CON2D Getting Started

1. System Requirement
   a) CON2D can be compiled and run on generic UNIX system with FORTRAN 77 compiler.
   b) The text version of post processor, POSTT, can be compiled and run on generic UNIX system with FORTRAN 77 compiler.
   c) The X windows version of post processor, POSTX, requires HOOPS software installed in your UNIX system. Moreover, the following sentences have to be appeared in your environment settings to make the graphic windows to be displayed correctly:

   HOOPS_LICENSE=customer = uofillinois, product = (classic), key = 255ACE61-B29ED1A-5161AF8B-258ED3F
   HOOPS_SYSTEM_OPTIONS=license = (customer = uofillinois, product = (classic), key= 255ACE61-B29ED1A-5161AF8B-258ED3F)
   DISPLAY=bgtbim1.me.uiuc.edu:0.0
   HOOPS_PICTURE=x11/bgtibm1.me.uiuc.edu:0.0

   Certainly, you may have different keys and customer names in your system. The DISPLAY and HOOPS_PICTURE have to have the same displaying client to show the graphic windows.

2. Compiling and Installation
   a) Download the con2d.tar, con2dpostt.tar or con2dpostx.tar into your machine from CCC (Continuous Casting Consortium) member's page or FTP site.
   b) Un-tar the source code using the following command:

   tar –xf [filename].tar [destinate directory]

   c) Modify the BINDIR entry in the *makefile* to the directory which you want to put the executive files in.

   d) Compile your executives by using:

   make

3. Input Description

VALUE	VARIABLE NAME	REMARKS
1	*ioutput*	*Output Type [1=no, 2=short, 3=medium]*
UBC test run		*Run title*
3	*ianalyz*	*Analysis Type [1=Heat Transfer,2=Stress,3=Coupled]*
3	*iz*	*Stress State [1=Pl Strain,2=Pl stress,3=GPS,4=GGPS]*
1	*isimul*	*Simulation Zone [1=Mold,2=Mold to end,3=end to RH,4=RH*
0	*imoldis*	*Flag for mold distortion 1=ON, others=OFF*
y	*meshdec*	*To use previous meshfile: y otherwise n (in Column 1)*
Lmesh01	*meshname*	*Name of the newly created Mesh file*
n	*str0dec*	*For Initial Stress file: y else n (no stress)*
1576.00	*temp0*	*Initial temp of the strand*
1	*ifixt*	*Flag for fix temp data 1=Y,2=N*
mt01.shl	*fixtname*	*Name of file with temp data*
2	*iflux*	*Flag for Super heat flux 1=Y,2=N*
2	*isteel*	*Steel Type [1=Austenitic Stainless, 2=Plain-Carbon Steel, 3 = Test-Properties, 4 = Ferritic Stainless]*
0.003	*pc*	*Carbon Content*
1.096 (m)	*rmolng*	*Mold Length*
.0254 (m/s)	*casvel*	*Casting Speed*
.0000 (s)	*tinit*	*Initial time*
45.00 (s)	*tmax*	*Final time*
100001	*imax*	*Maximum Number of time steps*
0	*icurve*	*Mold Type 0= Straight, 1=Curved*
2	*itaper*	*Type of Mold Taper: 1=ideal,2=linear,3=Tri-linear*
.0 .0 (%/m)	*taprw1 taprn1*	*Wide & Narrow face taper*
4	*numstep*	*No. of diff time step sizes*
.010 0.500	*dtstep stepchg*	*time step size(s) final time (s)*

.020 1.000	*dtstep stepchg*	*time step size(s) final time (s)*
.050 5.000	*dtstep stepchg*	*time step size(s) final time (s)*
.100 45.00	*dtstep stepchg*	*time step size(s) final time (s)*
1	*idef*	*Heat Transfer soln procedure(default=1):*
3	*isloads*	*choice of surface load handling*
0	*force*	*surface load (-99999 for user defined subroutine in user.f)*
0	*ipress*	*Flag for Ferrostatic Pressure:* *1=ON,2=OFF*
0	*imoldf*	*Flag for Mold wall restraint:* *1=ON, 2=OFF*
2	*ipsfunc*	*Flag for plasticity:* *0=Elastic,1=Pl Str Rate,2=Tot Pl Str*
9	*iconlaw*	*Constitutive Law:* *1: Strain hardening constitutive law (Kowlowski Model II)* *2: Strain hardening constitutive law (Kowlowski Model III)* *3: Vacant* *4: Steady State Creep Law* *5: Elasto Plastic Creep Law (Weak Powered -- n=1)* *6: Elasto Plastic Creep Law (Strong Powered – n=5)* *7: Vacant* *8: Strain and time hardening constitutive law* *9: Kowlowski Model III enhanced by Power Law in ? region*
1.0000	*dispcnv*	*HT Conv parameter, >1.0 suppresses HT conv iteration*
4	*numprint*	*Number of print interval desired*
.050 0.500	*printi timeprnt*	*Print interval(s) Final time(s)*
.100 1.000	*printi timeprnt*	*Print interval(s) Final time(s)*
.500 5.000	*printi timeprnt*	*Print interval(s) Final time(s)*
1.00 45.00	*printi timeprnt*	*Print interval(s) Final time(s)*
y	*outask*	*Flag for similar print interval as binout y or n*