```
"Author: Bertrand Schyns"

"Procedure to create the evolution of the current in time"
Procedure current(tau:I_curr)
If (tau<5+0,1) Then
    I_curr = 25,2[Amp] "The current is stable over the fivefew seconds"
Endif

If (tau>5) AND (tau<10+0,1) Then
    I_curr = 25,2+(tau-5)/5*(35,6-25,2) "The current will go from 25,2 to 35,6 in 5s"
Endif

If (tau>5) Then
    I_curr = 35,6[Amp]
Endif


End

Procedure massflux(tau:G_chan)
If (tau<5+0,1) Then
    G_chan = 490,29 [kg/m^2-s] "The mass flux is stable over the first few seconds"
Endif

If (tau>5) AND (tau<10+0,1) Then
    G_chan = 490,29+(tau-5)/5*(578,7-490,29) "The mass flux is varying in 5s"
Endif

If (tau>10) Then
    G_chan = 578,70[kg/m^2-s]
Endif
End


"Procedure to calculate the heat capacity of the side edge part"
Procedure c_se(T_SE;m_SE:C_int_SE)
cp_se = cp(Inconel_718; T=T_SE)
C_int_SE = m_SE*cp_se
End

"Procedure to calculate the heat capacity from the center of the material underneath the flow channel"
Procedure c_ch(T_ch;m_ch:C_int_ch)
cp_ch = cp(Inconel_718; T=T_ch)
C_int_ch = m_ch*cp_ch
End

"Procedure to calculate the heat capacity of the channel side walls"
Procedure c_sw(T_SW;m_SW:C_int_SW)
cp_sw = cp(Inconel_718; T=T_SW)
C_int_SW = m_SW*cp_sw
End

"Procedure to calculate the heat capacity of the fluid"
Procedure c_f(T_F;P_F;m_F:C_int_F)
cp_F = cp(CarbonDioxide;T=T_F;P=P_F)
C_int_F = m_F*cp_F
End

"Procedure for the n value for the Jackson and Hall (1979) correlation"
Procedure nval_jackson (T_PC; T_W;T_F: n_val)
T_PC_K=converttemp(C;K;T_PC)                                         "Converting
the pseudo-Critical temperature from C to K"
T_W_K=converttemp (C;K;T_W )                                         "Converting
the wall temperature from C to K"
T_F_K=converttemp (C;K;T_F )                                         "Converting
the bulk fluid temperature from C to K"

"Conditional statements for calculating the n values to be used in the Jackson and Hall (1979) correlation"
```

```
If ((T_F_K<T_W_K) AND (T_W_K<T_PC_K)) OR ((1,2*T_PC_K<T_F_K) AND (T_F_K<T_W_K)) Then "Case with nothing accros the PC point"
n_val =0,4
Endif

If ((T_F_K<T_PC_K) AND (T_PC_K<T_W_K)) Then
n_val = 0,4+ 0,3* ((T_W_K/T_PC_K)-1)
Endif

If ((T_PC_K<T_F_K) AND (T_F_K< 1,2*T_PC_K)) AND (T_F_K<T_W) Then
n_val = 0,4+ 0,2 * ((T_W_K/T_PC_K)-1)* (1-5*((T_W_K/T_PC_K)-1))
Else
n_val=0,4
Endif



End


"Subprogram for Jackson and Hall (1979) correlation"

Subprogram jackson (P_sCO2;G_chan;D_h;T_W;T_F;n_val:Nus_b;Re_b;St)

rho_w= Density (CarbonDioxide; T= T_W;P=P_sCO2)          "Density of the fluid evaluated at the wall temperature"
rho_b=Density (CarbonDioxide;T=T_F;P=P_sCO2)             "Density of the fluid evaluated at the bulk fluid temperature"
h_w= enthalpy(CarbonDioxide; T=T_W;P=P_sCO2)             "Enthalpy of the fluid evaluated at the wall temperature"
h_b=enthalpy(CarbonDioxide;T=T_F;P=P_sCO2)               "Enthalpy of the fluid evaluated at the bulk fluid temperature"
C_p_b=cp(CarbonDioxide;T=T_F;P=P_sCO2)                   "Specific heat capacity of the fluid evaluated at the bulk fluid temperature"
T_W_K= converttemp(C;K;T_W)                              "Converting the wall temperature from C to K"
T_F_K=converttemp(C;K;T_F)                               "Converting the bulk fluid temperature from C to K"
c_p_bar= (h_w-h_b)/(T_W_K-T_F_K)                         "Integrated specific heat capacity"
Pr_b=prandtl(CarbonDioxide;T=T_F;P=P_sCO2)               "Prandtl number evaluated at the bulk fluid temperature"
mu_b=viscosity(CarbonDioxide;T=T_F;P=P_sCO2)             "Viscosity evaluated at the bulk fluid temperature"
Re_b= (G_chan*D_h)/mu_b                                  "Reynolds number"

Nus_b=0,01831*Re_b^(0,82)*Pr_b^(0,5)* (rho_w/rho_b)^(0,3)* (c_p_bar/C_p_b)^(n_val)
St= Nus_b/Re_b*Pr_b
End

"Procedures for several resistance to be used in the governing equations"


"Procedure for the thermal resistances in the side edge of the bottom portion of the test section"

Procedure r_ech (Th_SE;W_ch;T_int_se1;T_int_ch;L_CV;Th_B;DELTA_seg:R_ax_E_ch)

DELTA_nodes = Th_SE/2 + W_ch/2              "Distance between the node in the edge and the one underneath the flow channel"
T_av1= average(T_int_se1;T_int_ch)          "Average of the temperatures of the nodes in the side edge and the one underneath the flow channel"
k_1=conductivity(Inconel_718; T=T_av1)      "Thermal conductivity of the test section material evaluated at the average temperature"
A_cross1= L_CV*Th_B                         "Cross sectional area for conduction"
R_ax_E_ch = DELTA_nodes/(k_1*A_cross1)      "Thermal resistance to heat transfer from the edge to the location underneath the flow channel"
```

```
End


Procedure r_ee (Delta_seg;T_int_se1;T_int_se2;Th_SE;Th_B:R_ax_EE)
T_av2=average(T_int_se1;T_int_se2)          "Average of the temperature of the adjacent nodes inside the side edge of the test section"
K_2=conductivity(Inconel_718;T=T_av2)       "Thermal conductivity"
A_cross2=Th_SE*Th_B                         "Cross sectional area for conduction between adjacent nodes in the side edge"
R_ax_EE= Delta_seg/(K_2*A_cross2)           "Thermal resistance to heat transfer- axial conduction inside the side edge"
End

"Procedure for the calculation of thermal resistance from the center of the material underneath the flow channel to the wall of the flow channel"

Procedure r_chw (Th_B;T_W;T_int_ch;L_CV;W_ch:R_Ax_ch_w)
DELTA_res= Th_B/2                           "Distance from the node to the bottom of the flow channel wall"
T_av=average(T_W;T_int_ch)                  "Average of the temperatures"
k=conductivity(Inconel_718;T=T_av)          "Thermal conductivity"
A_cross= L_CV*W_ch                          "Cross sectional area for heat transfer"
R_Ax_ch_w= DELTA_res/(k*A_cross)            "Thermal resistance to the flow channel wall"
End

"Procedure for calculation of thermal resistance from the node underneath the flow channel to the one underneath the channel side wall"
 Procedure r_chsw (W_ch;Th_SW;T_int_ch;T_int_sw;L_CV;Th_B:R_ax_ch_sw;tau_diff)
DELTA_res = (W_ch/2 + Th_SW/4)              "Distance between the nodes"
T_Av=average(T_int_ch;T_int_sw)             "Average of the temperatures"
k=conductivity(Inconel_718;T=T_Av)          "Thermal conductivity of the bottom portion of the test section"
A_cross=L_CV*Th_B                           "Cross sectional area for heat transfer"
R_ax_ch_sw= DELTA_res/(k*A_cross)           "Thermal resistance"
rho=density(Inconel_718;T=T_Av)             "Density of Iconel"
c=cp(Inconel_718; T=T_Av)                   "specific heat capacity"
alpha=k/(rho*c)                             "Thermal diffusivity"
tau_diff=Th_B^2/(4*alpha)                   "Time for the thermal wave to reach the outer edge of the wall"
End

 "Procedure for calculation of thermal resistance for axial conduction underneath the flow channel"
Procedure r_axch (DELTA_Seg;T_int_ch1;T_int_ch2;Th_B;W_ch:R_ax_ch)
DELTA_res = DELTA_Seg                       "Distance between the nodes"
T_av=average(T_int_ch1;T_int_ch2)           "Average of the temperatures"
k=conductivity (Inconel_718;T=T_av)         "Thermal conductivity of the bottom portion of the test section"
A_cross=Th_B*W_ch                           "Cross sectional area for condutcion heat transfer"
R_ax_ch= DELTA_res/(k*A_cross)              "Thermal resistance for axial conduction underneath the flow channel"
End


"Procedure for axial conduction underneath the channel side walls"

Procedure r_axsw (DELTA_seg;T_int_sw1;T_int_sw2;Th_SW;Th_B:R_ax_sw)
DELTA_res=DELTA_seg                         "Disatnce between the nodes"
T_av=average(T_int_sw1;T_int_sw2)           "Average of temperatures"
k=conductivity(Inconel_718;T=T_av)          "Thermal conductivity of the bottom portion of the test section"
A_cross=Th_SW*Th_B/2                        "Cross sectional area for thermal conduction"
R_ax_sw= DELTA_res/(k*A_cross)              "Thermal resistance for axial conduction underneath the channel edge"
End

"Procedure for the convective resistance"

 Procedure r_convective (Nus_b;P_sCO2;T_F;D_h;L_CV;W_ch:R_conv;h_conv)
 k=conductivity(CarbonDioxide; T=T_F; P=P_sCO2)                   "Thermal conductivity of the fluid"
h_conv=(Nus_b*k)/D_h                                              "Heat transfer coefficient"
```

```
A_conv = L_CV*W_ch                                                    "Heat transfer area for convection"
R_conv= 1/(h_conv*A_conv)                                             "Convective resistance"
End


"Procedure for calculating the resistance associated with the conduction dominated region of the boundary layer"

Procedure bl_cond (delta_cl;T_W;L_CV;W_ch;P_sCO2:R_cond_bl)
k_w = conductivity(CarbonDioxide;T=T_W;P=P_sCO2)                      "Thermal conductivity of the fluid evaluated at the wall temperature"
A_cond = L_CV*W_ch                                                    "Heat transfer area for conduction"
R_cond_bl ={ 200} delta_cl/(k_w*A_cond)                               "Resistance associated with the conductive portion of the boundary layer"

End



"-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------"
"Procedure for heat generation inside the bottom portion of the test section"
Procedure heat_gen (N_ch;L_B;T_int_se;T_int_ch;T_int_sw;Th_SE;Th_B;W_ch;Th_SW;L_CV;I_density:q_gen_se;q_gen_ch;q_gen_sw)
T_1= T_int_se
rho_e_se=electricalresistivity(Inconel_718; T=T_1)                    "Electrical resistivity in the side edge - bonding region"
rho_e_ch=electricalresistivity(Inconel_718; T=T_int_ch)               "Electrical resistivity underneath the flow channel"
rho_e_sw=electricalresistivity(Inconel_718; T=T_int_sw)               "Electrical resistivity in the channel side wall"
A_cross_se=Th_SE*Th_B
    "Cross sectional area for current flow in the side edge"
A_cross_ch= W_ch*Th_B
    "Cross sectional area for current flow underneath the flow channel"
A_cross_sw= (Th_SW/2)*Th_B
    "Cross sectional area for the current flow in the side wall"
R_se=(L_CV*rho_e_se)/A_cross_se
    "Electrical resistance in the side edge of the test section"
R_ch=(L_CV*rho_e_ch)/A_cross_ch
    "Electrical resistance underneath the flow channel"
R_sw=(L_CV*rho_e_sw)/A_cross_sw
    "Electrical resistance in the side wall"
q_gen_se= (I_density*A_cross_se)^2*R_se                              "Heat generation inside the side edge"
q_gen_ch=(I_density*A_cross_ch)^2*R_ch                               "Heat generation under the flow channel"
q_gen_sw=(I_density*A_cross_sw)^2*R_sw                               "Heat generation in the channel side wall"

W_total = (2*Th_SE)+(N_ch-1)*(Th_SW)+N_ch*W_ch                        "Total width of the test section"
A_cross_total = W_total*Th_B
    "Total cross sectional area for current flow"
R_total= (rho_e_ch*L_B)/A_cross_total
End

 "Procedure for evaluating the buoyancy threshold criterion proposed by Petukhov and flow acceleration effects according to Jackson's criterion"
 Procedure grashof (T_W;T_F;D_h;P_sCO2;q_flux;G_chan;Re_b;per;per_heated:Bo_pet;Psi_Jackson;K_v)
 T_av=average(T_W;T_F)
    "Film temperature- average of the wall temperature and the bulk fluid temperature"
rho_b= density(CarbonDioxide;T=T_F;P=P_sCO2)                          "Density of carbon dioxide evaluated at the bulk fluid temperature"
rho_w=density(CarbonDioxide;T=T_W;P=P_sCO2)                           "Density of
```

```
carbon dioxide evaluated at the channel wall temperature"
rho_film=density(CarbonDioxide;T=T_av;P=P_sCO2)                                   "Density of
carbon dioxide evaluated at the film temperature"
mu_b=viscosity(CarbonDioxide;T=T_F;P=P_sCO2)                                      "Viscosity
of carbon dioxide evaluated at the bulk fluid temperature"
nu_b=kinematicviscosity(CarbonDioxide;T=T_F;P=P_sCO2)                             "Kinematic
viscosity of carbon dioxide evaluated at the bulk fluid temperature"
h_w=enthalpy(CarbonDioxide;T=T_W;P=P_sCO2)                                        "Enthalpy
of carbon dioxide evaluated at the channel wall temperature"
h_b=enthalpy(CarbonDioxide; T=T_F;P=P_sCO2)                                       "Enthalpy
of carbon dioxide evaluated at the bulk fluid temperature"
k_b=conductivity(CarbonDioxide; T=T_F;P=P_sCO2)                                   "
Conductivity of carbon dioxide evaluated at the bulk fluid temperature"
Pr_bar = (h_w-h_b)/(T_W-T_F) * (mu_b/k_b)                                            "
Integrated Prandtl number"
Beta_bar= 1/(rho_film)* ((rho_b-rho_w)/(T_W-T_F))
Gr_q= (g#*Beta_bar*q_flux*D_h^4)/ (nu_b^2*k_b)
Gr_th=3*10^(-5)*Re_b^(2,75)*Pr_bar^(0,5)*(1+2,4*Re_b^(-1/8)*(Pr_bar^(2/3)-1)) "!see Petukhov"
 Bo_pet = Gr_q/Gr_th
"---------------------------------------------------------------------------------------------------------------------------------------------------------
-----------------------------------------------------------------------------------------------------"
"Jackson's flow acceleration"
Beta_B=volexpcoef(CarbonDioxide;T=T_F;P=P_sCO2)                                      "
Volume expansion coefficient for carbon dioxide evaluated at the bulk fluid temperature"
Pr_b=Prandtl (CarbonDioxide;T=T_F;P=P_sCO2)                                          "
Prandtl number for carbon dioxide evaluated at the bulk fluid temperature"
mu_film=viscosity(CarbonDioxide;T=T_av;P=P_sCO2)                                  "Viscosity
of carbon dioxide evaluated at the film temperature"
F_VP1= (mu_film/mu_b)*(rho_film/rho_b)^(-0,5)                                        "
Jackson's property function"
A_cb = (Beta_B*q_flux*D_h)/(k_b*Re_b^(1,625)*Pr_b)                                "
Acceleration function"
C_A= 10^4                                                                         "Constant"
Psi_Jackson= C_A*A_cb*F_VP1                                                       "Jackson's
accelration parameter"

"Jackson acceleration parameter applied to asymmetric heating"
cp_b=cp(CarbonDioxide;T=T_F;P=P_sCO2)                                             "Specific
heat capacity  for carbon dioxide evaluated at the bulk fluid temperature"

q_plus = Beta_B*q_flux/(G_chan*cp_b);
K_v = 4*(per_heated/per)*q_plus/Re_b;
End

"Procedure for evaluating the bulk fluid velocity in the channel"
Procedure velocity (N_ch;W_ch;H_ch;D_h;AR;m_dot_total;T_F;T_W;P_sCO2;L_ax:V_channel;nu_acc;span_spacing;
y_plus;Re_chan;F_channel;dp_dx;dp;rho_bulk;DELTA_Bankston;DELTA_P_W;tau_near_wall;u_friction;rho_wall;delta_cl)
A_chan= W_ch*H_ch                                                                 "Flow area
of a single channel"
m_dot_chan= m_dot_total/N_ch                                                      "Mass flow
rate through a single channel"
G_chan= m_dot_chan/A_chan                                                         "Mass flux
through a single channel"
rho_bulk= density(CarbonDioxide;T=T_F;P=P_sCO2)                                   "Density of
the fluid evaluated at the bulk fluid temperature"
rho_wall= density(CarbonDioxide;T=T_W;P=P_sCO2)                                   "Density of
the fluid evaluated at the wall temperature"
mu_wall=viscosity(CarbonDioxide; P=P_sCO2;T=T_W)                                     "
Viscosity of the fluid evaluated at the wall temperature"
V_channel= G_chan/rho_bulk                                                        "Bulk fluid
velocity in the channel"
nu_acc=kinematicviscosity(CarbonDioxide;T=T_W;P=P_sCO2)                           "Kinematic
viscosity evaluated at the wall temperature"
RR= AR/D_h                                                                        "Relative
roughness for the flow channel"
mu_b=viscosity(CarbonDioxide;T=T_F;P=P_sCO2)                                      "Viscosity
evaluated at the bulk fluid temperature"
Re_chan= (G_chan*D_h)/mu_b                                                        "Reynolds
```

```
number in the channel"
Theta_1_pipe=(2,457*ln(((7/Re_chan)**0,9+0,27*RR)**(-1)))**16          "!Churchill
Friction factor"
Theta_2_pipe=(37530/Re_chan)**16
F_channel=8*((8/Re_chan)**(12)+(Theta_1_pipe+Theta_2_pipe)**(-1,5))**(1/12)          " Darcy
friction factor for turbulent flow in channels"
tau_s= (rho_wall*F_channel*V_channel^2)/8
dp_dx= (F_channel*rho_bulk*V_channel^2)/(2*D_h)          "Pressure
drop/length"
dp=dp_dx*L_ax
f_flowacc= 0,079*Re_chan^(-0,25)
tau_wall= (rho_bulk*V_channel^2)*f_flowacc/2
 u_friction=sqrt(tau_wall/rho_bulk)          "Friction
velocity"
 Span_spacing_ND = 100          "Non-
dimensional span wise spacing of coherent structures"
span_spacing=  (Span_spacing_ND*nu_acc)/u_friction
y_star= nu_acc/u_friction          "Viscous
length scale"
y_plus=H_ch/y_star
delta_vs_plus = 5 [-]          "
Dimensionless thickness of the viscous sub-layer"
Pr_w = prandtl(CarbonDioxide;T=T_W;P=P_sCO2)          "Prandtl
number evaluated at the wall temperature"
delta_cl_plus = delta_vs_plus/(Pr_w^(1/3))          "
Dimensionless thickness of the conduction dominated region of the boundary layer"
delta_cl = delta_cl_plus*y_star          "
Dimensional thickness of the conduction dominated region of the boundary layer"
DELTA_Bankston = (nu_acc/(rho_wall*u_friction^3))*dp_dx

DELTA_P_W= (-4*mu_wall/D_h)* (1/sqrt(rho_wall*tau_s))
 tau_near_wall =rho_wall*u_friction^2
C_f= 0,0376*(Re_chan)^(-1/6)          "Skin
friction coefficient as defined in Patel and Head 1968"
DELTA_patel= -nu_acc/(rho_wall*u_friction^3)*dp_dx
End

"Procedure implementing Wien's law of radiation"
Procedure wien (T_BW:Lambda_max;F_rad)
C_rad= 2897,8
T=converttemp(C;K; T_BW)
Lambda_max=C_rad/T
Lambda_1= 7,5
Lambda_2 = 13
F_rad=blackbody(T;Lambda_1;Lambda_2)
End


"-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------"

"Simulation specification in time"
tau_1=0
tau_2=20
dtau = 0,25

"Initial value for the iterations taken from the steady state point at I_curr = 25,2A"
T_int_se_i[1..40] = [60,13;60,13;60,11;60,09;60,06;60,03;60,01;59,98;59,95;59,92;59,9;59,87;59,85;59,82;59,8;59,77;59,
75;59,73;59,71;59,69;59,67;59,65;59,63;59,61;59,59;59,57;59,56;59,54;59,53;59,52;59,5;59,49;59,48;59,47;59,46;59,45;
59,44;59,44;59,43;59,43]
T_int_ch_i[1..40] = [39,61;39,59;39,57;39,54;39,51;39,48;39,45;39,42;39,39;39,36;39,33;39,3;39,27;39,25;39,22;39,2;39,
17;39,15;39,12;39,1;39,08;39,06;39,04;39,02;39;38,98;38,96;38,95;38,93;38,92;38,9;38,89;38,88;38,87;38,86;38,85;38,84;
38,83;38,82;38,82]
T_int_sw_i[1..40] = [47,8;47,78;47,76;47,73;47,71;47,68;47,65;47,62;47,59;47,56;47,54;47,51;47,48;47,46;47,43;47,41;47,
38;47,36;47,34;47,31;47,29;47,27;47,25;47,23;47,21;47,2;47,18;47,16;47,15;47,14;47,12;47,11;47,1;47,09;47,08;47,07;47,
06;47,05;47,05;47,04]
T_F_i[1..40] = [32,73;32,74;32,75;32,75;32,76;32,76;32,77;32,77;32,78;32,79;32,79;32,8;32,8;32,81;32,81;32,82;32,82;32,
83;32,83;32,83;32,84;32,84;32,85;32,85;32,86;32,86;32,87;32,87;32,87;32,88;32,88;32,89;32,89;32,89;32,9;32,9;32,91;32,
```

```
91;32,91;32,92]


"Equation to use instead of the balance equation to update the different guesses"
{Duplicate i=1;N_nodes
T_int_se_i[i]= T_int_se[i]
T_int_ch_i[i] = T_int_ch[i]
T_int_sw_i[i] = T_int_sw[i]
T_F_i[i] = T_F[i]
End}

"Dimensions of the bottom portion of the test section"
Th_B=254*convert(micron;m)          "Thickness of the bottom Inconel 718sheet"
L_B= 50*convert(mm;m)               "Length of the bottom Inconel 718 sheet"
W_ch= 1,92*convert(mm;m)            "Width of the flow channel"
Th_SW=1,5*convert(mm;m)             "Width of the wall seperating the channels"
Th_SE=1,5*convert(mm;m)             "Width of the side edge used for bonding the Inconel 718 to Macor"

 AR= 0,78*convert(micron;m)          "Absolute roughness of the channel walls"


"Flow channel hydraulic diameter"
H_ch= 600*convert(micron;m)         "Height of the flow channels"
A_cross_flow = H_ch*W_ch            "Flow cross sectional area"
per= 2*(H_ch+W_ch)                  "Perimeter"
per_heated = W_ch                   "Heated perimeter"
D_h= (4*A_cross_flow)/per           "Hydraulic diamter"


"Flow conditions"
P_sCO2_psi= 1120,2 [psi]                          "Absolute pressure in the test section in psi"
P_sCO2 =P_sCO2_psi*convert(psi;kPa)               "Absolute pressure in the test section in kPa"
P_sCO2_bar= P_sCO2_psi*convert(psi;bar)           "Absolute pressure in the test section in bar"
{Call massflux(tau:G_chan)}                       "Function used for a varying Mass flux"
G_chan = 411,61 [kg/m^2-s]                        "Mass flux in a single channel"
m_dot_chan= G_chan*A_cross_flow                   "Mass flow rate in a single channel"
m_dot_total=m_dot_chan*N_ch                       "Total mass flow rate through the test section"
G_total=G_chan
N_ch=3
 T_F_in = 32,73 [C]                               "Inlet temperature"
"Defining the pseudo-critical temperature"
T_PC= -122,6+6,124*P_sCO2_bar-0,1657*P_sCO2_bar^2+0,01773*P_sCO2_bar^(2,5)-0,0005608*P_sCO2_bar^3  " Pseudo-critical temperature - Liao and Zhao (2002)"


"Current" "System tested for the current fixed at 55A and the results are the same tham for the steady state cae for the whole period of testing"
Call current(tau:I_curr)
{I_curr=25,2[Amp]}                                "Current through the test section"

W_total = (2*Th_SE)+(N_ch-1)*Th_SW+N_ch*W_ch      "Total width of the test section"
A_cross_total = W_total*Th_B                      "Total cross sectional area for current flow"
I_density= I_curr/A_cross_total
```

```
"Segmenting the model in the third dimension"
N_nodes =40[-]                    "Number of nodes in the bottom portion of the test section"
L_begin = 0 [m]                   "Left edge of the test section"
DELTA_seg= L_B/(N_nodes-1)        "Distance between the nodes"
 "Assigning the position to the nodes"
Duplicate i= 1;N_nodes
L_ax[i]= L_begin+ DELTA_seg *(i-1)
End

"Assigning the length and the mass to the control volumes"  "Works fine before adding the differential equations"
rho_Inc=density(Inconel_718; T=25)
L_CV[1]= DELTA_seg/2              "Length of the first control volume"
m_CV_SE[1] = L_CV[1]*Th_SE*Th_B*rho_Inc "Mass of the side edge"
m_CV_ch[1] = L_CV[1]*W_ch*Th_B*rho_Inc "Mass of the middle part bellow the fluid"
m_CV_SW[1] = L_CV[1]*Th_SW/2*Th_B*rho_Inc "Mass of the side wall separating the channels"
m_CV_F[1] =  L_CV[1]*W_ch*H_ch*rho_sCO2[1]
rho_sCO2[1] =density(CarbonDioxide;T=T_F[1];P=P_sCO2)
L_CV[N_nodes]= L_CV[1]            "Length of the last control volume"
m_CV_SE[N_nodes] = L_CV[N_nodes]*Th_SE*Th_B*rho_Inc "Mass of the side edge"
m_CV_ch[N_nodes] = L_CV[N_nodes]*W_ch*Th_B*rho_Inc "Mass of the middle part bellow the fluid"
m_CV_SW[N_nodes] = L_CV[N_nodes]*Th_SW/2*Th_B*rho_Inc "Mass of the side wall separating the channels"
m_CV_F[N_nodes] =  L_CV[N_nodes]*W_ch*H_ch*rho_sCO2[N_nodes]
rho_sCO2[N_nodes] =density(CarbonDioxide;T=T_F[N_nodes];P=P_sCO2)
"Length of the control volumes in the internal control volumes"
Duplicate i=2;(N_nodes-1)
L_CV[i]= DELTA_seg
m_CV_SE[i] = L_CV[i]*Th_SE*Th_B*rho_Inc "Mass of the side edge"
m_CV_ch[i] = L_CV[i]*W_ch*Th_B*rho_Inc "Mass of the middle part bellow the fluid"
m_CV_SW[i] = L_CV[i]*Th_SW/2*Th_B*rho_Inc "Mass of the side wall separating the channels"
m_CV_F[i] =  L_CV[i]*W_ch*H_ch*rho_sCO2[i]
rho_sCO2[i] =density(CarbonDioxide;T=T_F[i];P=P_sCO2)
End

 "Calling all the procedures here"

Duplicate i=1;N_nodes
Call c_se(T_int_se[i];m_CV_SE[i]:C_int_SE[i])
Call c_ch(T_int_ch[i];m_CV_ch[i]:C_int_ch[i])
Call c_sw(T_int_sw[i];m_CV_SW[i]:C_int_SW[i])
Call c_f(T_F[i];P_sCO2;m_CV_F[i]:C_int_F[i])

Call nval_jackson (T_PC; T_W[i];T_F[i]: n_val[i])
Call  jackson (P_sCO2;G_chan;D_h;T_W[i];T_F[i];n_val[i]:Nus_b[i];Re_b[i];St[i])
{Nus_b[i] =20}

Call r_chw (Th_B;T_W[i];T_int_ch[i];L_CV[i];W_ch:R_Ax_ch_w[i])

Call r_chsw (W_ch;Th_SW;T_int_ch[i];T_int_sw[i];L_CV[i];Th_B:R_ax_ch_sw[i];tau_diff[i])

Call r_convective (Nus_b[i];P_sCO2;T_F[i];D_h;L_CV[i];W_ch:R_conv[i];h_conv[i])
 Call bl_cond (delta_cl[i];T_W[i];L_CV[i];W_ch;P_sCO2:R_cond_bl[i])

Call heat_gen (N_ch;L_B;T_int_se[i];T_int_ch[i];T_int_sw[i];Th_SE;Th_B;W_ch;Th_SW;L_CV[i];I_density:q_gen_se[i];
q_gen_ch[i];q_gen_sw[i])
 Call r_ech (Th_SE;W_ch;T_int_se[i];T_int_ch[i];L_CV[i];Th_B;DELTA_seg:R_ax_E_ch[i])
Call grashof (T_W[i];T_F[i];D_h;P_sCO2;q_flux[i];G_chan;Re_b[i];per;per_heated:Bo_pet[i];Psi_Jackson[i];K_v[i])
 Call velocity (N_ch;W_ch;H_ch;D_h;AR;m_dot_total;T_F[i];T_W[i];P_sCO2;L_ax[i]:V_channel[i];nu_acc[i];
span_spacing[i];y_plus[i];Re_chan[i];F_channel[i];dp_dx[i];dp[i];rho_bulk[i];DELTA_Bankston[i];DELTA_P_W[i];
tau_near_wall[i];u_friction[i];rho_wall[i];delta_cl[i])
 Call wien (T_int_ch[i]:Lambda_max[i];F_rad[i])

End

Duplicate i=1;(N_nodes-1)

Call r_axch (DELTA_Seg;T_int_ch[i];T_int_ch[i+1];Th_B;W_ch:R_ax_ch[i])

Call r_axsw (DELTA_seg;T_int_sw[i];T_int_sw[i+1];Th_SW;Th_B:R_ax_sw[i])
```

```
Call r_ee (DELTA_seg;T_int_se[i];T_int_se[i+1];Th_SE;Th_B:R_ax_EE[i])

End

"Max amd Min to show results"
Bo_pet_max = max(Bo_pet[1..40])
Bo_pet_min = min(Bo_pet[1..40])
K_v_max = max(K_v[1..40])
K_v_min = min(K_v[1..40])


"-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------"
"Governing equations"

"Energy balance for the nodes at the edge"

"The first node"
"SE"
C_int_SE[1]*dT_int_SEdtau[1] = q_gen_se[1] - ((T_int_se[1]-T_int_ch[1])/R_ax_E_ch[1] + (T_int_SE[1]-T_int_SE[2])/
R_ax_EE[1])
DELTAT_int_SE[1]=integral(dT_int_SEdtau[1];tau;tau_1;tau_2; dtau)
DELTAT_int_SE[1]=T_int_se[1]-T_int_se_i[1]



 "CH"
C_int_ch[1]*dT_int_chdtau[1] =  (T_int_se[1]-T_int_ch[1])/R_ax_E_ch[1]+ q_gen_ch[1] - ((T_int_ch[1]-T_W[1])/
R_ax_ch_w[1] + (T_int_ch[1]-T_int_ch[2])/R_ax_ch[1] + (T_int_ch[1]-T_int_sw[1])/R_Ax_ch_sw[1])
DELTAT_int_ch[1]=integral(dT_int_chdtau[1];tau;tau_1;tau_2; dtau)
DELTAT_int_ch[1]=T_int_ch[1]-T_int_ch_i[1]



 "SW"
C_int_SW[1]*dT_int_SWdtau[1] =  q_gen_sw[1] + (T_int_ch[1]-T_int_sw[1])/R_Ax_ch_sw[1] - (T_int_sw[1]-T_int_sw[2])/
R_ax_sw[1]
DELTAT_int_SW[1]=integral(dT_int_SWdtau[1];tau;tau_1;tau_2; dtau)
DELTAT_int_SW[1]=T_int_sw[1]-T_int_sw_i[1]


(T_int_ch[1]-T_W[1])/R_ax_ch_w[1]  = (T_W[1]-T_F[1])/R_conv[1] "Wall"

q_conv[1] =  (T_W[1]-T_F[1])/R_conv[1]

"Fluid"
C_int_F[1]*dT_int_Fdtau[1] = m_dot_chan * h_in +(T_W[1]-T_F[1])/R_conv[1] - m_dot_chan *h[1]
DELTAT_int_F[1]=integral(dT_int_Fdtau[1];tau;tau_1;tau_2; dtau)
DELTAT_int_F[1]=T_F[1]-T_F_i[1]


h_in= enthalpy (CarbonDioxide; P=P_sCO2;T=T_F_in)
h[1]=enthalpy(CarbonDioxide;P=P_sCO2;T=T_F[1])

"The last node"

"SE"
C_int_SE[N_nodes]*dT_int_SEdtau[N_nodes] = q_gen_se[N_nodes]+(T_int_se[N_nodes-1]-T_int_se[N_nodes])/
R_ax_EE[N_nodes-1] - (T_int_se[N_nodes]-T_int_ch[N_nodes])/R_ax_E_ch[N_nodes]
DELTAT_int_SE[N_nodes]=integral(dT_int_SEdtau[N_nodes];tau;tau_1;tau_2; dtau)
DELTAT_int_SE[N_nodes]=T_int_se[N_nodes]-T_int_se_i[N_nodes]



"CH"
C_int_ch[N_nodes]*dT_int_chdtau[N_nodes] =  (T_int_se[N_nodes]-T_int_ch[N_nodes])/R_ax_E_ch[N_nodes]+
q_gen_ch[N_nodes] + (T_int_ch[N_nodes-1]-T_int_ch[N_nodes])/R_ax_ch[N_nodes-1] - ((T_int_ch[N_nodes]-
```

```
T_W[N_nodes])/R_ax_ch_w[N_nodes]+ (T_int_ch[N_nodes]-T_int_sw[N_nodes])/R_Ax_ch_sw[N_nodes])
DELTAT_int_ch[N_nodes]=integral(dT_int_chdtau[N_nodes];tau;tau_1;tau_2; dtau)
DELTAT_int_ch[N_nodes]=T_int_ch[N_nodes]-T_int_ch_i[N_nodes]



"SW"
C_int_SW[N_nodes]*dT_int_SWdtau[N_nodes] =  q_gen_sw[N_nodes] + (T_int_ch[N_nodes]-T_int_sw[N_nodes])/
R_ax_ch_sw[N_nodes] + (T_int_sw[N_nodes-1]-T_int_sw[N_nodes])/R_ax_sw[N_nodes-1]
DELTAT_int_SW[N_nodes]=integral(dT_int_SWdtau[N_nodes];tau;tau_1;tau_2; dtau)
DELTAT_int_SW[N_nodes]=T_int_sw[N_nodes]-T_int_sw_i[N_nodes]


(T_int_ch[N_nodes]-T_W[N_nodes])/R_ax_ch_w[N_nodes] = (T_W[N_nodes]-T_F[N_nodes])/R_conv [N_nodes] "Wall"

q_conv[N_nodes] =  (T_W[N_nodes]-T_F[N_nodes])/R_conv[N_nodes]

"Fluid"
C_int_F[N_nodes]*dT_int_Fdtau[N_nodes] = m_dot_chan*h[N_nodes-1] + (T_W[N_nodes]-T_F[N_nodes])/
R_conv[N_nodes] - m_dot_chan * h[N_nodes]
DELTAT_int_F[N_nodes]=integral(dT_int_Fdtau[N_nodes];tau;tau_1;tau_2; dtau)
DELTAT_int_F[N_nodes]=T_F[N_nodes]-T_F_i[N_nodes]


h[N_nodes]=enthalpy (CarbonDioxide; P=P_sCO2;T=T_F[N_nodes])




"The internal nodes"

Duplicate i = 2; (N_nodes-1)
"SE"
C_int_SE[i]*dT_int_SEdtau[i] =  q_gen_se[i]+(T_int_SE[i-1]-T_int_SE[i])/R_ax_EE[i] - ((T_int_se[i]-T_int_ch[i])/
R_ax_E_ch[i] + (T_int_SE[i]-T_int_SE[i+1])/R_ax_EE[i])
DELTAT_int_SE[i]=integral(dT_int_SEdtau[i];tau;tau_1;tau_2; dtau)
DELTAT_int_SE[i]=T_int_se[i]-T_int_se_i[i]



 "CH"
C_int_ch[i]*dT_int_chdtau[i] =  (T_int_se[i]-T_int_ch[i])/R_ax_E_ch[i]+ q_gen_ch[i] +  (T_int_ch[i-1]-T_int_ch[i])/R_ax_ch[i] -
 ((T_int_ch[i]-T_W[i])/R_ax_ch_w[i] + (T_int_ch[i]-T_int_ch[i+1])/R_ax_ch[i] + (T_int_ch[i]-T_int_sw[i])/R_Ax_ch_sw[i])
DELTAT_int_ch[i]=integral(dT_int_chdtau[i];tau;tau_1;tau_2; dtau)
DELTAT_int_ch[i]=T_int_ch[i]-T_int_ch_i[i]

"SW"
C_int_SW[i]*dT_int_SWdtau[i] =  q_gen_sw[i] + (T_int_ch[i]-T_int_sw[i])/R_Ax_ch_sw[i] + (T_int_sw[i-1]-T_int_sw[i])/
R_ax_sw[i] - (T_int_sw[i]-T_int_sw[i+1])/R_ax_sw[i]
DELTAT_int_SW[i]=integral(dT_int_SWdtau[i];tau;tau_1;tau_2; dtau)
DELTAT_int_SW[i]=T_int_sw[i]-T_int_sw_i[i]


(T_int_ch[i]-T_W[i])/R_ax_ch_w[i]  = (T_W[i]-T_F[i])/R_conv[i] "Wall"

q_conv[i] =  (T_W[i]-T_F[i])/R_conv[i]

"fluid"
C_int_F[i]*dT_int_Fdtau[i] = (m_dot_chan * h[i-1]) +(T_W[i]-T_F[i])/R_conv[i] - (m_dot_chan*h[i])
DELTAT_int_F[i]=integral(dT_int_Fdtau[i];tau;tau_1;tau_2; dtau)
DELTAT_int_F[i]=T_F[i]-T_F_i[i]


h[i]=enthalpy(CarbonDioxide;P=P_sCO2;T=T_F[i])

End

Duplicate i=1; N_nodes
```

```
Q_duty_channel[i]=  (T_int_ch[i]-T_W[i])/R_ax_ch_w[i]
A_flux[i]=W_ch*L_CV[i]
q_flux[i]=q_duty_channel[i]/A_Flux[i]

End


"Check on the model"
Q_in_se =sum(q_gen_se[i];I=1;N_nodes)
Q_in_sw =sum(q_gen_sw[i];I=1;N_nodes)
Q_in_ch =sum(q_gen_ch[i];I=1;N_nodes)
Q_total_in = Q_in_se+Q_in_sw+Q_in_ch
q_conv_tot = sum(q_conv[i];i=1;N_nodes)
Q_out=m_dot_chan*(h[N_nodes]-h_in)



"Estimation of the frictional and accelaration pressure drops in the channel"
DP_total=sum(dp[i];i=1;N_nodes)                                    "Total
frictional pressure drop inside a channel"

rho_in=density(CarbonDioxide; P=P_sCO2; T=T_F_in)                  "Density of
carbon dioxide at the test section inlet"
V_in= m_dot_chan/(rho_in*A_cross_flow)                             "Velocity at
the inlet"
DELTA_acc[1] = rho_bulk[1]*V_channel[1]^2 - rho_in*V_in^2
DELTA_Acc[1]=rho_in[1]*V_channel[1]^2-rho_in*V_in^2
Duplicate i=2;N_nodes
DELTA_Acc[i]= rho_bulk[i]*V_channel[i]^2-rho_bulk[i-1]*V_channel[i-1]^2

End
"Flow acceleration parameter -- Bankston 1970 "
Duplicate i=1;N_nodes-1
K[i]= (nu_acc[i]/V_channel[i]^2)* ((V_channel[i+1]-V_channel[i])/L_CV[i])
{Acc_st[i]= K[i]/St[i]}
End
Duplicate i=1;N_nodes
DELTA_acc_m[i]=DELTA_Acc[i]/L_CV[i]
DELTA_patel[i]= -nu_acc[i]/(rho_wall[i]*u_friction[i]^3)*(dp_dx[i] + DELTA_acc_m[i])
DELTA_tot[i]= dp[i]+Delta_Acc[i]
End
h_conv_mean = average(h_conv[1..40])
h_conv_max = max(h_conv[1..40])
T_W_mean = average(T_W[1..40])
T_W_max = max(T_W[1..40])


"Integration in time"
$IntegralTable tau:dtau  I_curr; G_chan; q_flux[20]; T_F[1]; T_F[10]; T_F[20]; T_F[30]; T_F[N_nodes]; T_int_SE[1];
T_int_SE[10]; T_int_SE[20]; T_int_SE[30]; T_int_SE[N_nodes]; T_int_ch[1]; T_int_ch[10]; T_int_ch[20]; T_int_ch[30];
T_int_ch[N_nodes]; T_int_SW[1]; T_int_SW[10]; T_int_SW[20]; T_int_SW[30]; T_int_SW[N_nodes]; T_W[1]; T_W[10];
T_W[20]; T_W[30]; T_W[N_nodes]; Bo_pet_max; Bo_pet_min; K_v_max; K_v_min; h_conv_mean; T_W_mean;
T_W[1..40]; h_conv[1..40]
```