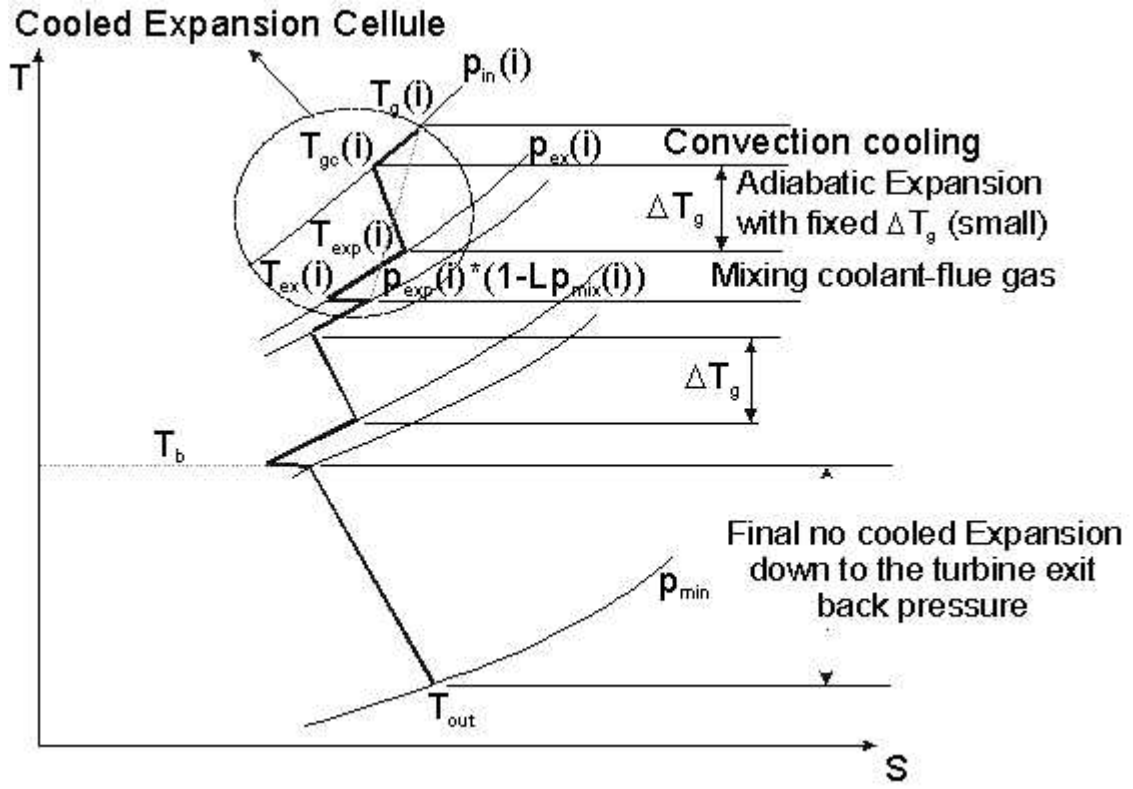
APPLICATIVO SOFTWARE PER IL CALCOLO TERMODINAMICO DEL CICLO JOULE CON MODELLO CELLULARE DI ESPANSIONE REFRIGERATA



Firenze, 25 Settembre 2003

INTRODUZIONE

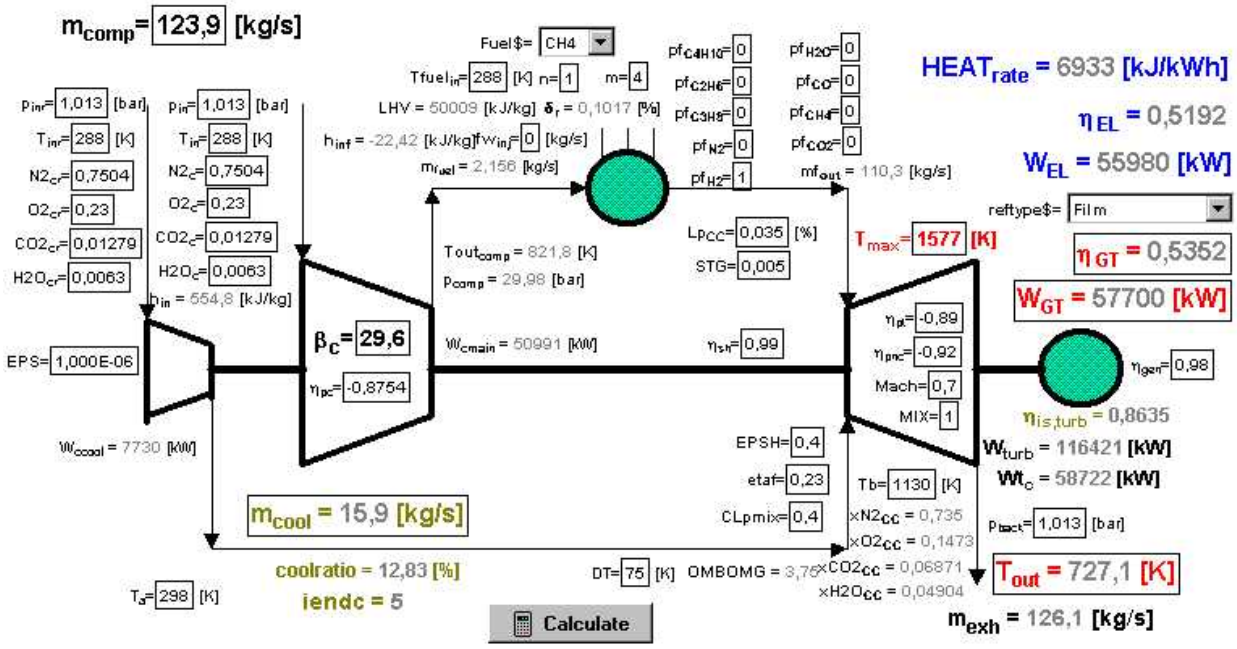
Il lavoro svolto riguarda il modello cellulare di espansione refrigerata, sviluppato da Daniele Fiaschi, tramite l'applicativo EES. Sostanzialmente il programma è rimasto invariato, a parte la possibilità di inserire direttamente dal diagramma della macchina alcune variabili come la temperatura di riferimento, il tipo di combustibile e il tipo di raffreddamento delle pale. E' stato tolto la possibilità di impostare sia DT di espansione, sia OMBOMG, avendo impostato OMBOMG, come da biografia Facchini, pari al 5% del DT. Più importante è l'introduzione dei differenti livelli di pressione di spillamento dal compressore del refrigerante, con relative temperature, anziché l'unico livello di spillamento, pari alla pressione di mandata del compressore. Sono state inoltre staccate tutte le routines che il programma utilizza per effettuare le operazioni elementari di compressione, espansione non refrigerata, calcolo del cp medio, ecc. in modo da non essere obbligati a riscriverle in qualsiasi altro programma, ma richiamandole semplicemente dalle librerie.

INDICE

1 Esempio con GE LM 6000

Le librerie in cui è stato suddiviso il programma principale sono:

- COMPRproc
- RefCELL
- TURBproc
- Fuelmix
- Cpmedioproc



"Simbols explanation"

"Inlet data"

"Fuel$=name of combustible"
"m=Moles of carbon into the generic CnHm hydrocarbon fuel"
"n=Moles of hydrogen into the generic CnHm hydrocarbon fuel"
"m_comp = Inlet comburent mass flow rate (air+eccess air or stechiometric air+other gas)"
"fw_inj=water injecterd after compression"
"N2_c=Inlet N2 mass fraction"
"O2_c=Inlet O2 mass fraction"
"CO2_c=Inlet CO2 mass fraction"
"H2O_c=Inlet H2O mass fraction"
"p_in=compressor suction pressure"
"p_inr=coolant compressor suction pressure"

"beta_c=compressor pressure ratio"
"P_back=counterpressure out of turbine"
"N2_cr=refrigerant Inlet N2 mass fraction"
"O2_cr=refrigerant Inlet O2 mass fraction"
"CO2_cr=refrigerant Inlet CO2 mass fraction"
"H2O_cr=refrigerant Inlet H2O mass fraction"
"T_in=Comburent Inlet temperature"
"T_inr=Coolant Inlet temperature"
"Tfuel_in=Fuel Inlet temperature"
"T_max=Products outlet temperature"
"pf_CH4= CH4 inlet mass fraction"
"pf_C2H6=C2H6 inlet mass fraction"
"pf_C3H8=C3H8 inlet mass fraction"
"pf_C4H10=C4H10 inlet mass fraction"
"pf_CO=CO inlet mass fraction"
"pf_H2=H2 inlet mass fraction"
"pf_N2=N2 inlet mass fraction"
"pf_H2O=H2O inlet mass fraction"
"pf_CO2=CO2 inlet mass fraction"
"T_a=ambient temperature"
"MACH=turbine mach number"
"Clpmix=mixture losses in turbine"
"STG=stanton number of gas"
"EPSH=cooler yield"
"ETAF=film cooling yield"
"Tb=maximum blade surface temperature"
"Dt=difference temerature in cooling refrigeration"
"eta_pnc=politropic yield of uncooled turbine"
"eta_pt=politropic yield of cooled turbine"
"LP_CC=losses in combustion chamber"
"eta_pc=compressor politropic/isentropic yield"
"EPS=relative error"
"reftype$=heat exchange method: traspirazione, film cooling, convezione"
"MIX=flows mixture"

"Outlet data"

p_comp=compressor discharge pressure"
"W_turb=total turbine work"
"Wt_c=total compressor work"
"W_ccool=cooler compressor work"
"W_cmain=compressor work"
"m_fuel=inlet fuel mass flow"
"LHV=low heating value"
"xN2_CC=Outlet N2 mass fraction"
"xO2_CC=Outlet O2 mass fraction"
"xCO2_CC=Outlet CO2 mass fraction"
"xH2O_CC=Outlet H2O mass fraction"
"OMBOMG=blade surface on gas passage ratio"
"Delta_r=mass flow percentage error"
"mf_out=outlet gas mass flow"

"m_cool=cooler mass flow"
"T_out=turbine discharge temperature"
"Tout_comp=compressor discharge temperature"
mf_out=outlet mass flow rate"
"pexc=outlet cooling turbine mass flow pressure"
"iendc=film cooling index"
"m_c[j]=cooler mass flow per step"
"Wtc[j]=cooler work per step"
"pex[i]=pressure per step"
"Tg[i]=inlet stage gas flow temperature"

"Other data"

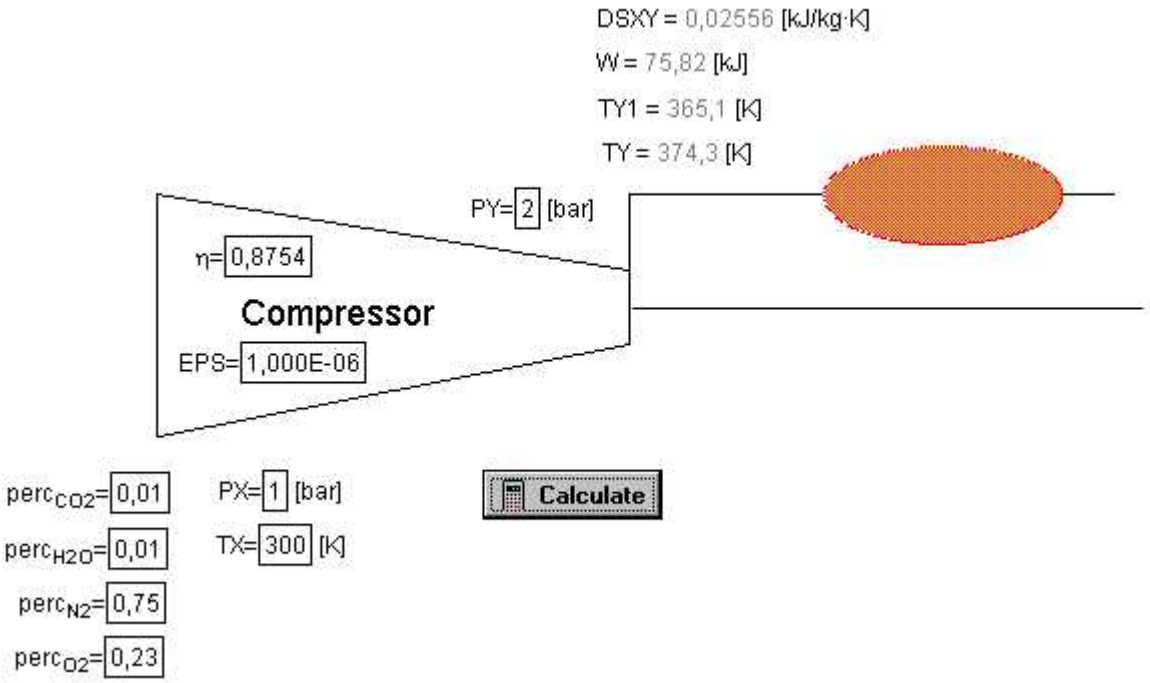
"h_inf=inlet fuel entalpy"
"h_in=inlet comburent mixture entalpy"

Il programma principale, è pertanto rimasto il seguente:

```
"MAIN PROGRAM"
"Calcolo uscita dalla turbina refrigerata + camera di combustione"
call
refCELL(m_comp;fw_inj;beta_c;p_back;O2_c;N2_c;CO2_c;H2O_c;O2_cr;N2_cr;CO2_cr;H2O_cr;p_in
;p_inr;T_in;T_inr;Tfuel_in;Fuel$;T_max;Lp_CC;eta_pt;eta_pc;eta_pnc;Mach;CLpmix;STG;EPSH;etaf;
Tb;DT;EPS;T_a;m;n;pf_CH4;pf_C2H6;pf_C3H8;pf_C4H10;pf_CO;pf_H2;pf_N2;pf_H2O;pf_CO2;reftyp
e$;MIX:W_turb;W_cmain;W_ccool;Wt_c;m_fuel;LHV;m_cool;T_out;Tout_comp;p_comp;pexc;iendc;mf
_out;h_inf;h_in;xN2_CC;xO2_CC;xCO2_CC;xH2O_CC;Delta_r;OMBOMG)
"Calcolo uscita teorica con turbina non refrigerata e rendimento unitario"
call
refCELL(m_comp;fw_inj;beta_c;p_back;O2_c;N2_c;CO2_c;H2O_c;O2_cr;N2_cr;CO2_cr;H2O_cr;p_in
;p_inr;T_in;T_inr;Tfuel_in;Fuel$;T_max;Lp_CC;1;eta_pc;1;Mach;CLpmix;STG;EPSH;etaf;T_max+1;DT
;EPS;T_a;m;n;pf_CH4;pf_C2H6;pf_C3H8;pf_C4H10;pf_CO;pf_H2;pf_N2;pf_H2O;pf_CO2;reftype$;MI
X:Wtt;W_cmaint;Wtcrlt;W_ccoolt;m_fuelt;LHVt;m_coolt;Toutt;Tout_compt;p_compt;pexct;iendct;mf_ou
tt;h_inft;h_int;xN2_CCt;xO2_CCt;xCO2_CCt;xH2O_CCt;Delta_rt;OMBOMGt)
eta_is_turb=W_turb/Wtt
"Calcolo prestazioni"
W_GT=W_turb-Wt_c
eta_GT=W_GT/(m_fuel*LHV)
W_sp=W_GT/m_comp
W_sp_turb=W_turb/(m_comp+m_cool+m_fuel)
coolratio=m_cool/m_comp*100
W_EL=W_GT*eta_sh*eta_gen
eta_EL=W_EL/(m_fuel*LHV)
m_exh=m_comp+m_fuel
HEAT_rate=m_fuel*3600*LHV/W_EL   {[kJ/kWh]}
```

1.1. COMPRproc

La procedura, note la composizione del gas in ingresso e le sue condizioni termodinamiche, il rendimento isentropico o politropico del compressore e la pressione in uscita, calcola il lavoro specifico di compressione e la temperatura isentropica o politropica in uscita.



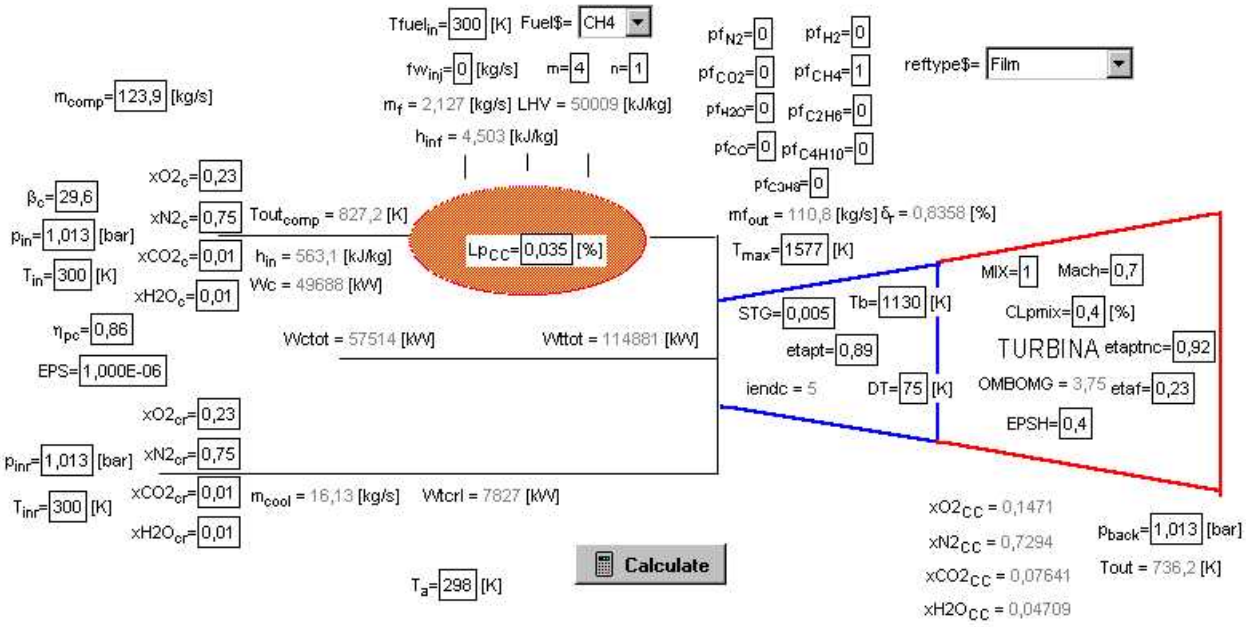
1.1.1. Programma COMPRproc

```
"Procedure for the compression calculation"

"Simbols explanation"
"Inlet data"

"perc_N2=Inlet N2 mass fraction"
"perc_O2=Inlet O2 mass fraction"
"perc_CO2=Inlet CO2 mass fraction"
"perc_H2O=Inlet H2O mass fraction"
"TX=mixture Inlet temperature"
"PX=mixture Inlet pressure"
"ETA=compressor politropic/isentropic yield"
"EPS=relative difference of K"
```

```
"Outlet data"

"TY=isentropic mixture outlet temperature"
"TY1=politropic mixture outlet temperature"
"DSXY=entropy difference"
"W=work of the compressor"

PROCEDURE
COMPRproc(perc_N2;perc_O2;perc_CO2;perc_H2O;PX;TX;PY;ETA;EPS:TY;TY1;DSXY;W)

"Procedure per il calcolo di una compressione note la pressione e la temperatura iniziali, la pressione
finale e il rendimento (isoentropico o politropico) di compressione"

"CALCOLO COMPRESSIONE ISOENTROPICA"

K=1,4
300: TY1=TX*(PY/PX)^((K-1)/K)
CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TX;TY1:cpXY1;cvXY1;k1)
IF(ABS((k1-K)/K)<EPS) THEN
      K:=k1

"CALCOLO RENDIMENTO ADIABATICO"

      IF(ETA>0) THEN
            ETA1:=ETA
      ELSE
            ETA1:=(1-(PY/PX)^(ABS(ETA)*(K-1)/K))/(1-(PY/PX)^((K-1)/K))
      ENDIF

"CALCOLO PUNTO DI FINE COMPRESSIONE REALE"

      TY:=TY1
400:
      CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TX;TY:cpXY;cvXY;kXY)
      TYY=CPXY1*(TY1-TX)/(CPXY*ETA1)+TX
      IF(ABS((TYY-TY)/TY)<EPS) THEN
            TY:=TYY
            CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TX;TY:cpXY;cvXY;kXY)
            CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TY1;TY:cpXY1;cvXY1;kXY1)
            DSXY:=cpXY1*ln(TY/TY1)
            W:=CPXY*(TY-TX)
      ELSE
            TY:=TYY
            goto 400
      ENDIF
ELSE
      K:=k1
      GOTO 300
ENDIF

END PROCEDURE
```

1.2. RefCELL

I dati di input di questa procedura sono: tipo combustibile, temperatura ed eventuale iniezione d'acqua presente, perdite di pressione in camera di combustione; composizione, massa e condizioni termodinamiche del gas sia in ingresso al compressore del refrigerante, che dell'aria comburente, oltre al rendimento isentropico o politropico del compressore e suo rapporto di compressione. Per l'espansione occorre sapere: rendimento isentropico o politropico della turbina refrigerata e non, temperatura massima di ciclo, temperatura massima della pala, tipo di refrigerazione, Stanton del gas, Mach della turbina, contropressione allo scarico, salto di temperatura di refrigerazione. In output: massa di refrigerante, combustibile e dei gas di scarico, con composizione; stadi refrigerati, potenza del compressore dell'aria comburente, del refrigerante e della turbina, OMBONG, potere calorifico inferiore del combustibile.



1.2.1. Programma RefCELL

```
"Simbols explanation"
"Inlet data"

"Fuel$=name of combustible"
"m=Moles of carbon into the generic CnHm hydrocarbon fuel"
"n=Moles of hydrogen into the generic CnHm hydrocarbon fuel"
"mcc_in = Inlet comburent mass flow rate (air+eccess air or stechiometric air+other gas)"
"fw_inj=water injecterd after compression"
```

"xN2_in=Inlet N2 mass fraction"
"xO2_in=Inlet O2 mass fraction"
"xCO2_in=Inlet CO2 mass fraction"
"xH2O_in=Inlet H2O mass fraction"
"xN2_mixw=Inlet N2 mass fraction after injection"
"xO2_mixw=Inlet O2 mass fraction after injection"
"xCO2_mixw=Inlet CO2 mass fraction after injection"
"xH2O_mixw=Inlet H2O mass fraction after injection"
"PY=compressor discharge pressure"
"TY=compressor discharge temperature"
"xN2_inc=refrigerant Inlet N2 mass fraction"
"xO2_inc=refrigerant Inlet O2 mass fraction"
"xCO2_inc=refrigerant Inlet CO2 mass fraction"
"xH2O_inc=refrigerant Inlet H2O mass fraction"
"TCC_in=Comburent Inlet temperature"
"Tfuel_in=Fuel Inlet temperature"
"T_max=Products outlet temperature"
"pf_CH4= CH4 inlet mass fraction"
"pf_C2H6=C2H6 inlet mass fraction"
"pf_C3H8=C3H8 inlet mass fraction"
"pf_C4H10=C4H10 inlet mass fraction"
"pf_CO=CO inlet mass fraction"
"pf_H2=H2 inlet mass fraction"
"pf_N2=N2 inlet mass fraction"
"pf_H2O=H2O inlet mass fraction"
"pf_CO2=CO2 inlet mass fraction"
"T_a=ambient temperature"
"MACH=turbine mach number"
"Clpmix=mixture losses in turbine"
"STG=stanto number of gas"
"omong=blade surface on gas passage ratio"
"EPSH=cooler yield"
"ETAF=film cooling yield"
"Tb=maximum blade surface temperature"
"Dt=difference temerature in cooling refrigeration"
"etaptnc=politropic yield of uncooled turbine"
"etapt=politropic yield of cooled turbine"
"P_back=counterpressure out of turbine"
"LP_CC=losses in combustion chamber"
"ETA=turbine politropic/isentropic yield"
"EPS=relative error"
"reftype$=heat exchange method: traspirazione, film cooling, convezione"
"MIX=flows mixture"

"Outlet data"

"Wttot=total turbine work"
"Wtc=cooled turbine work"
"Wtnc=uncooled turbine work"
"Wcrl=cooler compressor work"
"Wc=compressor work"

```
"m_f=inlet fuel mass flow"
"LHV=low heating value"
"xN2_CC=Outlet N2 mass fraction"
"xO2_CC=Outlet O2 mass fraction"
"xCO2_CC=Outlet CO2 mass fraction"
"xH2O_CC=Outlet H2O mass fraction"
"Delta_r=mass flow percentage error"
"mf_out=outlet gas mass flow"
"M_cool=cooler mass flow"
"Tout=turbine discharge temperature"
"pexc=outlet cooling turbine mass flow pressure"
"iendc=film cooling index"
"m_c[j]=cooler mass flow per step"
"Wtc[j]=cooler work per step"
"pex[i]=pressure per step"
"Tg[i]=inlet stage gas flow temperature"

"Other data"

"h_inf=inlet fuel entalpy"
"h_in=inlet comburent mixture entalpy"

PROCEDURE
refCELL(m_comp;fw_inj;beta_c;p_back;xO2_c;xN2_c;xCO2_c;xH2O_c;xO2_cr;xN2_cr;xCO2_cr;xH2
O_cr;p_in;p_inr;T_in;T_inr;Tfuel_in;Fuel$;T_max;Lp_CC;etapt;eta_pc;etaptnc;Mach;CLpmix;STG;EPS
H;etaf;Tb;DT;EPS;T_a;m;n;pf_CH4;pf_C2H6;pf_C3H8;pf_C4H10;pf_CO;pf_H2;pf_N2;pf_H2O;pf_CO
2;reftype$;MIX:Wttot;Wc;Wtcrl;Wctot;m_f;LHV;m_cool;Tout;Tout_comp;p_comp;pexc;iendc;mf_out;h_i
nf;h_in;xN2_CC;xO2_CC;xCO2_CC;xH2O_CC;Delta_r;OMBOMG)
"Dati in ingresso al compressore"

p_comp=p_in*beta_c

"Calcolo uscita dal compressore principale"

call
COMPRproc(xN2_c;xO2_c;xCO2_c;xH2O_c;p_in;T_in;p_comp;eta_pc;EPS:Tout_comp;Tout_compis;
DSXY_comp;W)

"Calcolo composizione dopo eventuale iniezione d'acqua o vapore"
xO2_mixw=xO2_c*m_comp/(m_comp+m_comp*fw_inj)
xN2_mixw=xN2_c*m_comp/(m_comp+m_comp*fw_inj)
xCO2_mixw=xCO2_c*m_comp/(m_comp+m_comp*fw_inj)
xH2O_mixw=xH2O_c*m_comp/(m_comp+m_comp*fw_inj)

OMBOMG=DT*0,05
s= abs(floor(-(T_max-Tb)/DT))
j:=0
repeat
      j:=j+1;
      m_c[j]=0
      Wtc[j]=0
      Wcrl[j]=0
```

```
until (j=s)
m_cool=0

9527: mCC_in=m_comp*(1+fw_inj)-m_cool

Call
fuelmix(mCC_in;xN2_mixw;xO2_mixw;xCO2_mixw;xH2O_mixw;T_a;Tout_comp;Tfuel_in;T_max;Fuel
$;m;n;pf_CH4;pf_C2H6;pf_C3H8;pf_C4H10;pf_CO;pf_H2;pf_N2;pf_H2O;pf_CO2:m_f;mf_out;h_inf;h_i
n;LHV;xN2_CC;xO2_CC;xCO2_CC;xH2O_CC;Delta_r)

mgas[1]=mCC_in+m_f
m_cool1=0
pin[1]=p_comp*(1-Lp_CC)
Tg[1]=T_max
i=1
REPEAT      "do 20, i=1,(2*s)"
"Determinazione temperatura di mandata refrigerante dal compressore alla pin richiesta"

IF (Tg[i]>Tb) THEN
      "Calcolo uscita dal compressore del refrigerante"
      beta_cr[i]=pin[i]/p_inr
      call
      COMPRproc(xN2_cr;xO2_cr;xCO2_cr;xH2O_cr;p_inr;T_inr;pin[i];eta_pc;EPS:Tc[i];Tout_compris[i
];DSXY_compr[i];Wcr[i])

      {Tratto espansione refrigerata senza produzione di lavoro. Cooling modellato con tratto piccolo
convezione-espansione-miscelazione. Refrigerazione per convezione interna}

      phi=(Tg[i]-Tb)/(Tg[i]-Tc[i])
      if(reftype$='convezione') then
            DELTA_Tgc[i]=Stg*ombomg*(Tg[i]-Tb)
            Tgc[i]=Tg[i]-DELTA_Tgc[i]
            DELTA_Tc[i]=(Tb-Tc[i])*EPSH
            Tcconv=Tc[i]+DELTA_Tc[i]
            if (i=1) then
                  call
cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tgc[i];Tg[i]:cp_g;cv_g;k_g)
            else
                  call
cpmedioproc(xO2_gas;xN2_gas;xCO2_gas;xH2O_gas;Tgc[i];Tg[i]:cp_g;cv_g;k_g)
            endif
            CALL cpmedioproc(xO2_cr;xN2_cr;xCO2_cr;xH2O_cr;Tc[i];Tcconv:cp_c;cv_c;k_c)
            m_c[i]=mgas[i]*cp_g/cp_c*Stg*ombomg*1/EPSH*phi/(1-phi)
      endif
      if(reftype$='film') then
            DELTA_Tgc[i]=Stg*ombomg*(Tg[i]-Tb)
            Tgc[i]=Tg[i]-DELTA_Tgc[i]
            DELTA_Tc[i]=(Tb-Tc[i])*EPSH/(1-etaf)
            Tcconv=Tc[i]+DELTA_Tc[i]
            if (i=1) then
                  call
```

```
 cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tgc[i];Tg[i]:cp_g;cv_g;k_g)
                else
                        call
cpmedioproc(xO2_gas;xN2_gas;xCO2_gas;xH2O_gas;Tgc[i];Tg[i]:cp_g;cv_g;k_g)
                endif
                CALL cpmedioproc(xO2_cr;xN2_cr;xCO2_cr;xH2O_cr;Tc[i];Tcconv:cp_c;cv_c;k_c)
                m_c[i]=mgas[i]*cp_g/cp_c*Stg*ombomg*1/EPSH*phi/(1-phi)*(1-etaf)
        endif
        if(reftype$='traspirazione') then
                if (i=1) then
                        call
cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tgc[i];Tg[i]:cp_g;cv_g;k_g)
                else
                        call
cpmedioproc(xO2_gas;xN2_gas;xCO2_gas;xH2O_gas;Tgc[i];Tg[i]:cp_g;cv_g;k_g)
                endif
                CALL cpmedioproc(xO2_cr;xN2_cr;xCO2_cr;xH2O_cr;Tc[i];Tcconv:cp_c;cv_c;k_c)
                deltaStg1=Stg*0,1
101: m_c[i]=mgas[i]*cp_g/cp_c*(Stg-deltaStg1)*ombomg*phi/(1-phi)
                deltaStg=coeftrasp*m_c[i]/mgas[i]*1/ombomg
                if((abs((deltaStg-deltaStg1)/deltaStg))<(EPS)) then
                        goto 100
                else
                        deltaStg1=deltaStg
                        goto 101
                endif
                "abbassamento T per cooling ISOBARO i-esimo nel caso di traspirazione"
100: DELTA_Tgc[i]=(Stg-deltaStg)*ombomg*(Tg[i]-Tb)
        endif
        "Tratto espansione con DELTA_Te fissato"
        Tesp[i]=Tgc[i]-DT
        if (i=1) then
                call cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tesp[i];Tgc[i]:cpm_g;cvm_g;kg)
        else
                call
cpmedioproc(xO2_gas;xN2_gas;xCO2_gas;xH2O_gas;Tesp[i];Tgc[i]:cpm_g;cvm_g;kg)
        endif
        kg=cpm_g/cvm_g
        if (etapt>0) then
                Tespis=Tgc[i]-(Tgc[i]-Tesp[i])/ETApt
                if (i=1) then
                        call
cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tespis;Tgc[i]:cpms_g;cvms_g;kgs)
                else
                        call
cpmedioproc(xO2_gas;xN2_gas;xCO2_gas;xH2O_gas;Tespis;Tgc[i]:cpms_g;cvms_g;kgs)
                endif
                pex[i]=pin[i]*(Tespis/Tgc[i])^(kgs/(kgs-1))
                goto 30
        endif

        ETAtis1=abs(etapt)
```

```
10: Tespis=Tgc[i]-(Tgc[i]-Tesp[i])/ETAtis1
    if (i=1) then
        call
cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tespis;Tgc[i]:cpms_g;cvms_g;kgs)
    else
        call
cpmedioproc(xO2_gas;xN2_gas;xCO2_gas;xH2O_gas;Tespis;Tgc[i]:cpms_g;cvms_g;kgs)
    endif
    {Calcolo p uscita i-esima con espansione isoentropica}
    pex[i]=pin[i]*(Tespis/Tgc[i])^(kgs/(kgs-1))
    ETAtis[i]=(1-(pex[i]/pin[i])^(abs(ETApt)*(Kgs-1)/Kgs))/(1-(pex[i]/pin[i])^((Kgs-1)/Kgs))
    if((ABS((etatis[i]-etatis1)/etatis[i]))<(EPS)) then
        goto 30
    else
        etatis1=etatis[i]
        goto 10
    endif
30:
    {controllo se c'è miscelazione}
    if(MIX=0) then
        Tex[i]=Tesp[i]
        Wtc[i]=mgas[i]*cpm_g*(Tgc[i]-Tesp[i])
        mgex[i]=mgas[i]


    cp_gmix=xO2_CC*SPECHEAT(O2;T=Tesp[i])+xN2_CC*SPECHEAT(N2;T=Tesp[i])+xCO2_CC*S
PECHEAT(CO2;T=Tesp[i])+xH2O_CC*SPECHEAT(H2O;T=Tesp[i])
        xO2_gas=xO2_CC
        xN2_gas=xN2_CC
        xCO2_gas=xCO2_CC
        xH2O_gas=xH2O_CC
        xO2_mix=xO2_CC
        xN2_mix=xN2_CC
        xCO2_mix=xCO2_CC
        xH2O_mix=xH2O_CC
        goto 60
    endif

    Wtc[i]=mgas[i]*cpm_g*(Tgc[i]-Tesp[i])
    Wcrl[i]=Wcr[i]*m_c[i]
    {Miscelamento tra refrigerante e corrente principale}
    mgex[i]=mgas[i]+m_c[i]
    IF(i=1) THEN
        xO2_mix=(xO2_CC*mgas[1]+xO2_cr*m_c[1])/(mgas[1]+m_c[1])
        xN2_mix=(xN2_CC*mgas[1]+xN2_cr*m_c[1])/(mgas[1]+m_c[1])
        xCO2_mix=(xCO2_CC*mgas[1]+xCO2_cr*m_c[1])/(mgas[1]+m_c[1])
        xH2O_mix=(xH2O_CC*mgas[1]+xH2O_cr*m_c[1])/(mgas[1]+m_c[1])
    ELSE
        xO2_mix=(xO2_gas*mgas[1]+xO2_cr*m_c[i])/(mgas[1]+m_c[i])
        xN2_mix=(xN2_gas*mgas[1]+xN2_cr*m_c[i])/(mgas[1]+m_c[i])
        xCO2_mix=(xCO2_gas*mgas[1]+xCO2_cr*m_c[i])/(mgas[1]+m_c[i])
        xH2O_mix=(xH2O_gas*mgas[1]+xH2O_cr*m_c[i])/(mgas[1]+m_c[i])
```

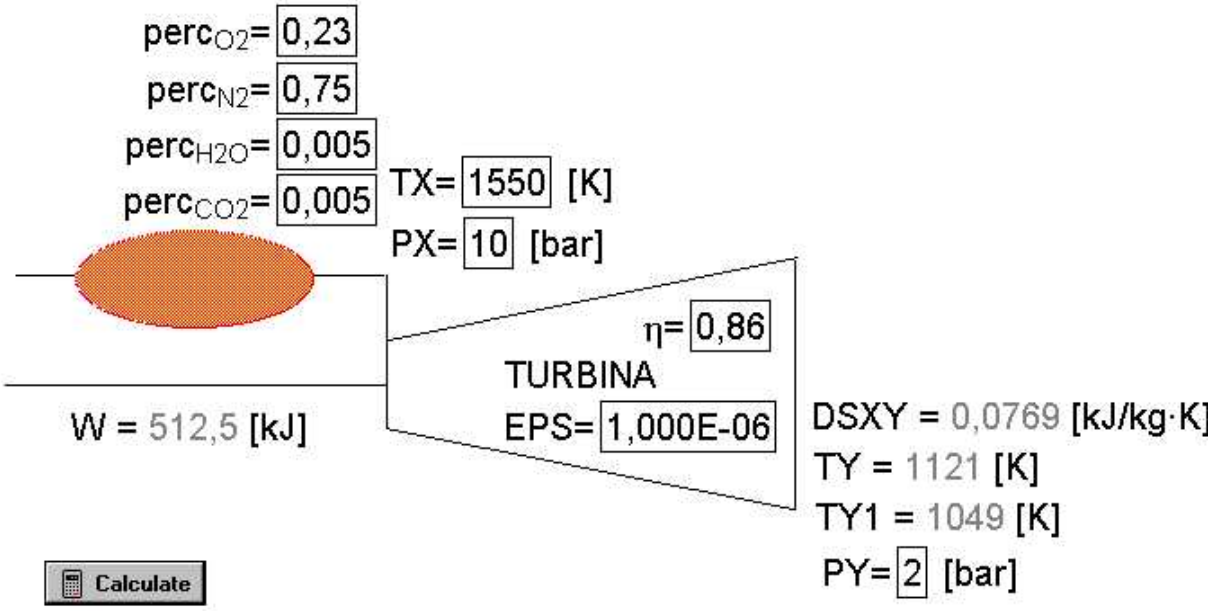
ENDIF

```
      cp_g=xO2_mix*SPECHEAT(O2;T=Tesp[i])+xN2_mix*SPECHEAT(N2;T=Tesp[i])+xCO2_mix*SP
ECHEAT(CO2;T=Tesp[i])+xH2O_mix*SPECHEAT(H2O;T=Tesp[i])

      Tex1=Tesp[i]
61:
      call cpmedioproc(xO2_mix;xN2_mix;xCO2_mix;xH2O_mix;Tex1;Tesp[i]:cp_g1;cv_g1g;k_g1)
      call cpmedioproc(xO2_cr;xN2_cr;xCO2_cr;xH2O_cr;Tcconv;Tex1:cp_c1;cv_c1g;k_c1)
      Tex[i]=(m_c[i]*cp_c1*Tcconv+mgas[i]*cp_g1*Tesp[i])/(m_c[i]*cp_c1+mgas[i]*cp_g1)
      if((abs(Tex[i]-Tex1)/Tex[i])<=EPS) then
            xO2_gas=xO2_mix
            xN2_gas=xN2_mix
            xCO2_gas=xCO2_mix
            xH2O_gas=xH2O_mix


      cp_gmix=xO2_gas*SPECHEAT(O2;T=Tex[i])+xN2_gas*SPECHEAT(N2;T=Tex[i])+xCO2_gas*SP
ECHEAT(CO2;T=Tex[i])+xH2O_gas*SPECHEAT(H2O;T=Tex[i])


      nmolmix=xO2_gas/MOLARMASS(O2)+xN2_gas/MOLARMASS(N2)+xCO2_gas/MOLARMASS(C
O2)+xH2O_gas/MOLARMASS(H2O)
            fmolO2mix=xO2_gas/MOLARMASS(O2)/nmolmix
            fmolN2mix=xN2_gas/MOLARMASS(N2)/nmolmix
            fmolCO2mix=xCO2_gas/MOLARMASS(CO2)/nmolmix
            fmolH2Omix=xH2O_gas/MOLARMASS(H2O)/nmolmix


      pmmix=fmolO2mix*MOLARMASS(O2)+fmolN2mix*MOLARMASS(N2)+fmolCO2mix*MOLARMA
SS(CO2)+fmolH2Omix*MOLARMASS(H2O)
            R_gmix=R#/pmmix
            cv_gmix=cp_gmix-R_gmix
            k_gmix=cp_gmix/cv_gmix

            {Perdita di pressione dovuta a mix refrigerante-flusso principale proporzionale al rapporto
m_refrigerante/m_flusso principale (urto) Lpmix=m_c(i)/mgas(i)*CLpmix}

            Lpmix=k_gmix*Mach^2*m_c[i]/mgas[i]*CLpmix    {!Modello di Hodge}
            pex[i]=pex[i]*(1-Lpmix)
            goto 60
      else
            Tex1=Tex[i]
            goto 61 {Ricalcolo Tex dopo mixing}
      endif
60:
      mgas[i+1]=mgex[i]
      Tg[i+1]=Tex[i]
      pin[i+1]=pex[i]
      p_g=cp_gmix
else
      iendc=i-1
      goto 40
endif
```

```
i=i+1

UNTIL (i>=(1000))  "Fine del ciclo REPEAT"

40:
m_cool1=SUM(m_c[i];i=1;s)
if(MIX=1) then
      mgexcool=m_cool1+mgas[1]
else
      mgexcool=mgas[1]
endif
Wtcl=SUM(Wtc[i];i=1;s)
Wtcrl=SUM(Wcrl[i];i=1;s)
IF(i=1) THEN
      "caso espansione completa non refrigerata (Tb > Tg)"
      pexc=pin[1]
      Texcool=Tg[1]
ELSE
      pexc=pex[iendc]
      Texcool=Tex[iendc]
ENDIF
"Parte non refrigerata"
knc1=1,4
81:
IF (i=1) THEN
      call
TURBproc(xN2_CC;xO2_CC;xCO2_CC;xH2O_CC;pexc;Texcool;p_back;ETAptnc;EPS:Tout;Touts;DS
XY;W)
ELSE
      call
TURBproc(xN2_mix;xO2_mix;xCO2_mix;xH2O_mix;pexc;Texcool;p_back;ETAptnc;EPS:Tout;Touts;D
SXY;W)
ENDIF
80:
IF (i=1) THEN
      call cpmedioproc(xO2_CC;xN2_CC;xCO2_CC;xH2O_CC;Tout;Texcool:cpmout;cvmout;knc)
else
      call cpmedioproc(xO2_mix;xN2_mix;xCO2_mix;xH2O_mix;Tout;Texcool:cpmout;cvmout;knc)
endif
Wtnc=mgexcool*cpmout*(Texcool-Tout)
Wttot=Wtcl+Wtnc
IF(ABS((m_cool-m_cool1)/m_cool1)<EPS) THEN
      m_cool:=m_cool1
      GOTO 1122
ELSE
      m_cool:=m_cool1
      GOTO 9527
ENDIF
1122:
Wc=W*(m_comp-m_cool)
Wctot=Wc+Wtcrl
END PROCEDURE
```

1.3. TURBproc

Questa procedura, nota la composizione del gas in ingresso e le sue condizioni termodinamiche, il rendimento isentropico o politropico di espansione la pressione in uscita, calcola il lavoro specifico di espansione e la temperatura isentropica o politropica in uscita.



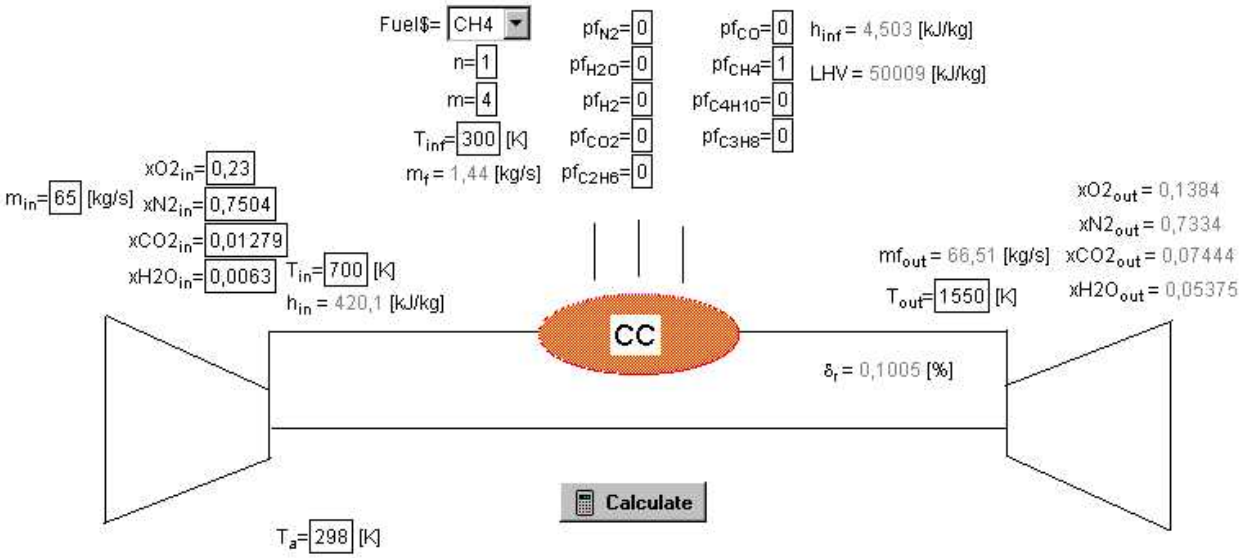
1.3.1. Programma TURBproc

```
"Procedure for the expansion calculation"

"Simbols explanation"
"Inlet data"

"perc_N2=Inlet N2 mass fraction"
"perc_O2=Inlet O2 mass fraction"
"perc_CO2=Inlet CO2 mass fraction"
"perc_H2O=Inlet H2O mass fraction"
"TX=mixture Inlet temperature"
"PX=mixture Inlet pressure"
"PX=mixture outlet pressure"
"ETA=turbine politropic/isentropic yield"
"EPS=relative difference of K"
```

```
"Outlet data"

"TY=isentropic mixture outlet temperature"
"TY1=politropic mixture outlet temperature"
"DSXY=entropy difference"
"W=work of the compressor"

PROCEDURE
TURBproc(perc_N2;perc_O2;perc_CO2;perc_H2O;PX;TX;PY;ETA;EPS:TY;TY1;DSXY;W)

"Procedure per il calcolo di un'espansione note la pressione e la temperatura iniziali, la pressione finale e il rendimento (isoentropico o politropico) di espansione"

"CALCOLO ESPANSIONE ISOENTROPICA"

K=1,4
300: TY1=TX*(PY/PX)^((K-1)/K)
CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TX;TY1:cpXY1;cvXY1;k1)
IF(ABS((K1-K)/K)<EPS) THEN
    K:=K1

    "CALCOLO RENDIMENTO ADIABATICO"

    IF(ETA>0) THEN
        ETA1:=ETA
        ELSE
        ETA1:=(1-(PY/PX)^(ABS(ETA)*(K-1)/K))/(1-(PY/PX)^((K-1)/K))
    ENDIF

    "CALCOLO PUNTO DI FINE ESPANSIONE REALE"

    TY:=TY1
400:
    CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TX;TY:cpXY;cvXY;kXY)
    TYY=TX-ETA1*CPXY1*(TX-TY1)/cpXY
    IF(ABS((TYY-TY)/TY)<EPS) THEN
        TY:=TYY
        CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TX;TY:cpXY;cvXY;kXY)
        CALL cpmedioproc(perc_O2;perc_N2;perc_CO2;perc_H2O;TY1;TY:cpXY1;cvXY1;kXY1)
        DSXY:=cpXY1*ln(TY/TY1)
        W:=CPXY*(TX-TY)
    ELSE
        TY:=TYY
        goto 400
    ENDIF
ELSE
    K:=K1
    GOTO 300
ENDIF

END PROCEDURE
```

1.4. Fuelmix

Questa procedura, nota la massa, la composizione del gas in ingresso e le sue condizioni termodinamiche e noto il tipo di combustibile utilizzato, sia tra quelli conosciuti tra le librerie di EES, sia tra gli idrocarburi selezionabili dalle label e le sue condizioni termodinamiche e nota la temperatura massima di ciclo, determina la massa di combustibile ed il suo potere calorifico inferiore.



1.4.1. Programma Fuelmix

```
{PROCEDURE FOR THE COMPLETE COMBUSTION OF A KNOWN MIXTURE FUEL OF  CH4, C2H6, C3H8, C4H10, CO, H2, N2, H2O, CO2 WITH FIXED OXIDANT MASS FLOW m_in}

"Simbols explanation"
"Inlet data"

"Fuel$=tipo di combustibile"
"m=Moles of carbon into the generic CnHm hydrocarbon fuel"
"n=Moles of hydrogen into the generic CnHm hydrocarbon fuel"
"m_in = Inlet comburent mass flow rate (air+eccess air or stechiometric air+other gas)"
"xN2_in=Inlet N2 mass fraction"
"xO2_in=Inlet O2 mass fraction"
"xCO2_in=Inlet CO2 mass fraction"
"xH2O_in=Inlet H2O mass fraction"
"T_in=Comburent Inlet temperature"
"T_inf=Fuel Inlet temperature"
```

```
"T_out=Products outlet temperature"
"pf_CH4= CH4 inlet mass fraction"
"pf_C2H6=C2H6  inlet mass fraction"
"pf_C3H8=C3H8 inlet mass fraction"
"pf_C4H10=C4H10 inlet mass fraction"
"pf_CO=CO inlet mass fraction"
"pf_H2=H2 inlet mass fraction"
"pf_N2=N2 inlet mass fraction"
"pf_H2O=H2O inlet mass fraction"
"pf_CO2=CO2 inlet mass fraction"
"T_a=ambient temperature"

"Outlet data"

"m_f=inlet fuel mass flow"
"LHV=low heating value"
"xN2_out=Outlet N2 mass fraction"
"xO2_out=Outlet O2 mass fraction"
"xCO2_out=Outlet CO2 mass fraction"
"xH2O_out=Outlet H2O mass fraction"
"Delta_r=mass flow percentage error"
"mf_out=outlet gas mass flow"

"Other data"

"h_inf=inlet fuel entalpy"
"h_in=inlet comburent mixture entalpy"

{Composition given in mass fractions}
procedure
fuelmix(m_in;xN2_in;xO2_in;xCO2_in;xH2O_in;T_a;T_in;T_inf;T_out;Fuel$;m;n;pf_CH4;pf_C2H6;pf_
C3H8;pf_C4H10;pf_CO;pf_H2;pf_N2;pf_H2O;pf_CO2:m_f;mf_out;h_inf;h_in;LHV;xN2_out;xO2_out;x
CO2_out;xH2O_out;Delta_r)

if Fuel$ = 'MIX' then

    "Calculation of components inlet enthalpy"
    h_infCH4=pf_CH4*(enthalpy(CH4;T=T_inf)-enthalpy(CH4;T=T_a))
    h_infC2H6=pf_C2H6*(enthalpy(C2H6;T=T_inf)-enthalpy(C2H6;T=T_a))
    h_infC3H8=pf_C3H8*(enthalpy(C3H8;T=T_inf)-enthalpy(C3H8;T=T_a))
    h_infC4H10=pf_C4H10*(enthalpy(C4H10;T=T_inf)-enthalpy(C4H10;T=T_a))
    h_infCO=pf_CO*(enthalpy(CO;T=T_inf)-enthalpy(CO;T=T_a))
    h_infH2=pf_H2*(enthalpy(H2;T=T_inf)-enthalpy(H2;T=T_a))
    h_infN2=pf_N2*(enthalpy(N2;T=T_inf)-enthalpy(N2;T=T_a))
    h_infH2O=pf_H2O*(enthalpy(H2O;T=T_inf)-enthalpy(H2O;T=T_a))
    h_infCO2=pf_CO2*(enthalpy(CO2;T=T_inf)-enthalpy(CO2;T=T_a))

    h_inf=h_infCH4+h_infC2H6+h_infC3H8+h_infC4H10+h_infCO+h_infH2+h_infN2+h_infH2O
```

h_in=xN2_in*(enthalpy(N2;T=T_in)-enthalpy(N2;T=T_a))+xO2_in*(enthalpy(O2;T=T_in)-enthalpy(O2;T=T_a))+xCO2_in*(enthalpy(CO2;T=T_in)-enthalpy(CO2;T=T_a))+xH2O_in*(enthalpy(H2O;T=T_in)-enthalpy(H2O;T=T_a))

"Evaluation of fuel LHV from its components"

LHV_CH4=enthalpy(CH4;T=T_a)+MOLARMASS(O2)*(1+4/4)/MOLARMASS(CH4)*enthalpy(O2;T=T_a)-MOLARMASS(CO2)*enthalpy(CO2;T=T_a)/MOLARMASS(CH4)-4/2*MOLARMASS(H2O)/MOLARMASS(CH4)*enthalpy(H2O;T=T_a)

LHV_C2H6=enthalpy(C2H6;T=T_a)+MOLARMASS(O2)*(2+6/4)/MOLARMASS(C2H6)*enthalpy(O2;T=T_a)-2*MOLARMASS(CO2)*enthalpy(CO2;T=T_a)/MOLARMASS(C2H6)-6/2*MOLARMASS(H2O)/MOLARMASS(C2H6)*enthalpy(H2O;T=T_a)

LHV_C3H8=enthalpy(C3H8;T=T_a)+MOLARMASS(O2)*(3+8/4)/MOLARMASS(C3H8)*enthalpy(O2;T=T_a)-3*MOLARMASS(CO2)*enthalpy(CO2;T=T_a)/MOLARMASS(C3H8)-8/2*MOLARMASS(H2O)/MOLARMASS(C3H8)*enthalpy(H2O;T=T_a)

LHV_C4H10=enthalpy(C4H10;T=T_a)+MOLARMASS(O2)*(4+10/4)/MOLARMASS(C4H10)*enthalpy(O2;T=T_a)-4*MOLARMASS(CO2)*enthalpy(CO2;T=T_a)/MOLARMASS(C4H10)-10/2*MOLARMASS(H2O)/MOLARMASS(C4H10)*enthalpy(H2O;T=T_a)

LHV_CO=enthalpy(CO;T=T_a)+MOLARMASS(O2)*0,5/MOLARMASS(CO)*enthalpy(O2;T=T_a)-MOLARMASS(CO2)*enthalpy(CO2;T=T_a)/MOLARMASS(CO)

LHV_H2=enthalpy(H2;T=T_a)+MOLARMASS(O2)*2/4/MOLARMASS(H2)*enthalpy(O2;T=T_a)-2/2*MOLARMASS(H2O)/MOLARMASS(H2)*enthalpy(H2O;T=T_a)

LHV=pf_CH4*LHV_CH4+pf_C2H6*LHV_C2H6+pf_C3H8*LHV_C3H8+pf_C4H10*LHV_C4H10+pf_CO*LHV_CO+pf_H2*LHV_H2

"Calculation of species molar fractions in the fuel mixture"

n_tot=pf_CH4/MOLARMASS(CH4)+pf_C2H6/MOLARMASS(C2H6)+pf_C3H8/MOLARMASS(C3H8)+pf_C4H10/MOLARMASS(C4H10)+pf_CO/MOLARMASS(CO)+pf_H2/MOLARMASS(H2)+pf_N2/MOLARMASS(N2)+pf_H2O/MOLARMASS(H2O)+pf_CO2/MOLARMASS(CO2)

fm_CH4=pf_CH4/MOLARMASS(CH4)/n_tot
fm_C2H6=pf_C2H6/MOLARMASS(C2H6)/n_tot
fm_C3H8=pf_C3H8/MOLARMASS(C3H8)/n_tot
fm_C4H10=pf_C4H10/MOLARMASS(C4H10)/n_tot
fm_CO=pf_CO/MOLARMASS(CO)/n_tot
fm_H2=pf_H2/MOLARMASS(H2)/n_tot
fm_N2=pf_N2/MOLARMASS(N2)
fm_H2O=pf_H2O/MOLARMASS(H2O)
fm_CO2=pf_CO2/MOLARMASS(CO2)/n_tot

"Mean fuel molecular mass"

pmol_fuel=fm_CH4*MOLARMASS(CH4)+fm_C2H6*MOLARMASS(C2H6)+fm_C3H8*MOLARMASS(C3H8)+fm_C4H10*MOLARMASS(C4H10)+fm_CO*MOLARMASS(CO)+fm_H2*MOLARMASS(H2)+fm_N2*MOLARMASS(N2)+fm_H2O*MOLARMASS(H2O)+fm_CO2*MOLARMASS(CO2)

"Reference to a reaction of fuel mixture with FIXED amount of comburent air"

"n_f*(fm_CH4+fm_C2H6+fm_C3H8+fm_C4H10+fm_CO+fm_H2+fm_N2+fm_H2O+fm_CO2)+a O2 + b N2 + c H2O + d CO2=x*CO2+y*H2O+z*O2+w*N2"

"Initialization of m_f considering the combustions gas as air"

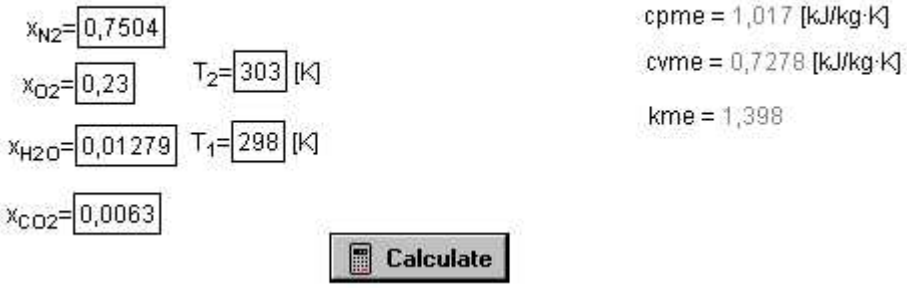
m_f1=m_in*(enthalpy(air;T=T_out)-enthalpy(air;T=T_in))/LHV

```
a=m_in*xO2_in/molarmass(O2)
b=m_in*xN2_in/molarmass(N2)
c=m_in*xCO2_in/molarmass(CO2)
d=m_in*xH2O_in/molarmass(H2O)

10: n_f:=m_f1/pmol_fuel   "N° moles of fuel"

x=n_f*(fm_CH4+2*fm_C2H6+3*fm_C3H8+4*fm_C4H10+fm_CO+fm_CO2)+d     {C balance }
y=n_f*(2*fm_CH4+3*fm_C2H6+4*fm_C3H8+5*fm_C4H10+fm_H2+fm_H2O)+c   {H balance }
z=n_f*(fm_CO/2+fm_H2O/2+fm_CO2)+a+c/2+d-x-y/2                    {O balance}
w=n_f*fm_N2+b                    {N balance}

mO2_out=z*molarmass(O2)
mN2_out=w*molarmass(N2)
mCO2_out=x*molarmass(CO2)
mH2O_out=y*molarmass(H2O)

xO2_out=mO2_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)
xN2_out=mN2_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)
xCO2_out=mCO2_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)
xH2O_out=mH2O_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)

"Enthalpy balance in CC"
h_out=xN2_out*(enthalpy(N2;T=T_out)-enthalpy(N2;T=T_a))+xO2_out*(enthalpy(O2;T=T_out)-
enthalpy(O2;T=T_a))+xCO2_out*(enthalpy(CO2;T=T_out)-
enthalpy(CO2;T=T_a))+xH2O_out*(enthalpy(H2O;T=T_out)-enthalpy(H2O;T=T_a))

m_f=m_in*(h_out-h_in)/(h_inf+LHV-h_out)
if (abs((m_f-m_f1)/m_f)<0,000001) then
        m_f:=m_f1
else
        m_f1:=m_f
        goto 10
endif

mf_out=mO2_out+mN2_out+mCO2_out+mH2O_out
Delta_r=abs((m_f+m_in-mO2_out-mN2_out-mCO2_out-mH2O_out)/(m_f+m_in))*100

endif

if Fuel$ = 'CH4' then

"Reference to a complete combustion reaction of :"
"n_f CnHm + a O2 + b N2 + c H2O + d CO2 = (n_f*n+d) CO2 + (n_f*m/2+c) H2O + (a-n_f*n-
n_f*m/4) O2 + b N2"

h_inf=enthalpy(Fuel$;T=T_inf)-enthalpy(Fuel$;T=T_a)
h_in=xN2_in*(enthalpy(N2;T=T_in)-enthalpy(N2;T=T_a))+xO2_in*(enthalpy(O2;T=T_in)-
enthalpy(O2;T=T_a))+xCO2_in*(enthalpy(CO2;T=T_in)-
enthalpy(CO2;T=T_a))+xH2O_in*(enthalpy(H2O;T=T_in)-enthalpy(H2O;T=T_a))
```

```
LHV=enthalpy(Fuel$;T=T_a)+(n+m/4)*enthalpy(O2;T=T_a)*MOLARMASS(O2)/MOLARMASS(Fu
el$)-n*enthalpy(CO2;T=T_a)*MOLARMASS(CO2)/MOLARMASS(Fuel$)-
m/2*enthalpy(H2O;T=T_a)*MOLARMASS(H2O)/MOLARMASS(Fuel$)

    "Initialization of m_f considering the combustions gas as air"
    m_f1=m_in*(enthalpy(air;T=T_out)-enthalpy(air;T=T_in))/LHV

20: n_f:=m_f1/molarmass(Fuel$)   "N° moles of fuel"

    a=m_in*xO2_in/molarmass(O2)        "inlet O2 kmoles"
    b=m_in*xN2_in/molarmass(N2)        "inlet N2 kmoles"
    c=m_in*xCO2_in/molarmass(CO2)      "inlet CO2 kmoles"
    d=m_in*xH2O_in/molarmass(H2O)      "inlet H2O kmoles"

    mO2_out=(a-n_f*n-n_f*m/4)*molarmass(O2)        "outlet O2 kgs"
    mN2_out=b*molarmass(N2)                        "outlet N2 kgs"
    mCO2_out=(n_f*n+d)*molarmass(CO2)              "outlet CO2 kgs"
    mH2O_out=(n_f*m/2+c)*molarmass(H2O)            "outlet H2O kgs"

    xO2_out=mO2_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)       "outlet O2 mass fraction"
    xN2_out=mN2_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)       "outlet N2 mass fraction"
    xCO2_out=mCO2_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)     "outlet CO2 mass fraction"
    xH2O_out=mH2O_out/(mO2_out+mN2_out+mCO2_out+mH2O_out)     "outlet H2O mass fraction"

    "Enthalpy balance in CC"
    h_out=xN2_out*(enthalpy(N2;T=T_out)-
    enthalpy(N2;T=T_a))+xO2_out*(enthalpy(O2;T=T_out)-
    enthalpy(O2;T=T_a))+xCO2_out*(enthalpy(CO2;T=T_out)-
    enthalpy(CO2;T=T_a))+xH2O_out*(enthalpy(H2O;T=T_out)-enthalpy(H2O;T=T_a))

    m_f=m_in*(h_out-h_in)/(h_inf+LHV-h_out)
    if (abs((m_f-m_f1)/m_f)<0,000001) then
            m_f:=m_f1
    else
            m_f1:=m_f
            goto 20
    endif

    mf_out=mO2_out+mN2_out+mCO2_out+mH2O_out
    Delta_r=abs((m_f+m_in-mf_out)/(m_f+m_in))*100

endif

end procedure
```

1.5. Cpmedioproc

Questa procedura, nota la massa, la composizione del gas in ingresso e le sue condizioni termodinamiche e le temperature d'inizio e fine trasformazione, calcola il cp medio.



1.5.1. Programma Cpmedioproc

```
"Module for the calculation of mean cp (cpme), mean cv (cvme), mean k (kme) given the MASS fractions of the GAS MIXTURE containing O2, N2, CO2, H2O and the end temperatures T_1 (min) and T_2 (max)."

"Simbols explanation"
"Inlet data"

"x_N2=Inlet N2 mass fraction"
"x_O2=Inlet O2 mass fraction"
"x_CO2=Inlet CO2 mass fraction"
"x_H2O=Inlet H2O mass fraction"
"T_1=mixture Inlet temperature"
"T_2=mixture outlet temperature"

"Outlet data"

"cpme=average specific heat at costant pressure of mixture"
"cvme=average specific heat at costant volume of mixture"
"kme=specific heat ratio"

"Procedure for the calculation of the mean cp with 5th order interpolation from EES specheat tables (JANAF)"

procedure cpmedioproc(x_O2;x_N2;x_CO2;x_H2O;T_1;T_2:cpme;cvme;kme)
aO2=0,82390
bO2=0,00033840
```

```
cO2=-0,000000034992
dO2=-0,000000000060213
eO2=0,000000000000018831
aN2=1,0491
bN2=-0,00020297
cN2=0,00000062186
dN2=-0,00000000037256
eN2=0,000000000000070546

aCO2=0,45627
bCO2=0,0016313
cCO2=-0,0000012737
dCO2=0,00000000049469
eCO2=-0,000000000000075456

aH2O=1,8534
bH2O=-0,00028108
cH2O=0,0000012990
dH2O=-0,00000000070144
eH2O=0,000000000000012196

a_mix=x_O2*aO2+x_N2*aN2+x_CO2*aCO2+x_H2O*aH2O
b_mix=x_O2*bO2+x_N2*bN2+x_CO2*bCO2+x_H2O*bH2O
c_mix=x_O2*cO2+x_N2*cN2+x_CO2*cCO2+x_H2O*cH2O
d_mix=x_O2*dO2+x_N2*dN2+x_CO2*dCO2+x_H2O*dH2O
e_mix=x_O2*eO2+x_N2*eN2+x_CO2*eCO2+x_H2O*eH2O

if(abs(T_2-T_1)<0,000001) then
      cpme=a_mix+b_mix*T_1+c_mix*T_1^2+d_mix*T_1^3+e_mix*T_1^4
else
      cpme=1/(T_2-T_1)*(a_mix*(T_2-T_1)+b_mix*((T_2)^2-(T_1)^2)/2+c_mix*((T_2)^3-
(T_1)^3)/3+d_mix*((T_2)^4-(T_1)^4)/4+e_mix*((T_2)^5-(T_1)^5)/5)
endif

{Calcolo frazioni molari componenti}

nmol_tot=x_O2/MOLARMASS(O2)+x_N2/MOLARMASS(N2)+x_CO2/MOLARMASS(CO2)+x_H2O/M
OLARMASS(H2O)
fmol_O2=x_O2/MOLARMASS(O2)/nmol_tot
fmol_N2=x_N2/MOLARMASS(N2)/nmol_tot
fmol_CO2=x_CO2/MOLARMASS(CO2)/nmol_tot
fmol_H2O=x_H2O/MOLARMASS(H2O)/nmol_tot

pm_mix=fmol_O2*MOLARMASS(O2)+fmol_N2*MOLARMASS(N2)+fmol_CO2*MOLARMASS(CO2)+f
mol_H2O*MOLARMASS(H2O)

R_mix=R#/pm_mix
cvme=cpme-R_mix
kme=cpme/cvme

end procedure
```