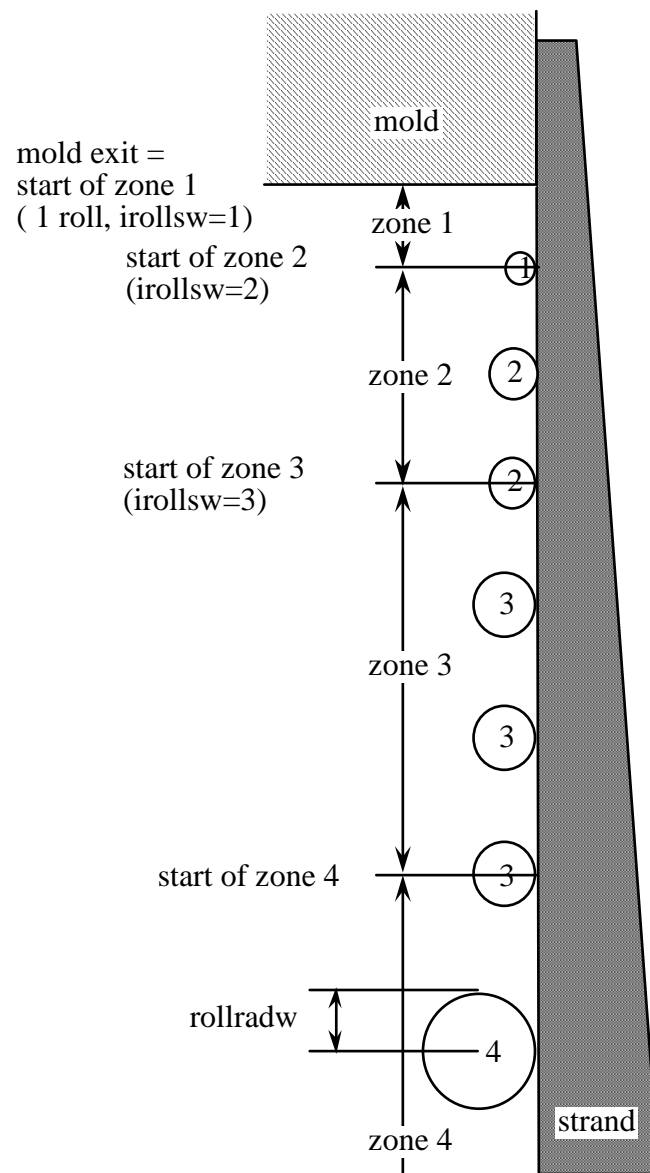


Fig. 4.0 Schematic of a typical spray zone configuration below the mold

Part IV Input data:

The input file should be put in a file 'xxxx.inp'. The data included in this file can be written in free format. The rest of section defines the input variables in the input file. Units are specified in both the input data files and in the Nomenclature section, so are omitted here.

Section (1) Casting conditions:

<i>Variables in program</i>	<i>Comments</i>
<i>nvc</i>	number of time-cast speed data points (if=1, constant speed)
<i>vctime</i>	time
<i>vcmmmin</i>	casting speed It is used (1) to calculate time step (2) to calculate increment in casting direction (3) in calculation of negative casting time (4) to output the heat into the shell
<i>tinit</i>	pour temperature
<i>dmen</i>	distance of meniscus from top of mold, z_{men}
<i>submerg</i>	nozzle submergence depth (meniscus to top of port) it is used to determine the superheat fluxes
<i>zmax</i>	maximum simulation length
<i>ibound</i>	flag to specify treatment of mold/shell interface
=0	to model heat flux across the interfacial gap between mold and shell, using input mold flux properties and parameters. This means that the shell solidification is coupled with mold temperature simulation and flux motion, so iteration is required.
=2	to model heat flux across the interfacial gap between mold and shell for oil casting case, only contact resistance, oscillation mark and air gap are taken into account. In these two case, the input data (<i>nbddata</i> , <i>zbdata</i> , <i>bdata</i>) following are ignored.
=-1	to enter interface heat flux data between the shell and mold as a function of the distance below the meniscus and calculate both the temperature in the mold and solidification of the shell.
=-2	to enter average interface heat flux data between the shell and mold, the distribution

of the heat flux is calculated by the default distribution function with user entered mold flux tuning parameters q_fac , tc_fac , q_n

$=-3$ to enter cooling water temperature increase

$nldata$ number of input heat flux data points (< 20) (see *ibound*)

$zldata$ z data (distance below meniscus) (see *ibound*)

$bdata$ q data (interface heat flux) (see *ibound*)

$qtot0$ $qtot0_l$ average mold heat flux (MW/m²)

q_fac tc_fac q_n mold heat flux tuning parameters used to adjust heat flux shape along the mold

$dtwater0$, $dtwater0_l$ mold cooling water increase for calculating mold heat flux

Section (2) spray zone variables:

<i>Variables in program</i>	<i>Comments</i>
---------------------------------	-----------------

<i>hncov0</i>	minmum convection heat transfer coefficient (natural) (W/m ² K)
---------------	--

Right side of caster:

<i>tsprgwater_r</i>	water and ambient temperature after spray zone (Deg C) We choose heat transfer coefficient function: $h=ACW^n(1-bT)$ Nozaki Model: $A*C=0.3925$, $n=0.55$, $b=0.0075$ Ishiguro Model: $A*C=0.581$, $n=0.451$, $b=0.0075$
<i>sprycoefa_r</i>	coefficient A
<i>sprycoefn_r</i>	coefficient n
<i>sprycoefb_r</i>	coefficient b
<i>nzone</i>	number of spray zones (for narrow face $nzone=1$) (< 30)

The table in the following lines includes the valuables needed to define the heat flux in right side spray zone:

zone start positions, number of rolls in zones, roll radius, water flow rate, spray zone width, spray zone length, roll contact angle, fraction of heat flux through roll. i.e. the variables:
izone_r, dzone_r, irollsw_r, rollradw_r, wflow_r, sprywidth_r, sprydist_r, rollcontact_r, fracrol_r, sprycoefc_r, convect_r, tambi_r

<i>izone</i>	index of spray cooling zones below the mold
<i>dzone_r</i>	spray zone start points below meniscus
<i>irollsw_r</i>	number of rolls in the particular spray zone
<i>rollradw_r</i>	roll radius
<i>wflow_r</i>	total flow rate of all nozzles between each pair of rolls in this zone
<i>sprywidth_r</i>	spray zone width
<i>sprydist_r</i>	spry zone length
<i>rollcontact_r</i>	roll contact angle
<i>fracrol_r</i>	fraction of heat flux through the roll
<i>sprycoefc_r</i>	coefficient C in heat transfer coefficient
<i>convect_r</i>	convection coefficient
<i>tambi_r</i>	ambient temperature for each zone
<i>sprycoefz_r</i>	$z1$ equals to AB/AE , $z2$ equals to AC/AE and $z3$ equals to AD/AE (Fig 3.9)

<i>sprycoefh_r</i>	magnifying factors of h_{spray}
	The end of last spray zone
<i>dzone_r</i>	end of last spray zone (mm)
<i>flagLF_r</i>	consider Leidenfrost effect? (-1=Yes, 0=No)
<i>npLF_r</i>	number of points in the Leidenfrost effect curve (The following two lines give Leidenfrost effect h-multipliers and temperatures)

Left side of caster:

<i>tsprgwater_l</i>	water and ambient temperature after spray zone (Deg C) We choose heat transfer coefficient function: $h=ACW^n(1-bT)$ Nozaki Model: $A*C=0.3925$, $n=0.55$, $b=0.0075$ Ishiguro Model: $A*C=0.581$, $n=0.451$, $b=0.0075$
<i>sprycoefa_l</i>	coefficient A
<i>sprycoefn_l</i>	coefficient n
<i>sprycoefb_l</i>	coefficient b
<i>nzone</i>	number of spray zones (for narrow face $nzone=1$) (< 30)

The table in the following lines includes the valuables needed to define the heat flux in left side spray zone:

zone start positions, number of rolls in zones, roll radius, water flow rate, spray zone width, spray zone length, roll contact angle, fraction of heat flux through roll. i.e. the variables:
izone_l, *dzone_l*, *irollsw_l*, *rollradw_l*, *wflow_l*, *sprywidth_l*, *sprydist_l*, *rollcontact_l*, *fracrol_l*, *sprycoefc_l*, *convect_l*, *tambi_l*

<i>dzone_l</i>	spray zone start points below meniscus
<i>irollsw_l</i>	number of rolls in the particular spray zone
<i>rollradw_l</i>	roll radius
<i>wflow_l</i>	total flow rate of all nozzles between each pair of rolls in this zone
<i>sprywidth_l</i>	spray zone width
<i>sprydist_l</i>	spry zone length
<i>rollcontact_l</i>	roll contact angle
<i>fracrol_l</i>	fraction of heat flux through the roll
<i>sprycoefc_l</i>	coefficient C in heat transfer coefficient
<i>convect_l</i>	convection coefficient
<i>tambi_l</i>	ambient temperature for each zone
<i>sprycoefz_l</i>	$z1$ equals to AB/AE , $z2$ equals to AC/AE and $z3$ equals to AD/AE (Fig 3.9)
<i>sprycoefh_l</i>	magnifying factors of h_{spray}

	The end of last spray zone
<i>dzone_l</i>	end of last spray zone (mm)
<i>flagLF_l</i>	consider Leidenfrost effect? (-1=Yes, 0=No)
<i>npLF_l</i>	number of points in the Leidenfrost effect curve (The following two lines give Leidenfrost effect h-multipliers and temperatures)

Section (3) mold cooling water properties:

<i>Variables in program</i>	<i>Comments</i>
<i>twater</i>	cooling water temperature mold top (Deg C) It is used to calculate cooling water heat transfer coefficient
<i>pwater</i>	cooling water pressure (Mpa) used only to take adjust water boiling point for warning)
<i>nvwater</i>	form of cooling water velocity / flowrate (1=m/s, 2=L/s) (L/s refers to volumetric flow rate through each channel)
<i>rvwaterWF</i>	wide face cooling water velocity / flowrate
<i>rvwaterWNF</i>	narrow face cooling water velocity / flowrate

Section(4) slab geometry

<i>xslbmm</i>	slab thickness (mm)
<i>yslbmm</i>	slab width (mm)
<i>zmoldtmm</i>	total mold length (mm)
<i>dmoldl</i>	WF mold thickness with water channel (mm, outer radius, top)
<i>dmolda</i>	WF mold thickness with water channel (mm, inner radius, top)

Section (5) steel properties:

Variables *Comments*
in program

The compositions of the steel are used to find
steel liquidus and solidus temperatures and phase fractions for TLE.

<i>pc pmn ps pp psi</i>	%C %Mn %S %P %Si
<i>pcr pni pcu pmo pti</i>	%Cr %Ni %Cu %Mo %Ti
<i>pal pv pn pnb pw</i>	%Al %V %N %Nb %W
<i>pco</i>	%Co

icp flag of steel grade (1000: carbon steels,
304, 316, 317, 347, 410, 419, 420, 430: AISI stainless steels,
999: user subroutine) Determines which temperature-
dependent properties will be used as default.

isegflag CK simple Seg. Model flag (1=yes, 0=no)
CR Cooling rate used in Seg.Model(if above =1) (K/sec)

In the following eight lines, the program will use the constant property input, or
will default to the appropriate grade and temperature-dependent function if = -1.

<i>slig</i>	steel liquidus temperature (if =-1, then based on grade - see icp)
<i>ssoli</i>	steel solidus temperature (if =-1, then based on grade - see icp)
<i>sdense</i>	steel density (if =-1, then based on grade - see icp)
<i>stlheat</i>	heat fusion of steel (if =-1, then based on grade - see icp)
<i>semmis</i>	steel emissivity (if =-1, then based on grade - see icp)
<i>scp</i>	steel specific heat (if =-1, then based on grade - see icp)
<i>scon</i>	steel thermal conductivity (if =-1, then based on grade - see icp)
<i>stk</i>	steel thermal expansion coef. (if =-1, then based on grade - see icp) (otherwise, use for taper calculation to replace phase-fraction TLE functions)

Section (6) Simulation parameters:

<i>Variables in program</i>	<i>Comments</i>
<i>version</i>	version of CON1D to run (1=offline version, 2=online version)
<i>iprocess</i>	new simulation or restart (0=new, 1=restart)
<i>zhbal</i>	z position for outputting heat balance information in the mold (This value should be smaller than working mold length.)
<i>inaro</i>	flag to specify wide face/narrow face simulation
=0	for wide face, meaning a mold (1) with a curvature mold hot face (2) including water channels (3) with spray zones containing rolls below mold (4) using slab thickness in taper calculation (5) using wide face default superheat flux
=1	for narrow face, meaning a mold (1) with straight hot face (2) without water channels (3) with only one spray zone(no rolls) below the mold (4) using slab width in taper calculation (5) using narrow face default superheat flux
<i>imoldflq1</i>	type of the mold
=0	slab
=1	funnel mold
=2	billet (oil lubrication)
=-1	shell only
<i>imoldflq2</i>	flag to specify the types of moldface Affects thickness variation in vertical direction for molds with curvature
=1	curved face
=2	straight wide face or narrow face (no curvature)
<i>fflux</i>	flag for treatment of superheat
=0	for calculating temperature in liquid steel to get the heat flux added to shell surface In this case the superheat flux input data are inactive

=1	to use default superheat flux data It means that the program will create superheat flux data according to superheat (pour temperature - liquidus temperature), nozzle submergence, and the simulation face (wide or narrow). In this case the superheat flux input data is ignored.
=-1	to enter superheat flux data
<i>nflux</i>	number of input superheat flux data points (< 20) (see <i>fflux</i>)
<i>zqdata</i>	z data (distance down mold below meniscus) (see <i>fflux</i>)
<i>qdata</i>	q data (superheat flux) (see <i>fflux</i>) (z=0 is at meniscus)
<i>i2d</i>	flag to choose more accurate calculation in meniscus region or not
=0	for 1D calculation of mold heat transfer, using approximate method meniscus region
=1	for more accurate 2D mold heat transfer simulation in meniscus region
=2	for more accurate 2D mold heat transfer simulation for whole mold (one extra loop for better taper calculation)
<i>relax</i>	Relaxation factor for 2-D calculation (default input=1)
<i>zana0</i>	Maximum distance below meniscus for 2D simulation in the mold
<i>dtime</i>	time step size

The program adopts explicit solution scheme, so a large time increment will cause solution to be more unstable: *dtime* should satisfy the criterion: $dtime < 0.5 * \Delta x * \Delta x * dense * Cp / k$
dtime depends on distance increment Δx (see *nstep*). The smaller the distance increment, the smaller *dtime* should be.
The time step must also be bigger than a lower limit which depends on the memory limit of the machine, or an error message appears.

<i>nstep</i>	number of slab sections (< 300) It is used to determine the increment of distance ($\Delta x = thmax / nstep$, x node = 1 is at the center of slab, x node = <i>nstep</i> +1 is at shell surface)
<i>freqz</i>	printout interval

<i>zpmin</i>	the z-coordinate (casting direction) of starting output
<i>thmax</i>	maximum simulation thickness (0 = default . The program will use half of slab thickness $xslab / 2$ for wide face simulation; and half of slab width $yslab / 2$ for narrow face simulation.
<i>itmax</i>	maximum number of time steps
<i>nshlt</i>	Shell thermocouple numbers below surface (less than 10)
<i>fsolid</i>	fraction solid for solid temperature location It is employed to calculate shell thickness: $txshell=(1.-fsolid)*tliq+fsolid*tsol$

Section (7) mold flux properties:

Variables
in program

The following lines are the compositions of mold flux

<i>wcao</i>	<i>wsio2</i>	<i>wmgo</i>	<i>wna2o</i>	<i>wk2o</i>	%CaO	%SiO2	%MgO	%Na2O	%K2O
<i>wfeo</i>	<i>wfe2o3</i>	<i>wnio</i>	<i>wmno</i>	<i>wcr2o3</i>	%FeO	%Fe2O3	%NiO	%MnO	%Cr2O3
<i>wal2o3</i>	<i>wtio2</i>	<i>wb2o3</i>	<i>wlio2</i>	<i>wsro</i>	%Al2O3	%TiO2	%B2O3	%Li2O	%SrO
<i>wzro2</i>	<i>wf</i>	<i>wfreec</i>	<i>wtotc</i>	<i>wco2</i>	%ZrO2	%F	%free C	%total C	%CO2

<i>ncryst</i>	number of Tfsol and viscosity exponent data points
<i>zcrys</i>	z data (distance down mold below meniscus) (see <i>ncryst</i>)
<i>tcrystal</i>	mold flux solidification temperature (°C) (see <i>ncryst</i>)
	It is used at two places in the program:
	(1) to calculate flux viscosity (corresponding to melting point temperature)
	(2) to calculate the thicknesses of liquid flux and solid flux layers
<i>expn</i>	exponent for temperature dependency of viscosity (see <i>ncryst</i>)
<i>tksolid</i>	solid flux conductivity
<i>nkl</i>	number of distance-Liquid flux conductivity data points
<i>zkl</i>	z data (distance down mold below meniscus) (see <i>nkl</i>)
<i>tkl</i>	liquid flux conductivity (see <i>nkl</i>)
<i>mu</i>	flux viscosity at 1300 °C
	It is the referent viscosity in flux viscosity calculation
<i>rhoflux</i>	mold flux or slag density (different from powder density)
	It is used in continuity equation of mold/shell interface model
<i>acoeff</i>	flux absorption coefficient
<i>rindex</i>	flux index of refraction
<i>eslag</i>	slag emissivity
<i>nconsf</i>	form of unit for mold powder consumption rate(see <i>cons</i>)
	(1 = kg of flux / m ²)
	(2 = kg of flux / tonne of steel, in this case the program will convert the unit kg/tonne to kg/m ² according to the thickness and width of slab, and steel density
<i>cons</i>	mold powder consumption rate, Q _f (kg m ⁻²)

<i>Variables in program</i>	<i>Comments</i>
<i>zpeak</i>	location of peak heat flux (relative to meniscus) It is used to define the size of the flux rim" size" (the solid flux layer attached to mold) between the meniscus and peak heat flux locations. If <i>zpeak</i> =0, the program assumes no rim, i.e. the peak heat flux is at the meniscus and the input data are inactive
<i>drim1mm</i>	slag rim thickness at metal level (meniscus) (read in mm)
<i>drim2mm</i>	slag rim thickness at heat flux peak (read in mm)
<i>hrimmm</i>	Liquid pool depth (read in mm)
<i>fluxstrenght</i>	Solid flux tensile fracture strength (read in KPa)
<i>fluxstrenghtc</i>	Solid flux compress fracture strength (read in KPa)
<i>poisson</i>	Solid flux Poisson ratio(-)
<i>nfrc</i>	number of distance-Static friction coeff
<i>zfrcs</i>	z data (distance down mold below meniscus) (see <i>nfrc</i>)
<i>dfricoeff</i>	Static friction coeff data (see <i>nfrc</i>)
<i>fricoeffm</i>	Moving friction coeff data (see <i>nfrc</i>)

Section (8) interface heat transfer variables:

<i>Variables in program</i>	<i>Comments</i>
<i>nratio</i>	number of distance-ratio data points (1=constant ratio of solid flux velocity to casting speed)
<i>zratio</i>	distance from meniscus (nratio # of data points)
<i>vratio</i>	ratio of average solid flux velocity to casting speed (nratio # of data points)
<i>rrcontact</i>	flux/mold contact resistance
<i>ecopper</i>	mold surface emissivity
<i>tkair</i>	air conductivity The program only considers an air gap to exist in oscillation mark root in the region far below meniscus.
<i>ioscflag</i>	flag to decide which oscillation marks model is used (0=average, 1=transient)
<i>dmark</i>	average oscillation mark depth (see Fig 4.1) It is used to define both volume of liquid flux removed in the marks and average oscillation mark thickness for interface heat transfer resistance. Marks are filled with liquid flux/solid flux/air.
<i>width</i>	width of oscillation mark (see Fig 4.1)
<i>freq1</i>	oscillation frequency (-1 = take default cpm=2*ipm casting speed)
<i>stroke</i>	oscillation stroke length (=2.* amplitude)

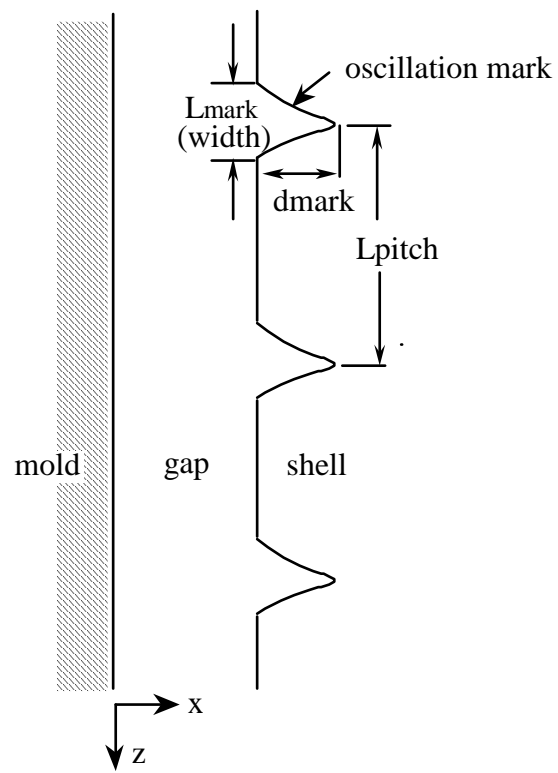


Fig 4.1 Oscillation marks

Section (9) mold water properties

<i>Variables</i> <i>in program</i>	<i>Comments</i>
---------------------------------------	-----------------

The water properties in the following four lines are used to get the heat transfer coefficient in the cooling water channel. If the input data is -1 for these valuables, the program will use default water properties which vary as a function of temperature.

<i>hwinput</i>	heat transfer coefficient (-1 = default = $f(T)$, based on Sleicher and Rouse Eqn)
<i>cpwater</i>	water heat capacity (-1 = default = $f(T)$)
<i>rhewater</i>	water density (-1= default = $f(T)$)

Section (10) mold geometry:

Variables in program	Comments
-------------------------	----------

$dmoldNF$	Narrow face (NF) mold thickness with water channel (mm)
$dwtrEquiv$	Equivalent thickness of water box (mm)
$tNFdiff$	Mean temperature diff between hot & cold face of NF (C)

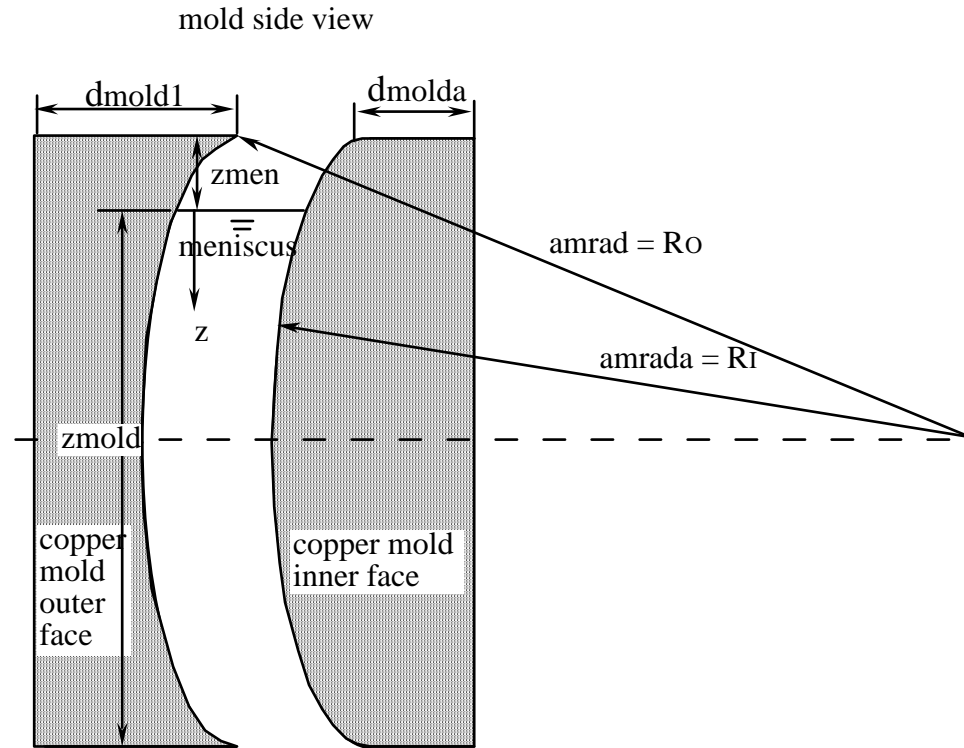


Fig 4.2 mold outer and inner face

<i>Variables in program</i>	<i>Comments</i>
---------------------------------	-----------------

<i>chdepthWF chdepthNF</i>	
----------------------------	--

	cooling water channel depth (WF,NF)
--	-------------------------------------

<i>chwidthWF chwidthNF</i>	
----------------------------	--

	cooling water channel width (WF,NF)
--	-------------------------------------

<i>dchannelWF, dchannelNF</i>	
-------------------------------	--

	distance between cooling water channels (center to center) (WF,NF)
--	--

	(for single water channel, set <i>dchannel=chwidth</i> , to make <i>heff=hwater</i>)
--	---

<i>totchareaWF totchareaNF</i>	
--------------------------------	--

	total channel cross sectional area (WF, NF)
--	---

	(served by water flow line where temp rise measured)
--	--

<i>tkmoldWF tkmoldNF</i>	
--------------------------	--

	mold thermal conductivity (WF,NF)
--	-----------------------------------

<i>alphamold</i>	Mold thermal expansion coeff. (1/K)
------------------	-------------------------------------

<i>funnelD</i>	funnel height (mm)
----------------	--------------------

<i>funnela</i>	funnel width (mm)
----------------	-------------------

<i>funneltop_b</i>	funnel depth at mold top (mm)
--------------------	-------------------------------

<i>amrada</i>	machine inner radius, R_I
---------------	-----------------------------

<i>amrad</i>	machine outer radius, R_O
--------------	-----------------------------

<i>offsetmold_flag</i>	mold offset flag (default=-1, user enter=1)
------------------------	---

<i>offsetmold1 offsetmold2</i>	
--------------------------------	--

	user entered wide face and narrow face mold offset distance (mm)
--	--

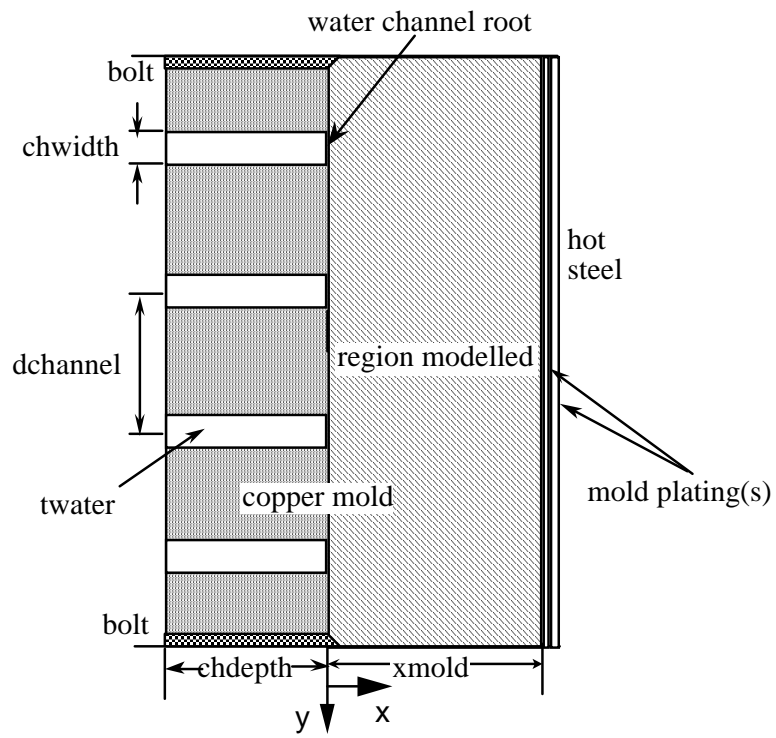


Fig 4.3 Water channel region

<i>Variables in program</i>	<i>Comments</i>
<i>npp</i>	number of mold coating/plating thickness changes below meniscus (< 20) The table in following lines defines the coating / plating (Ni, Cr, other polynite, and water side scale) thicknesses at defined distances below meniscus, and their conductivities. (Each coating is a piece-wise linear function of distance below meniscus found by connecting the points together) i.e. the variables: <i>ip, dscale, dni, dcr, dpoly, zpp, dairg, akscale, tkni, tkcr, tkpoly, tkairg</i>
<i>ip</i>	Index of data points for the interpolation of coating layer thicknesses (must be consecutive: 1,2,3,...)
<i>dscale</i>	The thickness of water scale at mold cold face
<i>dni,</i>	The thickness of Ni coating layer on mold hot face
<i>dcr</i>	The thickness of Cr coating layer on mold hot face
<i>dpoly</i>	The thickness of polynite coating layer on mold hot face
<i>dairg</i>	The thickness of air gap
<i>zpp</i>	The distance from meniscus
<i>akscale</i>	The conductivity of water scale at mold cold face
<i>tkni</i>	The conductivity of Ni coating layer on mold hot face
<i>tkcr</i>	The conductivity of Cr coating layer on mold hot face
<i>tkpoly</i>	The conductivity of polynite coating layer on mold hot face
<i>tkair</i>	The conductivity of air

The line numbers that follow assume 3 coating thickness change, although this is not required.

Section (11) mold thermocouples:

<i>Variables in program</i>	<i>Comments</i>
<i>xoffset</i>	offset distance towards hot face (mm)
<i>nthc</i>	the total number of thermocouples (< 100)

The table in following line to the end defines the thermocouple positions
(for comparing model results with available measurements).

i.e. the variables: *nnthc*, *xthc*, *zthc*

<i>nnthc</i>	Index of thermocouples
<i>xthc</i>	distance beneath mold hot surface
<i>zthc</i>	distance below meniscus

Part V. Nomenclature

Variable	Line # in input file	Symbol	Standard Value	Unit
<i>acoeff</i>	98	a	200	m ⁻¹
<i>cpcte</i>	63	C _p	690	J kg ⁻¹ °K ⁻¹
<i>cpwater</i>	130	C _{pwater}	4179	J kg ⁻¹
<i>dairg</i>	153-162	d _{air}	0	mm
<i>chdepth</i>	141	d _{ch}	25	mm
<i>xmold</i>		d _m		
<i>dmoldl</i>	134	d _{mO}	56.8	mm
<i>dmolda</i>	135	d _{mI}	46.8	mm
<i>dmark</i>	121	d _{mark}	0.4	mm
<i>dni, dcr, dpoly</i>	153-162	d _{ni, dcr, dpoly}	0,0,0	mm
<i>tkni, tkcr, tkpoly</i>	163	k _{ni, kcr, kpoly}	80, 72, 4.1	W m ⁻¹ °K ⁻¹
<i>dscale</i>	153-162	d _{scale}	0	mm
<i>vratio</i>	111	f _c	0.1	-
<i>fsolid</i>	48	f _s	0.7	-
<i>freq</i>	123	freq	1.417	cycles per second
<i>kcte</i>	64	k	30	W m ⁻¹ °K ⁻¹
<i>tkair</i>	119	k _{air}	0.05	W m ⁻¹ °K ⁻¹
<i>tkair</i>	153-162	k _{air}	0.06	W m ⁻¹ °K ⁻¹
<i>tkliquid</i>	95	k _{liq}	0.8	W m ⁻¹ °K ⁻¹
<i>tkmold</i>	138	k _m	314.7	W m ⁻¹ °K ⁻¹
<i>akscale</i>	153-162	k _{scale}	0.55	W m ⁻¹ °K ⁻¹
<i>tksolid</i>	94	k _{sol}	0.8	W m ⁻¹ °K ⁻¹
<i>tkwaterw</i>	128	k _{water}	0.615	W m ⁻¹ °K ⁻¹
<i>dchannel</i>	141	L _{ch}	29	mm
<i>width</i>	122	L _{mark}	5	mm
<i>expn</i>	102	n	0.85	-
<i>rindex</i>	99	m	1.5	-
<i>bdata</i>	28	q _{int}	-	kW m ⁻²
<i>qdata</i>	34	q _{sh}	-	kW m ⁻²
<i>cons</i>	104	Q _f	0.4	kg m ⁻²
<i>wflow</i>	81-85	Q _w	2.0	l /min
<i>amrada</i>	149	R _I	11.76	m
<i>amrad</i>	150	R _O	12	m
<i>tin</i>	71	T ₀	35	°C

<i>tcrystal</i>	93	T_{fsol}	920	$^{\circ}\text{C}$
<i>tinit</i>	12	T_{pour}	1550	$^{\circ}\text{C}$
<i>twater</i>	139	T_{water}	30	$^{\circ}\text{C}$
<i>vcmmin</i>	11	V_{c}	1.07	m min^{-1}
<i>yslab</i>	14	W	1774	mm
<i>chwidth</i>	142	w_{ch}	5	mm
<i>dmen</i>	136	z_{men}	100	mm
<i>drim1</i>	106	z_{rim1}	0	mm
<i>drim2</i>	107	z_{rim2}	0	mm
<i>zmold</i>	15	z_{mold}	810	mm
<i>delh</i>	61	ΔH_{L}	271	kJ kg^{-1}
<i>dtime</i>	38	Δt	0.02	s
<i>ecopper</i>	118	ϵ_{m}	0.5	-
<i>esteel</i>	62	ϵ_{s}	0.8	-
<i>mu</i>	96	μ_0	1.1	poise
<i>amuwatw</i>	129	μ_{water}	0.0008	Pa-s
<i>dense</i>	60	ρ	7400	kg m^3
<i>rhoflux</i>	97	ρ_{flux}	2500	kg m^{-3}
<i>rhowater</i>	131	ρ_{water}	995.6	kg m^3
<i>dzone</i>	81-85			mm
<i>fflux</i>	29		0, 1 or -1	-
<i>fracrol</i>	81-85			-
<i>freqz</i>	40		10	mm
<i>i2d</i>	35		0 or 1	-
<i>ibound</i>	23		0, 1 or -1	-
<i>iseflag</i>	67		0, or -1	-
<i>imoldflq</i>	21		0, 1 or 2	-
<i>inaro</i>	20		0 or 1	-
<i>ioscflag</i>	120		0 or 1	-
<i>irollsw</i>	81-85			-
<i>itmax</i>	46		25000	-
<i>nbdata</i>	25			-
<i>nconsf</i>	103			-
<i>nflux</i>	31			-
<i>AN</i>	165			-
<i>npp</i>	151		5	-
<i>nstep</i>	39		50	-
<i>nthc</i>	171		6	-

<i>nvwater</i>	145		-
<i>nzone</i>	77	5	-
<i>pc pmn ps</i>	51		
<i>pp psi</i>			%
<i>pcr pni</i>	52		%
<i>pcu pmo pti</i>			
<i>pal pv</i>	53		%
<i>pn pnb pw</i>			
<i>pco</i>	54		%
<i>pwater</i>	140	0.202	MPa
<i>rrcontact</i>	117	5.e-9	m ² °K W ⁻¹
<i>RI,RO</i>	167,168	5.e-3, 6.5e-2	mm
<i>rollradw</i>	81-85		m
<i>stroke</i>	125	10	mm
<i>submerg</i>	17	265	mm
<i>thmax</i>	43	115	mm
<i>totcharea</i>	143		mm ²
<i>vwater</i>	146	7.8	m s ⁻¹
<i>xshlt</i>	47	10	mm
<i>xslabmm</i>	13	230	mm
<i>xthc</i>	174-178		mm
<i>zana</i>	37	700	mm
<i>zbdata</i>	27		mm
<i>zhbal</i>	16	809	mm
<i>zmax</i>	42	811	mm
<i>zpeak</i>	105		m
<i>zpmin</i>	41	0	mm
<i>nnthc</i>	174-178		-
<i>icp</i>	55		-
<i>nvc</i>	7		-
<i>vctime</i>	9		s
<i>slig</i>	58		°C
<i>ssol</i>	59		°C
<i>alpha</i>	65		-
<i>eslag</i>	101		-
<i>nratio</i>	111		-
<i>zratio</i>	115		-

<i>wcao wsio2 wngo</i>	89		%
<i>wna2o wk2o</i>			
<i>wfeo wfe2o3 wnio</i>	90		%
<i>wmno wcr2o3</i>			
<i>wal2o3 wtio2</i>	91		%
<i>wb2o3 wlio2 wsro</i>			
<i>wzro2 wf wfreec</i>	92		%
<i>wtotc wco2</i>			
<i>zqdata</i>	33		mm
<i>zthc</i>	174-178		mm

Reference

- [1] Chandra S., Brimacombe J.K., and Samarasekera I.V. Ironing and Steelmaking, Vol. 20, 1993, n.2, pp.104-112
- [2] Nozaki, T., Matsuno, J., Murata, K., Ooi, H., Kodama, M., “A Secondary Cooling Pattern for Preventing Surface Cracks of Continuous Casting Slab”, Trans. ISIJ, V. 18, 1978, pp.330-338
- [3] J. K. Brimacombe, P.K. Agarwal, S. Hibbins, B.Prabhuker, L.A. Baptista, Spray Cooling in the Continuous Casting of Steel , Continuous Casting Vol. 2, ISS-AIME, 1984, pp.109-123

mass consumption rate of powder flux:

$$Q(\text{kg/min}) = \frac{Q(\text{kg/t}) * N * W * V_c * \rho}{1000} = \frac{.46 * .25 * 1 * 1.05 * 7330}{1000} = 0.89 \text{ kg/min}$$