

ALMA MATER STUDIORUM
UNIVERSITA' DI BOLOGNA

SCUOLA DI INGEGNERIA
Sede di Forlì

Corso di Laurea in
INGEGNERIA MECCANICA
Classe L-9

ELABORATO FINALE DI LAUREA
in Sistemi Energetici

Sviluppo di interfacce grafiche per l'analisi di Cicli Frigoriferi

CANDIDATO
Giovanni Santucci

RELATORE
Chiar.mo Prof. Ing. Davide Moro

Anno Accademico 2018/2019

Indice

1	Premessa	4
2	Introduzione	5
2.1	MATLAB [®]	5
2.2	CoolProp	6
3	Applicativo coord_term	8
3.1	Figura	8
3.2	Callback Functions	12
3.2.1	<i>calcola</i>	12
3.2.2	<i>reset</i>	13
3.2.3	<i>chiudi</i>	14
3.3	Esempio.....	14
4	Applicativo ciclo_frigo	16
4.1	Figure	18
4.1.1	<i>fig_1</i>	18
4.1.2	<i>fig_1a</i>	22
4.1.3	<i>fig_1b</i>	23
4.1.4	<i>fig_2</i>	26
4.1.5	<i>fig_2a</i>	28
4.1.6	<i>fig_3</i>	29
4.1.7	<i>fig_3a</i>	32
4.1.8	<i>fig_4</i>	34
4.1.9	<i>fig_4a</i>	36

4.2	Callback Functions	37
4.2.1	<i>curve_ciclo</i>	37
4.2.2	<i>curve_ciclo2</i>	41
4.2.3	<i>fig_1</i>	47
4.2.4	<i>fig_1a</i>	52
4.2.5	<i>fig_1b</i>	54
4.2.6	<i>fig_2</i>	56
4.2.7	<i>fig_3</i>	60
4.2.8	<i>fig_3a</i>	71
4.2.9	<i>fig_4</i>	72
4.3	Esempio	76
5	Conclusioni	82
6	Appendice	84
6.1	MATLAB®	84
6.1.1	<i>User Interface</i>	84
6.2	COOLPROP	88
6.2.1	<i>PropsSI</i>	88
6.2.2	<i>PropsISI</i>	92
6.2.3	<i>Lista fluidi</i>	94
6.3	FluidsList.mat	96
6.4	Funzioni.mat	97
6.5	Curve limite	97
7	Bibliografia	100
7.1	Sistemi Energetici	100
7.2	MATLAB®	100
7.3	CoolProp	100

1 Premessa

Lo scopo di questa tesi è la realizzazione di due procedure in MATLAB[®] che consentano di determinare rispettivamente:

- Le coordinate termodinamiche (pressione, temperatura, densità/volume specifico, entalpia, entropia e - in caso di condizione bifase - titolo) di un fluido specifico dati i valori di due funzioni di stato in ingresso, più eventualmente altri parametri “*trivial*”¹;
- I punti fondamentali di un ciclo frigorifero (ad uno o due livelli di compressione), energie scambiate ed efficienze, dati il fluido evolvente ed i parametri che definiscono univocamente un ciclo di questo genere (pressione/temperatura di evaporazione/condensazione, rendimento del compressore ed eventualmente i salti termici negli scambiatori² e le potenze termiche scambiate da questi ultimi³).

Un vantaggio, non trascurabile, di tali procedure, sta quindi nella comodità di poter effettuare valutazioni delle grandezze termodinamiche che caratterizzano un punto (o un ciclo frigorifero) senza l'utilizzo delle tabelle e dei grafici termodinamici disponibili in letteratura, ma grazie all'ausilio di CoolProp, una libreria *open source* che consente la determinazione dei parametri di stato (e non) di fluidi puri e pseudo-puri; il tutto attraverso un'interfaccia il più possibile *user-friendly*, in modo tale da ridurre - se non eliminare - la conoscenza del linguaggio MATLAB[®] necessaria all'invocazione delle funzioni necessarie.

Andremo ora a descrivere brevemente l'ambiente MATLAB[®] e le funzioni utilizzate per la creazione dell'interfaccia e il pacchetto CoolProp.

¹ Dettagli ulteriori verranno forniti in seguito

² Differenza, in valore assoluto, tra la temperatura del fluido evolvente in uscita dallo scambiatore e quella dell'ambiente frigorifero/esterno

³ In questo caso vengono calcolate anche le portate circolanti

2 Introduzione

2.1 MATLAB[®]

MATLAB[®] (abbreviazione di *Matrix Laboratory*) è un ambiente per il calcolo numerico e l'analisi statistica scritto in C, che comprende anche l'omonimo linguaggio di programmazione creato da MathWorks, Inc.

MATLAB[®] consente di manipolare matrici, visualizzare funzioni e dati, implementare algoritmi, creare interfacce utente ed interfacciarsi con altri programmi. (...) [Nonostante sia specializzato nel calcolo numerico è possibile utilizzare il Symbolic Math Toolbox[™] per risolvere, rappresentare e manipolare equazioni di matematica simbolica.]

MATLAB[®] è usato da milioni di persone nell'industria e nelle università per via dei suoi numerosi strumenti a supporto dei più disparati campi di studio applicati e funziona su diversi sistemi operativi, tra cui Windows, Mac OS, GNU/Linux e Unix.⁴

Per l'implementazione del codice ci si è serviti, oltre ai classici cicli *if*, *for* e *while*, del pacchetto di funzioni *uicontrol* - che permette di creare interfacce utente tramite l'inserimento di oggetti quali *text* (testi), *edit* (caselle editabili), *popupmenu* (menu a tendina), *pushbutton* (pulsanti), ecc., ai quali associare vere e proprie procedure chiamate *Callback Functions* -, *uitable*, per creare tabelle e *uipanel*, che fondamentalmente genera riquadri in cui inserire gli oggetti creati.⁵

⁴ Wikipedia. (n.d.). *MATLAB*. Retrieved from <https://it.wikipedia.org/wiki/MATLAB>

⁵ Per la sintassi di tali funzioni si veda l'Appendice

Nota: Lo scopo principale di questa tesi non è la creazione di un codice che sia il più efficiente possibile, né tantomeno snello; si noterà infatti che in alcuni casi sono presenti istruzioni che possono sembrare ridondanti o eccessivamente specifiche, tuttavia si è ritenuto necessario un approccio di questo genere per ridurre la complessità dell'algoritmo stesso ed aumentarne la robustezza.

2.2 CoolProp

CoolProp è una libreria *open source* scritta in C++ che implementa:

- Equazioni di stato di fluidi puri e pseudo-puri e proprietà di trasporto per 122 componenti;
- Proprietà delle miscele usando formulazioni di elevata accuratezza dell'energia di Helmholtz;
- Correlazione delle proprietà di fluidi incomprimibili e brine;
- Routine psicrometrica di elevatissima accuratezza;
- Interfaccia *user-friendly* sulle funzionalità complete di NIST REFPROP;
- Equazioni cubiche di stato (SRK, PR).

Tale libreria è sfruttabile pienamente su vari linguaggi di programmazione e piattaforme quali: Python (2.x, 3.x), Modelica, Octave, C#, VB.net, MathCAD, Java, Android, MATLAB[®], mentre è disponibile unicamente un'interfaccia di alto livello per: Labview, EES, Microsoft Excel, LibreOffice, Javascript, PHP, FORTRAN, Maple, Mathematica, Scilab, Delphi & Lazarus, Julia.

Sono supportate sia architetture a 32-bit che a 64-bit, su sistemi operativi quali: Windows, Linux, OSX, Raspberry PI, VxWorks Compact Rio, ecc. purché sia possibile compilare in C++.

I principali sviluppatori sono:

- Ian Bell - Bell Thermal Consultants;
- Jorrit Wronski - IPU Refrigeration and Energy Technology, Kgs. Lyngby, Denmark;

- Sylvain Quoilin, Vincent Lemort - Thermodynamics Laboratory, University of Liege, Liege, Belgium.

Nota: Quasi tutti i modelli dei fluidi implementati in CoolProp si basano su formulazioni dell'energia di Helmholtz, ciò consente di ottenere tutte le proprietà termodinamiche di interesse direttamente dalle derivate parziali di tale energia.⁶

Per l'implementazione del codice ci si è serviti di due funzioni della libreria: "Props1SI", per la determinazione dei parametri cosiddetti "*trivial*", ovvero propri del fluido in sé (non è richiesto in input il suo particolare stato termodinamico) e "PropsSI", per la determinazione dei parametri di stato del fluido (in input è necessario inserire il valore di due coordinate termodinamiche).⁷

Andremo ora ad analizzare le specifiche porzioni di codice, illustrandone le funzionalità.

⁶ CoolProp Team. (n.d.). *Welcome to CoolProp*. Retrieved from <http://www.coolprop.org/>

⁷ Per la sintassi di tali funzioni si veda l'Appendice

3 Applicativo coord_term

Questa procedura consente la determinazione - dopo che l'utente ha scelto nel pannello di input il fluido di interesse, le 2 coordinate termodinamiche, eventualmente la funzione speciale da calcolare (dai menu a tendina) ed ha inserito i valori rispettivi di tali coordinate (nelle caselle editabili) - di tutte le coordinate termodinamiche del fluido più il valore della funzione speciale (se questa è stata inserita).

Nota: Se la funzione speciale è una funzione “*trivial*” non è necessario inserire il valore delle due coordinate di input (in tal caso in output verrà mostrato unicamente il parametro della funzione speciale).

Il codice sorgente è il seguente:

```
function coord_term
addpath(genpath(pwd)) %include tutte le sottocartelle nel path
```

3.1 Figura

```
fig=figure; fig.Position=[0 0 600 420];
fig.Name='Coordinate termodinamiche'; fig.MenuBar='none';
fig.NumberTitle='off'; fig.Resize='off'; movegui(fig,'center')

%OUTPUT
C={'Pressione (Pa)', 'Temperatura (K)', 'Densità (kg/m^3)', ...
  'Volume specifico (kg/m^3)', 'Entalpia (J/kg)', 'Entropia (J/(kg*K))', ...
  'Titolo (0-1)', 'Funzione speciale'};
ut1=uitable('Parent',fig);
ut1.FontSize=10; ut1.RowName='Risultati';
ut1.ColumnName=C(1:4); ut1.RearrangeableColumns='on';
ut1.ColumnEditable=true;
ut1.Units='normalized'; ut1.Position=[.05 .2 ut1.Extent(3) ut1.Extent(4)];
```



```
ut2=uitable('Parent',fig);
ut2.FontSize=10; ut2.RowName='Risultati';
ut2.ColumnName=C(5:8); ut2.RearrangeableColumns='on';
ut2.ColumnEditable=true;
ut2.Units='normalized'; ut2.Position=[.05 .05 ut2.Extent(3) ut2.Extent(4)];

%Pannello di input
ip=uipanel('Parent',fig,'Title','Input'); ip.Position=[.05 .4 .65 .55];
ip.FontSize=12; ip.FontWeight='bold';

%Selezione fluido
t_fl=icontrol('Parent',ip,'Style','text');
t_fl.Units='normalized'; t_fl.Position=[.07 .8 .3 .15];
t_fl.FontSize=11; t_fl.String='Fluido';

FluidsList=load('FluidsList.mat');
pm_fl=icontrol('Parent',ip,'Style','popupmenu');
pm_fl.Units='normalized'; pm_fl.Position=[.31 .81 .35 .15];
pm_fl.FontSize=11; pm_fl.String=FluidsList.FluidsList;

%Coordinate
t1=icontrol('Parent',ip,'Style','text');
t1.FontSize=11; t1.String='1^ coordinata';
t1.Units='normalized'; t1.Position=[0 .6 .3 .15];

t2=icontrol('Parent',ip,'Style','text');
t2.FontSize=11; t2.String='2^ coordinata';
t2.Units='normalized'; t2.Position=[0 .4 .3 .15];

C(4)=[];
pm1=icontrol('Parent',ip,'Style','popupmenu');
pm1.Units='normalized'; pm1.Position=[.31 .61 .35 .15];
pm1.FontSize=11; pm1.String=C(1:6);

pm2=icontrol('Parent',ip,'Style','popupmenu');
pm2.Units='normalized'; pm2.Position=[.31 .41 .35 .15];
pm2.FontSize=11; pm2.String=C(1:6);
pm2.Value=2;
```

```

e1=uicontrol('Parent',ip,'Style','edit');
e1.Units='normalized'; e1.Position=[.675 .62 .3 .15];
e1.FontSize=10;

e2=uicontrol('Parent',ip,'Style','edit');
e2.Units='normalized'; e2.Position=[.675 .42 .3 .15];
e2.FontSize=10;

%Pannello Funzione speciale
ip_fz=uipanel('Parent',ip,'Title','Funzione speciale');
ip_fz.Position=[.025 .05 .95 .35];
ip_fz.FontSize=12; ip_fz.FontWeight='bold';

    Funzioni=load('Funzioni.mat');
    pm_fz=uicontrol('Parent',ip_fz,'Style','popupmenu');
    pm_fz.Units='normalized'; pm_fz.Position=[.025 .05 .925 .75];
    pm_fz.FontSize=11; pm_fz.String=Funzioni.Funzioni(:,2);

%PULSANTI
pb_cal=uicontrol('Parent',fig,'Style','pushbutton');
pb_cal.Units='normalized'; pb_cal.Position=[.75 .775 .2 .1];
pb_cal.FontSize=12; pb_cal.FontWeight='bold';
pb_cal.String='Calcola'; pb_cal.Callback=@calcola;

pb_res=uicontrol('Parent',fig,'Style','pushbutton');
pb_res.Units='normalized'; pb_res.Position=[.75 .625 .2 .1];
pb_res.FontSize=12; pb_res.FontWeight='bold';
pb_res.String='Reset'; pb_res.Callback=@reset;

pb_chi=uicontrol('Parent',fig,'Style','pushbutton');
pb_chi.Units='normalized'; pb_chi.Position=[.75 .475 .2 .1];
pb_chi.FontSize=12; pb_chi.FontWeight='bold';
pb_chi.String='Chiudi'; pb_chi.Callback=@chiudi;

```

Viene quindi innanzitutto creata la figura denominata “fig”, dopodiché vengono costruite le tabelle “ut1” e “ut2” che conterranno i risultati di output; successivamente viene inizializzato il pannello di input “ip”, il quale contiene:

- La casella di testo “t_fl” e il menu a tendina “pm_fl” (che contiene gli elementi di “FluidsList.mat”) per la selezione del fluido;
- La casella di testo “t1”, il menu a tendina “pm1” (che contiene “Pressione (Pa)”, “Temperatura (K)”, “Densità (kg/m³)”, “Entalpia (J/kg)”, “Entropia (J/(kg*K))” e “Titolo”) e la casella editabile “e1” per la selezione e l’inserimento della prima coordinata termodinamica;
- La casella di testo “t2”, il menu a tendina “pm2” (che contiene le stesse opzioni di “pm1”) e la casella editabile “e2” per la selezione e l’inserimento della seconda coordinata termodinamica;

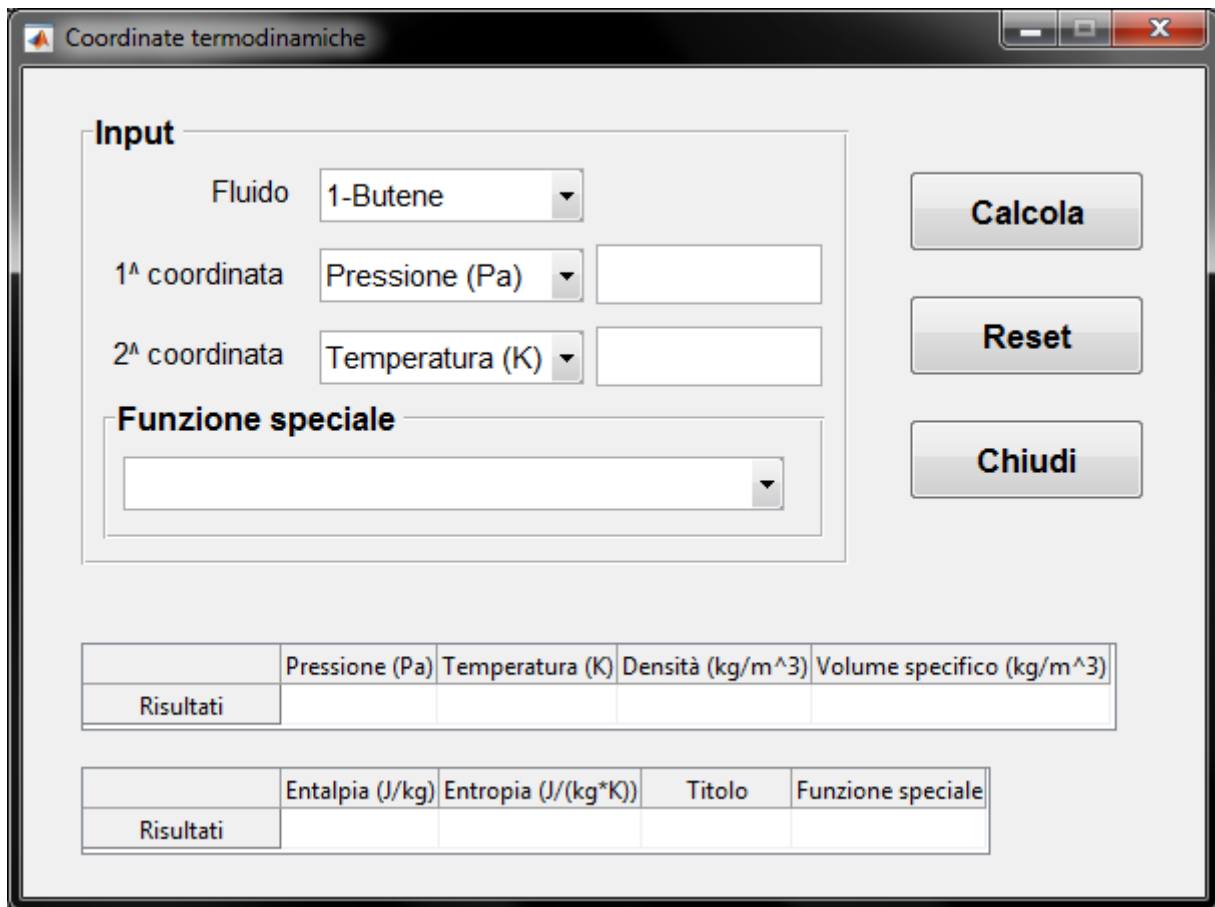
Nota: Il valore di “pm2” è stato impostato su 2, ovvero è selezionata l’opzione “Temperatura (K)”.

- Il pannello “ip_fz”, il quale a sua volta contiene il menu a tendina “pm_fz” (che contiene gli elementi della seconda colonna di “Funzioni.mat”) per l’eventuale selezione della funzione speciale da calcolare.

Nota: Il valore di default di “pm_fz” è 1, ovvero non è selezionata nessuna funzione speciale.

Infine vengono creati i pulsanti “pb_cal” (Calcola), “pb_res” (Reset) e “pb_chi” (Chiudi).

Questo è il risultato:



3.2 Callback Functions

3.2.1 calcola

Viene invocata alla pressione del pulsante “pb_cal”; calcola, utilizzando i valori inseriti nel pannello di input, tutte le coordinate termodinamiche del fluido (se sono state compilate le caselle editabili), eventualmente il valore del parametro speciale (se è stato selezionato nel menu a tendina “pm_fz”) e inserisce i risultati nelle tabelle “ut1” e “ut2”.

```
function calcola(source,eventdata)
    val={'P','T','D','H','S','Q'};
    S1=cell2mat(val(pm1.Value)); N1=str2num(e1.String);
    S2=cell2mat(val(pm2.Value)); N2=str2num(e2.String);
    Fluid=cell2mat(pm_fl.String(pm_fl.Value));

    ut1.Data={' ',' ',' ',' '}; ut2.Data={' ',' ',' ',' '};
```

```

if ~isempty(e1.String) && ~isempty(e2.String)
    %se sono state inserite le coordinate termodinamiche
    for i=1:6
        risultati(i)={CoolProp.PropsSI(cell2mat(val(i)),S1,N1,...
            S2,N2,Fluid)};
    end

    ut1.Data=risultati(1:3); ut1.Data(4)={1/cell2mat(ut1.Data(3))};
    ut2.Data=risultati(4:6);

    if pm_fz.Value>=27 %se è stata inserita una funz. non trivial
        ut2.Data(4)={CoolProp.PropsSI(...
            cell2mat(Funzioni.Funzioni(pm_fz.Value,1)),S1,N1,S2,...
            N2,Fluid)};
    end
end

if pm_fz.Value~=1 && pm_fz.Value<=26
    %se è stata inserita una funz. trivial
    ut2.Data(4)={CoolProp.PropsSI(cell2mat(...
        Funzioni.Funzioni(pm_fz.Value,1)),Fluid)};
end

if cell2mat(ut2.Data(3))== -1
    ut2.Data(3)={' '};
end
end
end

```

3.2.2reset

Viene invocata alla pressione del pulsante “pb_res”; elimina i valori inseriti nelle caselle editabili, i risultati inseriti nelle tabelle di output e riporta il menu a tendina “pm_fz” al valore di default (nessuna funzione speciale è inserita).

```

function reset(source,eventdata)
    e1.String=''; e2.String='';
    ut1.Data=[]; ut2.Data=[];
    pm_fz.Value=1;
end

```

3.2.3chiudi

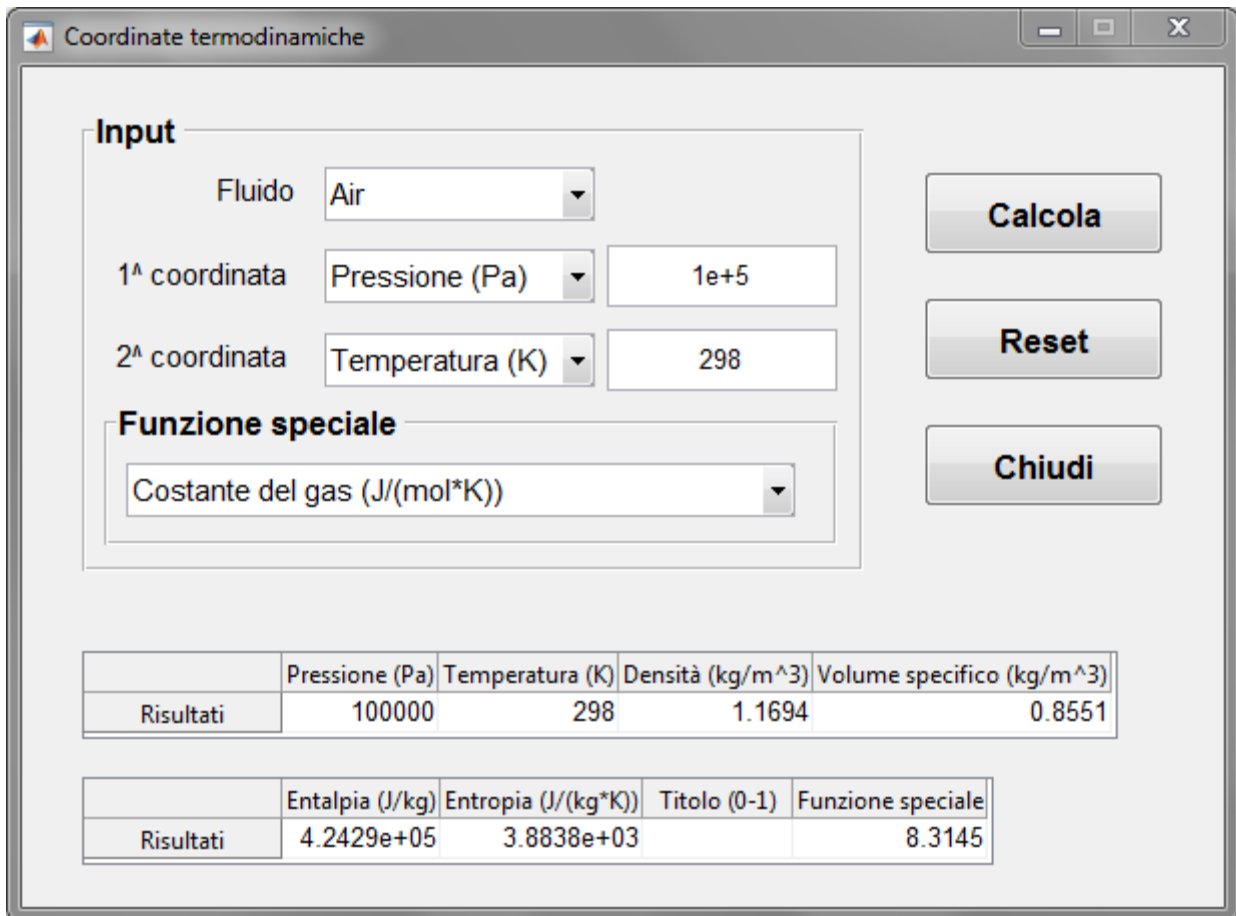
Viene invocata alla pressione del pulsante “pb_chi”; chiude la finestra.

```
function chiudi(source,eventdata)
    close(fig)
end
```

```
end
```

3.3 Esempio

Come parametri di input sono stati inseriti il fluido (aria), la pressione ambiente (1 bar) e la temperatura ambiente (25 °C); dopodiché si è scelto di calcolare anche la costante del gas ($R = \frac{R_0}{M}$, con R_0 costante universale dei gas e M massa molare del gas considerato).



Nota: Il valore del titolo non è presente nella tabella dei risultati in quanto l'aria in condizioni standard è una miscela di gas incondensabili (ovvero di fluidi supercritici).

4 Applicativo ciclo frigo

Questa procedura consente la determinazione - dopo che l'utente ha selezionato nella prima finestra ("fig_1") il fluido di interesse, ha inserito le pressioni (o temperature) di evaporazione e condensazione ed eventualmente modificato il rendimento di compressione ed il salto termico al condensatore e all'evaporatore - delle coordinate termodinamiche (pressione, temperatura, densità/volume specifico, entalpia, entropia e titolo) dei 4 punti fondamentali del corrispondente ciclo frigorifero semplice e dei parametri che lo caratterizzano: calore scambiato all'evaporatore e al condensatore, energia assorbita dal compressore ed efficienze⁸ (frigorifera e della pompa di calore).

Nella prima finestra è anche possibile inserire le potenze scambiate all'evaporatore e/o al condensatore (direttamente o tramite i parametri coefficiente globale e superficie di scambio termico).

Nota: Se almeno una potenza è stata inserita i risultati di output diventano: potenza termica scambiata all'evaporatore e al condensatore, potenza meccanica assorbita dal compressore, efficienze (frigorifera e della pompa di calore) e portata di refrigerante.

Inoltre se sono state inserite sia la potenza scambiata all'evaporatore che quella scambiata al condensatore, quest'ultima viene ignorata.

Nella seconda finestra ("fig_2") è presente uno spazio in cui viene generato il grafico contenente la curva limite, il ciclo frigorifero e le isoterme dell'ambiente esterno/frigorifero; tramite un menu a tendina è possibile selezionare il tipo di grafico visualizzato: p-h (default), T-s, h-s, p-v.

⁸ Solitamente indicate con EER (*Energy Efficiency Ratio*) per l'evaporatore e COP (*Coefficient Of Performance*) per il condensatore

Da tale finestra è possibile aprire un'ulteriore interfaccia (“fig_3”) per inserire un livello intermedio di evaporazione: l'utente può ancora una volta scegliere se inserire la pressione o la temperatura di 2° livello (l'inserimento è guidato da uno *slider*), eventualmente modificare il salto termico rispetto al nuovo ambiente frigorifero e/o inserire la potenza scambiata al 2° evaporatore (come avveniva per il 1° evaporatore ed il condensatore).

Dopodiché si apre una quarta finestra (“fig_4”) contenente le coordinate termodinamiche degli 8 punti fondamentali del nuovo ciclo ed i parametri che lo caratterizzano (calore scambiato al 1°/2° evaporatore e al condensatore ed energia assorbita dal 1°/2° compressore).

Nota: Se precedentemente sono state inserite (direttamente o tramite i parametri) almeno 2 potenze scambiate (evaporatori/compressore) i risultati di output diventano: potenza termica scambiata al 1°/2° evaporatore e al condensatore e potenza meccanica assorbita dal 1°/2° compressore.

Se è stata inserita solamente la potenza scambiata al 1° evaporatore vengono determinati: potenza meccanica assorbita dal 1° compressore, calore scambiato al 2° evaporatore e al condensatore ed energia assorbita dal 2° compressore.

Se è stata inserita solamente la potenza scambiata al condensatore vengono determinati: potenza meccanica assorbita dal 2° compressore, calore scambiato al 1° evaporatore ed energia assorbita dal 1° compressore.

Se invece è stata inserita solamente la potenza scambiata al 2° evaporatore vengono riportati i parametri di base, ovvero: calore scambiato al 1°/2° evaporatore e al condensatore ed energia assorbita dal 1°/2° compressore.

In concomitanza all'apertura di “fig_2” e “fig_4” vengono aperte rispettivamente “fig_2a” e “fig_4a” che contengono uno schema dell'impianto con i punti fondamentali numerati.

Il codice sorgente è il seguente:

```
function ciclo_frigo

addpath(genpath(pwd)) %include tutte le sottocartelle nel path
save('Ris.mat') %inizializzazione struct risultati
```

4.1 Figure

4.1.1 fig_1

È l'interfaccia che consente all'utente di inserire i parametri di input per il ciclo frigorifero semplice.

```
fig_1=figure; fig_1.Position=[0 0 560 360];
fig_1.Name='Ciclo frigorifero'; fig_1.MenuBar='none';
fig_1.NumberTitle='off'; fig_1.Resize='off'; movegui(fig_1,'center')

ip_1=uipanel('Parent',fig_1,'Title','Ciclo semplice');
ip_1.Position=[.05 .2 .7 .75];
ip_1.FontSize=12; ip_1.FontWeight='bold';

t_fl_1=icontrol('Parent',ip_1,'Style','text');
t_fl_1.Units='normalized'; t_fl_1.Position=[.2 .8 .2 .15];
t_fl_1.FontSize=12; t_fl_1.String='Fluidi';

FluidsList=load('FluidsList.mat');
pm_fl_1=icontrol('Parent',ip_1,'Style','popupmenu');
pm_fl_1.Units='normalized'; pm_fl_1.Position=[.4 .81 .3 .15];
pm_fl_1.FontSize=12; pm_fl_1.String=FluidsList.FluidsList;

pm1_1=icontrol('Parent',ip_1,'Style','popupmenu');
pm1_1.Units='normalized'; pm1_1.Position=[.05 .65 .65 .15];
pm1_1.FontSize=11;
pm1_1.String={'Pressione di evaporazione (Pa)',...
             'Temperatura di evaporazione (K)'};

e1_1=icontrol('Parent',ip_1,'Style','edit');
e1_1.Units='normalized'; e1_1.Position=[.75 .66 .2 .15];
e1_1.FontSize=10;
```

```

pm2_1=uicontrol('Parent',ip_1,'Style','popupmenu');
pm2_1.Units='normalized'; pm2_1.Position=[.05 .49 .65 .15];
pm2_1.FontSize=11;
pm2_1.String={'Pressione di condensazione (Pa)',...
    'Temperatura di condensazione (K)'};

e2_1=uicontrol('Parent',ip_1,'Style','edit');
e2_1.Units='normalized'; e2_1.Position=[.75 .5 .2 .15];
e2_1.FontSize=10;

t_n_1=uicontrol('Parent',ip_1,'Style','text');
t_n_1.Units='normalized'; t_n_1.Position=[.05 .32 .65 .15];
t_n_1.FontSize=11; t_n_1.String='Rendimento compressore';

e_n_1=uicontrol('Parent',ip_1,'Style','edit');
e_n_1.Units='normalized'; e_n_1.Position=[.75 .34 .2 .15];
e_n_1.FontSize=10; e_n_1.String='1';

t_dtf_1=uicontrol('Parent',ip_1,'Style','text');
t_dtf_1.Units='normalized'; t_dtf_1.Position=[.05 .16 .65 .15];
t_dtf_1.FontSize=11;
t_dtf_1.String=[char(916), 'T ambiente frigorifero (K)'];
    %char(916)->Delta maiuscola

e_dtf_1=uicontrol('Parent',ip_1,'Style','edit');
e_dtf_1.Units='normalized'; e_dtf_1.Position=[.75 .18 .2 .15];
e_dtf_1.FontSize=10; e_dtf_1.String='10';

t_dta_1=uicontrol('Parent',ip_1,'Style','text');
t_dta_1.Units='normalized'; t_dta_1.Position=[.05 0 .65 .15];
t_dta_1.FontSize=11;
t_dta_1.String=[char(916), 'T ambiente esterno (K)'];
    %char(916)->Delta maiuscola

e_dta_1=uicontrol('Parent',ip_1,'Style','edit');
e_dta_1.Units='normalized'; e_dta_1.Position=[.75 .02 .2 .15];
e_dta_1.FontSize=10; e_dta_1.String='10';

```

```
%PULSANTI
pb_cal_1=uicontrol('Parent',fig_1,'Style','pushbutton');
pb_cal_1.Units='normalized'; pb_cal_1.Position=[.775 .7 .2 .1];
pb_cal_1.FontSize=12; pb_cal_1.FontWeight='bold';
pb_cal_1.String='Calcola'; pb_cal_1.Callback=@calcola;

pb_res_1=uicontrol('Parent',fig_1,'Style','pushbutton');
pb_res_1.Units='normalized'; pb_res_1.Position=[.775 .5 .2 .1];
pb_res_1.FontSize=12; pb_res_1.FontWeight='bold';
pb_res_1.String='Reset'; pb_res_1.Callback=@reset;

pb_chi_1=uicontrol('Parent',fig_1,'Style','pushbutton');
pb_chi_1.Units='normalized'; pb_chi_1.Position=[.775 .3 .2 .1];
pb_chi_1.FontSize=12; pb_chi_1.FontWeight='bold';
pb_chi_1.String='Chiudi tutto'; pb_chi_1.Callback=@chiudi;

pb_evap_1=uicontrol('Parent',fig_1,'Style','pushbutton');
pb_evap_1.Units='normalized'; pb_evap_1.Position=[.05 .05 .425 .1];
pb_evap_1.FontSize=12; pb_evap_1.FontWeight='bold';
pb_evap_1.String='Caratteristiche evaporatore'; pb_evap_1.Callback=@evap;

pb_cond_1=uicontrol('Parent',fig_1,'Style','pushbutton');
pb_cond_1.Units='normalized'; pb_cond_1.Position=[.525 .05 .425 .1];
pb_cond_1.FontSize=12; pb_cond_1.FontWeight='bold';
pb_cond_1.String='Caratteristiche condensatore'; pb_cond_1.Callback=@cond;
```

Viene quindi innanzitutto creata la figura, denominata “fig_1”, dopodiché viene costruito il pannello “ip_1” (Ciclo semplice), il quale contiene:

- La casella di testo “t_fl_1” e il menu a tendina “pm_fl_1” (che contiene gli elementi di “FluidsList.mat”) per la selezione del fluido;
- Il menu a tendina “pm1_1” (che contiene “Pressione di evaporazione (Pa)” e “Temperatura di evaporazione (K)”) e la casella editabile “e1_1” per la selezione e l’inserimento della pressione/temperatura di evaporazione;

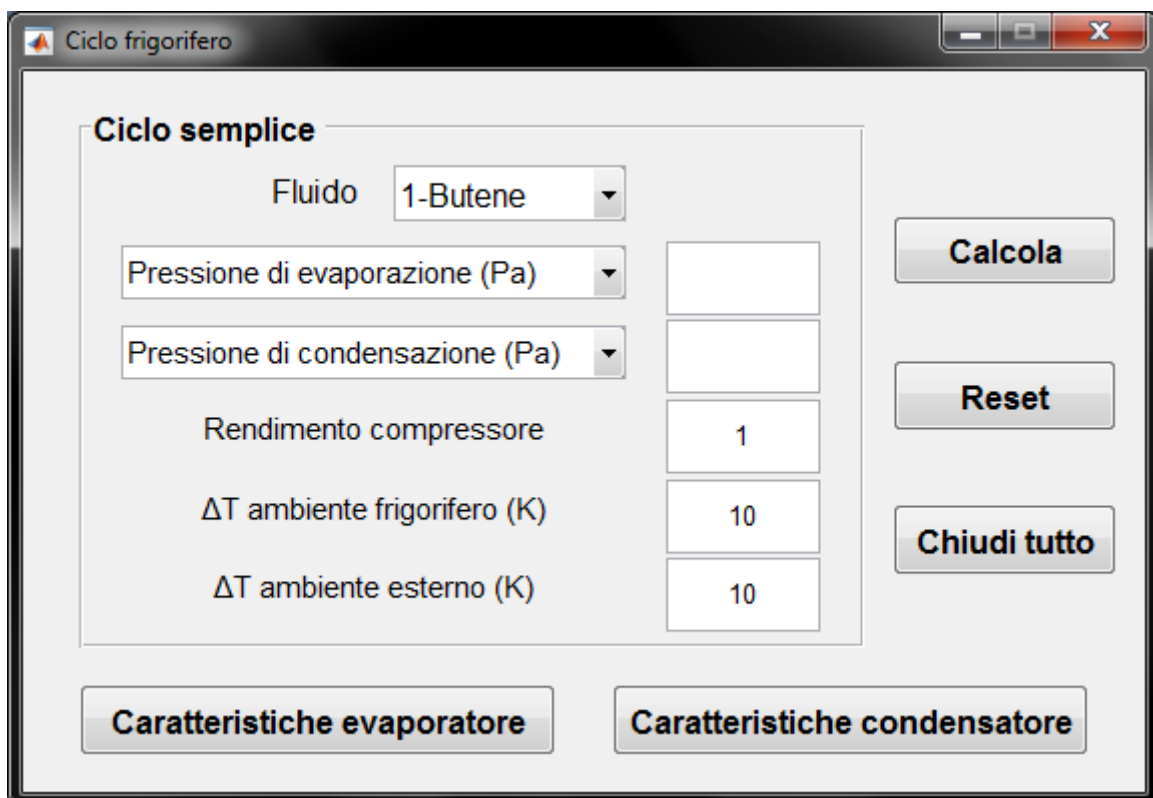
- Il menu a tendina “pm2_1” (che contiene “Pressione di condensazione (Pa)” e “Temperatura di condensazione (K)”) e la casella editabile “e2_1” per la selezione e l’inserimento della pressione/temperatura di condensazione;
- La casella di testo “t_n_1” e la casella editabile “e_n_1” per l’inserimento del rendimento di compressione;

Nota: Il valore impostato di default è 1.

- La casella di testo “t_dtf_1” e la casella editabile “e_dtf_1” per l’inserimento del salto termico all’evaporatore;
- La casella di testo “t_dta_1” e la casella editabile “e_dta_1” per l’inserimento del salto termico al condensatore.

Infine vengono creati i pulsanti “pb_cal_1” (Calcola), “pb_res_1” (Reset), “pb_chi_1” (Chiudi tutto), “pb_evap_1” (Caratteristiche evaporatore) e “pb_cond_1” (Caratteristiche condensatore).

Questo è il risultato:



4.1.2fig_1a

È l'interfaccia che consente all'utente di inserire la potenza scambiata al condensatore (direttamente o tramite l'inserimento del coefficiente globale e della superficie di scambio termico).

```

fig_1a=figure; fig_1a.Visible='off'; fig_1a.Position=[0 0 400 225];
fig_1a.Name='Caratteristiche condensatore'; fig_1a.MenuBar='none';
fig_1a.NumberTitle='off'; fig_1a.Resize='off'; movegui(fig_1a,'center')

pm_1a=uicontrol('Parent',fig_1a,'Style','popupmenu');
pm_1a.Units='normalized'; pm_1a.Position=[.3 .775 .4 .15];
pm_1a.FontSize=12; pm_1a.String={'Potenza scambiata', ...
    ['Parametri ',char(945),'-S']}; %char(945)->alfa minuscola
pm_1a.Callback=@popm_1a;

ip_1a=uipanel('Parent',fig_1a);
ip_1a.Position=[.05 .3 .9 .45];

    t_a_1a=uicontrol('Parent',ip_1a,'Style','text'); t_a_1a.Visible='off';
    t_a_1a.Units='normalized'; t_a_1a.Position=[0 .5 .7 .3];
    t_a_1a.FontSize=11; t_a_1a.String='Coeff. convezione (W/(m^2*K))';

    e_a_1a=uicontrol('Parent',ip_1a,'Style','edit'); e_a_1a.Visible='off';
    e_a_1a.Units='normalized'; e_a_1a.Position=[.65 .54 .3 .3];
    e_a_1a.FontSize=10;

    t_s_1a=uicontrol('Parent',ip_1a,'Style','text'); t_s_1a.Visible='off';
    t_s_1a.Units='normalized'; t_s_1a.Position=[0 .1 .7 .3];
    t_s_1a.FontSize=11; t_s_1a.String='Sup. scambio termico (m^2)';

    e_s_1a=uicontrol('Parent',ip_1a,'Style','edit'); e_s_1a.Visible='off';
    e_s_1a.Units='normalized'; e_s_1a.Position=[.65 .14 .3 .3];
    e_s_1a.FontSize=10;

    t_p_1a=uicontrol('Parent',ip_1a,'Style','text');
    t_p_1a.Units='normalized'; t_p_1a.Position=[0 .4 .7 .2];
    t_p_1a.FontSize=11; t_p_1a.String='Potenza scambiata (W)';

    e_p_1a=uicontrol('Parent',ip_1a,'Style','edit');
    e_p_1a.Units='normalized'; e_p_1a.Position=[.65 .35 .3 .3];
    e_p_1a.FontSize=10;

%PULSANTI
pb_ok_1a=uicontrol('Parent',fig_1a,'Style','pushbutton');
pb_ok_1a.Units='normalized'; pb_ok_1a.Position=[.15 .1 .3 .15];
pb_ok_1a.FontSize=12; pb_ok_1a.FontWeight='bold';
pb_ok_1a.String='Ok'; pb_ok_1a.Callback=@ok_1a;

pb_ann_1a=uicontrol('Parent',fig_1a,'Style','pushbutton');
pb_ann_1a.Units='normalized'; pb_ann_1a.Position=[.55 .1 .3 .15];
pb_ann_1a.FontSize=12; pb_ann_1a.FontWeight='bold';
pb_ann_1a.String='Annulla'; pb_ann_1a.Callback=@ann_1a;

```

Viene quindi innanzitutto creata la figura, denominata “fig_1a”, dopodiché viene creato il menu a tendina “pm_1a” (che contiene “Potenza scambiata” e “Parametri α -S”) e costruito il pannello “ip_1a”, il quale contiene:

- La casella di testo “t_a_1a” e la casella editabile “e_a_1a” per l’inserimento del coefficiente globale di scambio termico;

Nota: Di default la loro visualizzazione è disattivata (verrà attivata tramite una *Callback Function*).

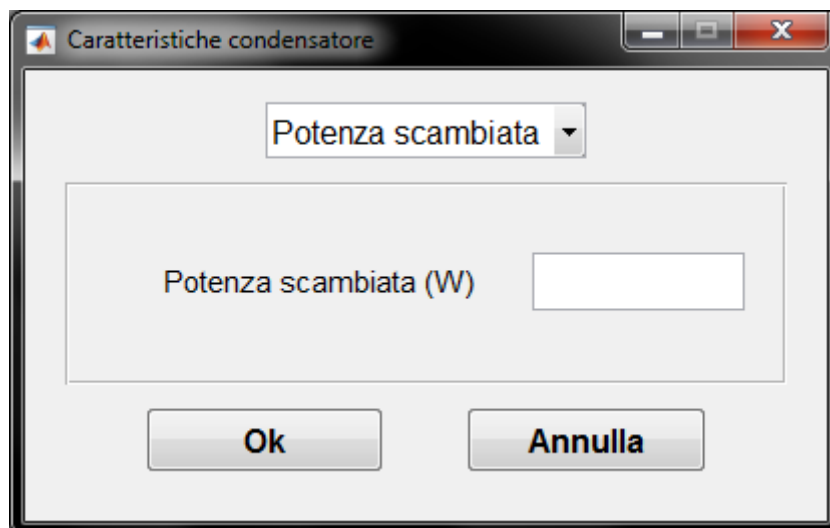
- La casella di testo “t_s_1a” e la casella editabile “e_s_1a” per l’inserimento della superficie di scambio termico;

Nota: Di default la loro visualizzazione è disattivata (verrà attivata tramite una *Callback Function*).

- La casella di testo “t_p_1a” e la casella editabile “e_p_1a” per l’inserimento diretto della potenza scambiata.

Infine vengono creati i pulsanti “pb_ok_1a” (Ok) e “pb_ann_1a” (Annulla).

Questo è il risultato:



4.1.3fig_1b

È l’interfaccia che consente all’utente di inserire la potenza scambiata all’evaporatore (direttamente o tramite l’inserimento del coefficiente globale e della superficie di scambio termico).

```

fig_1b=figure; fig_1b.Visible='off'; fig_1b.Position=[0 0 400 225];
fig_1b.Name='Caratteristiche evaporatore'; fig_1b.MenuBar='none';
fig_1b.NumberTitle='off'; fig_1b.Resize='off'; movegui(fig_1b,'center')

pm_1b=uicontrol('Parent',fig_1b,'Style','popupmenu');
pm_1b.Units='normalized'; pm_1b.Position=[.3 .775 .4 .15];
pm_1b.FontSize=12; pm_1b.String={'Potenza scambiata',...
    ['Parametri ',char(945),'-S']}; %char(945)->alfa minuscola
pm_1b.Callback=@popm_1b;

ip_1b=uipanel('Parent',fig_1b);
ip_1b.Position=[.05 .3 .9 .45];

    t_a_1b=uicontrol('Parent',ip_1b,'Style','text'); t_a_1b.Visible='off';
    t_a_1b.Units='normalized'; t_a_1b.Position=[0 .5 .7 .3];
    t_a_1b.FontSize=11; t_a_1b.String='Coeff. convezione (W/(m^2*K))';

    e_a_1b=uicontrol('Parent',ip_1b,'Style','edit'); e_a_1b.Visible='off';
    e_a_1b.Units='normalized'; e_a_1b.Position=[.65 .54 .3 .3];
    e_a_1b.FontSize=10;

    t_s_1b=uicontrol('Parent',ip_1b,'Style','text'); t_s_1b.Visible='off';
    t_s_1b.Units='normalized'; t_s_1b.Position=[0 .1 .7 .3];
    t_s_1b.FontSize=11; t_s_1b.String='Sup. scambio termico (m^2)';

    e_s_1b=uicontrol('Parent',ip_1b,'Style','edit'); e_s_1b.Visible='off';
    e_s_1b.Units='normalized'; e_s_1b.Position=[.65 .14 .3 .3];
    e_s_1b.FontSize=10;

    t_p_1b=uicontrol('Parent',ip_1b,'Style','text');
    t_p_1b.Units='normalized'; t_p_1b.Position=[0 .4 .7 .2];
    t_p_1b.FontSize=11; t_p_1b.String='Potenza scambiata (W)';

    e_p_1b=uicontrol('Parent',ip_1b,'Style','edit');
    e_p_1b.Units='normalized'; e_p_1b.Position=[.65 .35 .3 .3];
    e_p_1b.FontSize=10;

```



```
%PULSANTI
pb_ok_1b=uicontrol('Parent',fig_1b,'Style','pushbutton');
pb_ok_1b.Units='normalized'; pb_ok_1b.Position=[.15 .1 .3 .15];
pb_ok_1b.FontSize=12; pb_ok_1b.FontWeight='bold';
pb_ok_1b.String='Ok'; pb_ok_1b.Callback=@ok_1b;

pb_ann_1b=uicontrol('Parent',fig_1b,'Style','pushbutton');
pb_ann_1b.Units='normalized'; pb_ann_1b.Position=[.55 .1 .3 .15];
pb_ann_1b.FontSize=12; pb_ann_1b.FontWeight='bold';
pb_ann_1b.String='Annulla'; pb_ann_1b.Callback=@ann_1b;
```

Viene quindi innanzitutto creata la figura, denominata “fig_1b”, dopodiché viene creato il menu a tendina “pm_1b” (che contiene “Potenza scambiata” e “Parametri α -S”) e costruito il pannello “ip_1b”, il quale contiene:

- La casella di testo “t_a_1b” e la casella editabile “e_a_1b” per l’inserimento del coefficiente globale di scambio termico;

Nota: Di default la loro visualizzazione è disattivata (verrà attivata tramite una *Callback Function*).

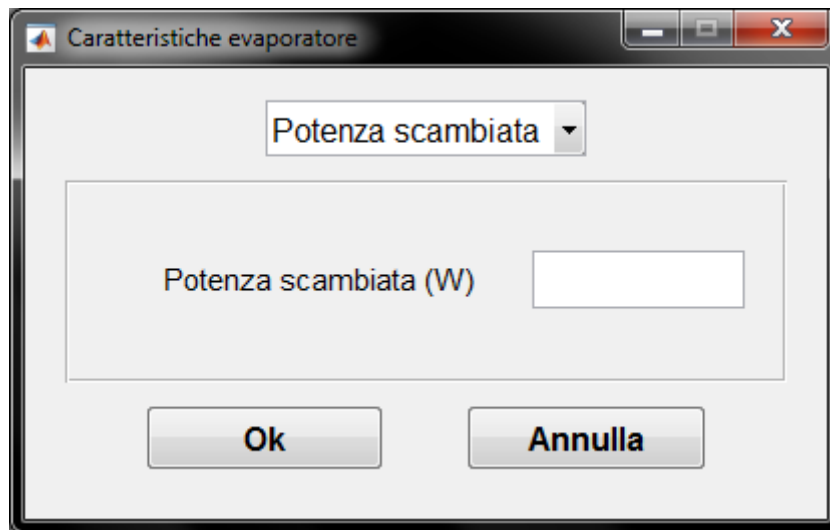
- La casella di testo “t_s_1b” e la casella editabile “e_s_1b” per l’inserimento della superficie di scambio termico;

Nota: Di default la loro visualizzazione è disattivata (verrà attivata tramite una *Callback Function*).

- La casella di testo “t_p_1b” e la casella editabile “e_p_1b” per l’inserimento diretto della potenza scambiata.

Infine vengono creati i pulsanti “pb_ok_1b” (Ok) e “pb_ann_1b” (Annulla).

Questo è il risultato:



4.1.4fig_2

È l'interfaccia che mostra i grafici e i parametri del ciclo frigorifero semplice, con le coordinate termodinamiche dei 4 punti fondamentali che lo caratterizzano.

```

fig_2=figure; fig_2.Visible='off'; fig_2.Position=[0 0 1100 625];
fig_2.Name='Ciclo semplice'; fig_2.MenuBar='none';
fig_2.NumberTitle='off'; fig_2.Resize='off'; movegui(fig_2,'center')

g_2=axes(fig_2); g_2.Parent=fig_2;
g_2.Units='normalized'; g_2.Position=[.05 .1 .4 .85];

pm_g_2=uicontrol('Parent',fig_2,'Style','popupmenu');
pm_g_2.Units='normalized'; pm_g_2.Position=[.46 .8 .06 .15];
pm_g_2.FontSize=11; pm_g_2.FontWeight='bold';
pm_g_2.String={'p_h','T_s','h_s','p_v'}; pm_g_2.Callback=@cambia_grafico;

%TABELLE RISULTATI
%energie/efficienze
utl_2=uitable('Parent',fig_2);
utl_2.FontSize=10; utl_2.ColumnName={'Risultati'};
utl_2.ColumnEditable=true;
    
```

```

%punti ciclo
ut2_2=uitable('Parent',fig_2);
ut2_2.FontSize=10; ut2_2.ColumnName={'P1','P2','P3','P4'};
ut2_2.ColumnEditable=true;
ut2_2.RowName={'Pressione (Pa)','Temperatura (K)','Densità (kg/m^3)',...
    'Volume specifico (kg/m^3)','Entalpia (J/kg)','Entropia (J/(kg*K))',...
    'Titolo'};
ut2_2.Units='normalized';
ut2_2.Position=[.46 .22 ut2_2.Extent(3) ut2_2.Extent(4)];

%PULSANTI
pb_ins_2=uicontrol('Parent',fig_2,'Style','pushbutton');
pb_ins_2.Units='normalized'; pb_ins_2.Position=[.46 .025 .255 .08];
pb_ins_2.FontSize=12; pb_ins_2.FontWeight='bold';
pb_ins_2.String='Inserisci secondo livello'; pb_ins_2.Callback=@ins_2;

pb_ind_2=uicontrol('Parent',fig_2,'Style','pushbutton');
pb_ind_2.Units='normalized'; pb_ind_2.Position=[.72 .025 .132 .08];
pb_ind_2.FontSize=12; pb_ind_2.FontWeight='bold';
pb_ind_2.String='Indietro'; pb_ind_2.Callback=@ind_2;

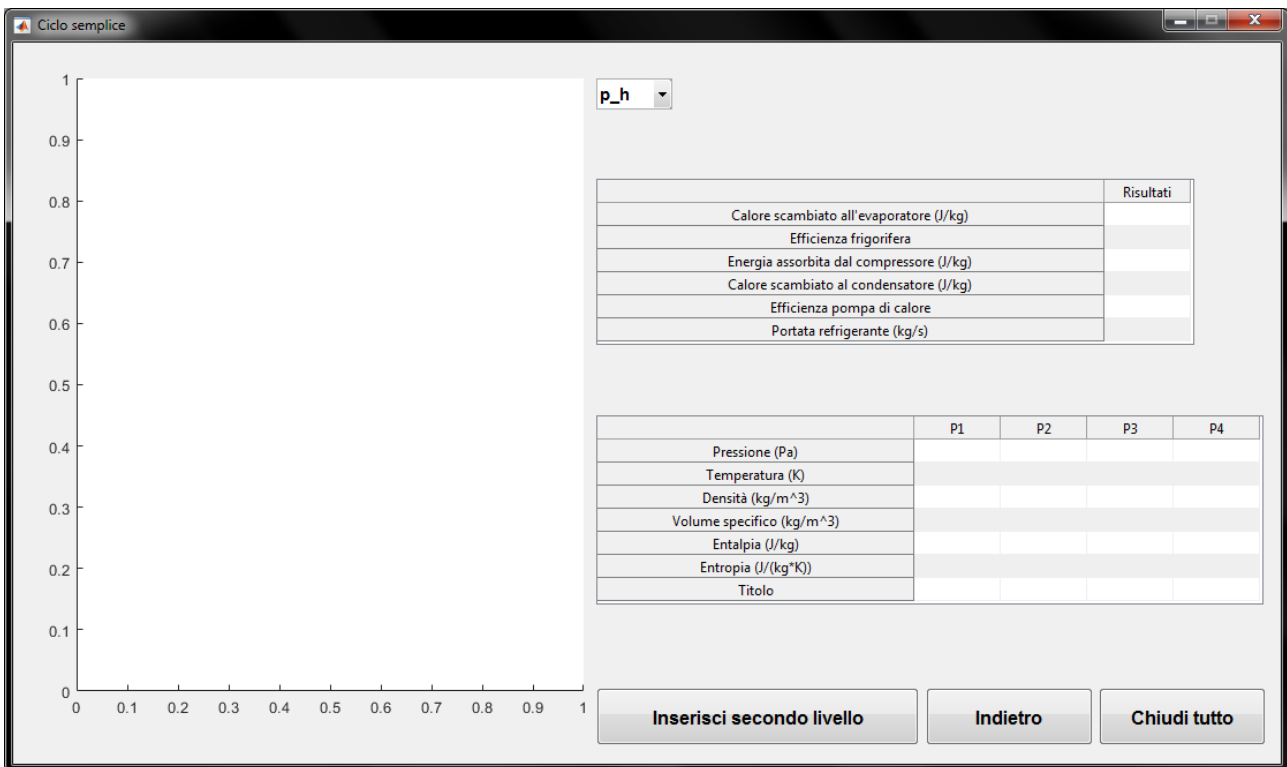
pb_chi_2=uicontrol('Parent',fig_2,'Style','pushbutton');
pb_chi_2.Units='normalized'; pb_chi_2.Position=[.857 .025 .132 .08];
pb_chi_2.FontSize=12; pb_chi_2.FontWeight='bold';
pb_chi_2.String='Chiudi tutto'; pb_chi_2.Callback=@chiudi;

```

Viene quindi innanzitutto creata la figura, denominata “fig_2”, dopodiché viene predisposto lo spazio “g_2” per inserire il grafico e viene creato il menu a tendina “pm_g_2” che servirà all’utente per selezionare che tipo di diagramma visualizzare (p-h, T-s, h-s o p-v). In seguito avviene la creazione delle tabelle “ut1_2” e “ut2_2” che raccoglieranno i risultati di output (rispettivamente i parametri del ciclo e le coordinate termodinamiche dei 4 punti).

Infine vengono creati i pulsanti “pb_ins_2” (Inserisci secondo livello), “pb_ind_2” (Indietro) e “pb_chi_2” (Chiudi tutto).

Questo è il risultato:



4.1.5fig_2a

È la figura che mostra lo schema dell'impianto frigorifero semplice.

```

fig_2a=figure; fig_2a.Visible='off';
fig_2a.Name='Schema 1 livello'; fig_2a.MenuBar='none';
fig_2a.NumberTitle='off'; fig_2a.Resize='off';

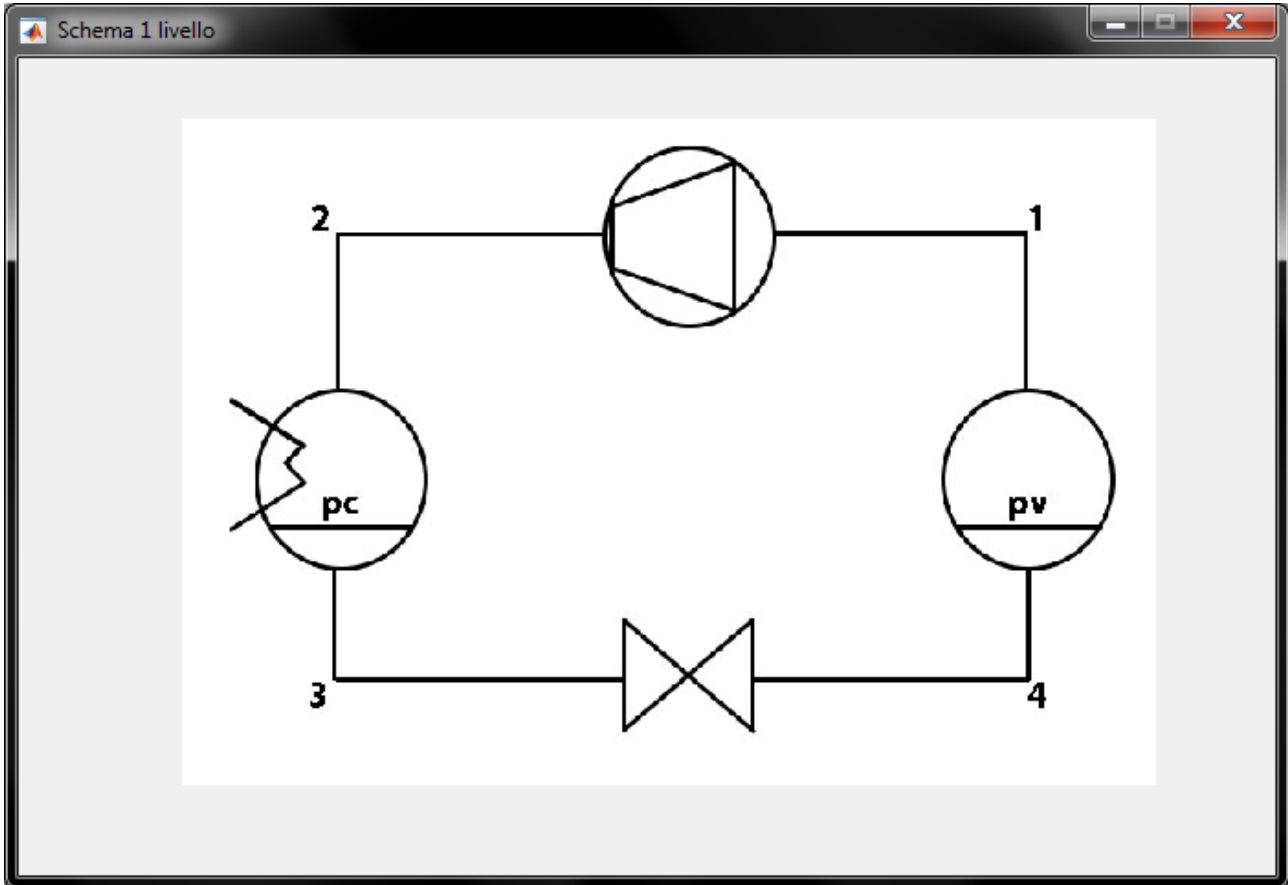
schema_1lv=imread('schema_1lv.jpg'); %crea una matrice 3d dall'immagine
[r,c,d]=size(schema_1lv); fig_2a.Position=[0 0 c r];
movegui(fig_2a,'center')

g_2a=axes(fig_2a); g_2a.Parent=fig_2a;
image(schema_1lv,'Parent',g_2a); axis(g_2a,'off')
    
```

Viene quindi innanzitutto creata la figura, denominata “fig_2a”, dopodiché viene creata la matrice 3d “schema_1lv” contenente l'immagine “schema_1lv.jpg” (contenente una rappresentazione dello schema dell'impianto); le dimensioni di tale immagine (lunghezza “r” e altezza “c”) vengono utilizzate per dimensionare la figura.

Infine all'interno della figura viene predisposto lo spazio "g_2a", nel quale viene visualizzata la matrice precedentemente creata.

Questo è il risultato:



4.1.6fig_3

È l'interfaccia che consente all'utente di inserire i parametri di input per il ciclo frigorifero a 2 livelli.

```
fig_3=figure; fig_3.Visible='off'; fig_3.Position=[0 0 460 250];
fig_3.Name='Secondo livello'; fig_3.MenuBar='none';
fig_3.NumberTitle='off'; fig_3.Resize='off'; movegui(fig_3,'center')

ip_3=uipanel('Parent',fig_3,'Title','Parametri secondo livello');
ip_3.Position=[0.05 .4 .9 .55];
ip_3.FontSize=12; ip_3.FontWeight='bold';
```

```

pm_3=uicontrol('Parent',ip_3,'Style','popupmenu');
pm_3.Units='normalized'; pm_3.Position=[.08 .7 .4 .25];
pm_3.FontSize=12; pm_3.String={'Pressione (Pa)', 'Temperatura (K)'};
pm_3.Value=pm1_1.Value; pm_3.Callback=@pop_3;

e_3=uicontrol('Parent',ip_3,'Style','edit');
e_3.Units='normalized'; e_3.Position=[.52 .7 .4 .25];
e_3.FontSize=11;

sl_3=uicontrol('Parent',ip_3,'Style','slider');
sl_3.Units='normalized'; sl_3.Position=[.25 .4 .5 .25];
sl_3.Value=0; sl_3.Callback=@slider;

t_sx_3=uicontrol('Parent',ip_3,'Style','text');
t_sx_3.Units='normalized'; t_sx_3.Position=[.05 .35 .2 .25];
t_sx_3.FontSize=11;

t_dx_3=uicontrol('Parent',ip_3,'Style','text');
t_dx_3.Units='normalized'; t_dx_3.Position=[.75 .35 .2 .25];
t_dx_3.FontSize=11;

t_dtf2_3=uicontrol('Parent',ip_3,'Style','text');
t_dtf2_3.Units='normalized'; t_dtf2_3.Position=[.05 .02 .7 .25];
t_dtf2_3.FontSize=11;
t_dtf2_3.String=[char(916), 'T secondo ambiente frigorifero (K)'];
    %char(916)->Delta maiuscola

e_dtf2_3=uicontrol('Parent',ip_3,'Style','edit');
e_dtf2_3.Units='normalized'; e_dtf2_3.Position=[.75 .05 .2 .25];
e_dtf2_3.FontSize=11;

%PULSANTI
pb_evap2_3=uicontrol('Parent',fig_3,'Style','pushbutton');
pb_evap2_3.Units='normalized'; pb_evap2_3.Position=[.05 .1 .65 .2];
pb_evap2_3.FontSize=12; pb_evap2_3.FontWeight='bold';
pb_evap2_3.String='Caratteristiche secondo evaporatore';
pb_evap2_3.Callback=@evap2;

```

```
pb_ok_3=uicontrol('Parent',fig_3,'Style','pushbutton');
pb_ok_3.Units='normalized'; pb_ok_3.Position=[.75 .21 .2 .15];
pb_ok_3.FontSize=12; pb_ok_3.FontWeight='bold';
pb_ok_3.String='Calcola'; pb_ok_3.Callback=@calcola2;

pb_ann_3=uicontrol('Parent',fig_3,'Style','pushbutton');
pb_ann_3.Units='normalized'; pb_ann_3.Position=[.75 .04 .2 .15];
pb_ann_3.FontSize=12; pb_ann_3.FontWeight='bold';
pb_ann_3.String='Annulla'; pb_ann_3.Callback=@ann_3;
```

Viene quindi innanzitutto creata la figura, denominata “fig_3”, dopodiché viene costruito il pannello “ip_3” (Parametri secondo livello), il quale contiene:

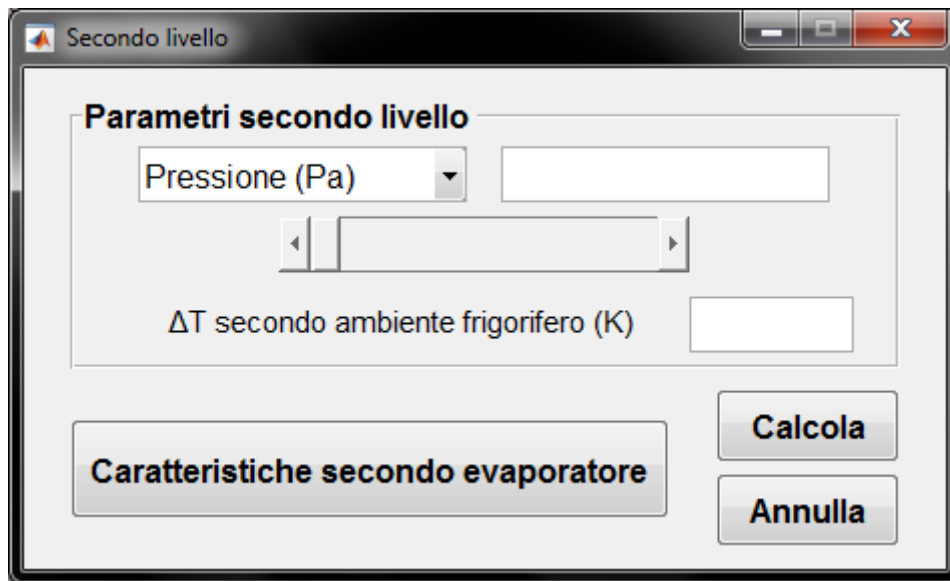
- Il menu a tendina “pm_3” (che contiene “Pressione (Pa)” e “Temperatura (K)”) e la casella editabile “e_3” per la selezione e l’inserimento della pressione/temperatura di evaporazione del secondo livello;
- Lo *slider* “sl_3” di ausilio all’inserimento dei parametri del secondo livello e le caselle di testo “t_sx_3” e “t_dx_3” che mostreranno i valori minimo e massimo della pressione/temperatura inseribile;

Nota: Di default le caselle di testo sono vuote, la loro compilazione avverrà tramite una *Callback Function*.

- La casella di testo “t_dtf2_3” e la casella editabile “e_dtf2_3” per l’inserimento del salto termico al secondo evaporatore.

Infine vengono creati i pulsanti “pb_evap2_3” (Secondo evaporatore), “pb_ok_3” (Ok) e “pb_ann_3” (Annulla).

Questo è il risultato:



4.1.7fig_3a

È l'interfaccia che consente all'utente di inserire la potenza scambiata al secondo evaporatore (direttamente o tramite l'inserimento del coefficiente globale e della superficie di scambio termico).

```
fig_3a=figure; fig_3a.Visible='off'; fig_3a.Position=[0 0 400 225];
fig_3a.Name='Caratteristiche secondo evaporatore'; fig_3a.MenuBar='none';
fig_3a.NumberTitle='off'; fig_3a.Resize='off'; movegui(fig_3a,'center')

pm_3a=uicontrol('Parent',fig_3a,'Style','popupmenu');
pm_3a.Units='normalized'; pm_3a.Position=[.3 .775 .4 .15];
pm_3a.FontSize=12; pm_3a.String={'Potenza scambiata',...
    ['Parametri ',char(945),'-S']}; %char(945)->alfa minuscola
pm_3a.Callback=@popm_3a;

ip_3a=uipanel('Parent',fig_3a);
ip_3a.Position=[.05 .3 .9 .45];

t_a_3a=uicontrol('Parent',ip_3a,'Style','text'); t_a_3a.Visible='off';
t_a_3a.Units='normalized'; t_a_3a.Position=[0 .5 .7 .3];
t_a_3a.FontSize=11; t_a_3a.String='Coeff. convezione (W/(m^2*K))';

e_a_3a=uicontrol('Parent',ip_3a,'Style','edit'); e_a_3a.Visible='off';
e_a_3a.Units='normalized'; e_a_3a.Position=[.65 .54 .3 .3];
e_a_3a.FontSize=10;

t_s_3a=uicontrol('Parent',ip_3a,'Style','text'); t_s_3a.Visible='off';
t_s_3a.Units='normalized'; t_s_3a.Position=[0 .1 .7 .3];
t_s_3a.FontSize=11; t_s_3a.String='Sup. scambio termico (m^2)';
```



```

e_s_3a=uicontrol('Parent',ip_3a,'Style','edit'); e_s_3a.Visible='off';
e_s_3a.Units='normalized'; e_s_3a.Position=[.65 .14 .3 .3];
e_s_3a.FontSize=10;

t_p_3a=uicontrol('Parent',ip_3a,'Style','text');
t_p_3a.Units='normalized'; t_p_3a.Position=[0 .4 .7 .2];
t_p_3a.FontSize=11; t_p_3a.String='Potenza scambiata (W)';

e_p_3a=uicontrol('Parent',ip_3a,'Style','edit');
e_p_3a.Units='normalized'; e_p_3a.Position=[.65 .35 .3 .3];
e_p_3a.FontSize=10;

%PULSANTI
pb_ok_3a=uicontrol('Parent',fig_3a,'Style','pushbutton');
pb_ok_3a.Units='normalized'; pb_ok_3a.Position=[.15 .1 .3 .15];
pb_ok_3a.FontSize=12; pb_ok_3a.FontWeight='bold';
pb_ok_3a.String='Ok'; pb_ok_3a.Callback=@ok_3a;

pb_ann_3a=uicontrol('Parent',fig_3a,'Style','pushbutton');
pb_ann_3a.Units='normalized'; pb_ann_3a.Position=[.55 .1 .3 .15];
pb_ann_3a.FontSize=12; pb_ann_3a.FontWeight='bold';
pb_ann_3a.String='Annulla'; pb_ann_3a.Callback=@ann_3a;

```

Viene quindi innanzitutto creata la figura, denominata “fig_3a”, dopodiché viene creato il menu a tendina “pm_1b” (che contiene “Potenza scambiata” e “Parametri α -S”) e costruito il pannello “ip_3a”, il quale contiene:

- La casella di testo “t_a_3a” e la casella editabile “e_a_3a” per l’inserimento del coefficiente globale di scambio termico;

Nota: Di default la loro visualizzazione è disattivata (verrà attivata tramite una *Callback Function*).

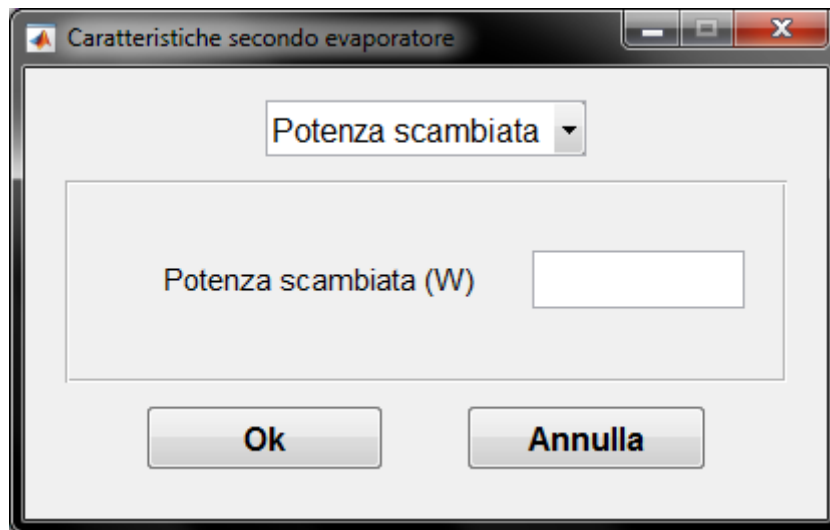
- La casella di testo “t_s_3a” e la casella editabile “e_s_3a” per l’inserimento della superficie di scambio termico;

Nota: Di default la loro visualizzazione è disattivata (verrà attivata tramite una *Callback Function*).

- La casella di testo “t_p_3a” e la casella editabile “e_p_3a” per l’inserimento diretto della potenza scambiata.

Vengono infine creati i pulsanti “pb_ok_3a” (Ok) e “pb_ann_3a” (Annulla).

Questo è il risultato:



4.1.8fig_4

È l'interfaccia che mostra i grafici e i parametri del ciclo frigorifero a due livelli, con le coordinate termodinamiche degli 8 punti fondamentali che lo caratterizzano.

```

fig_4=figure; fig_4.Visible='off'; fig_4.Position=[0 0 1100 625];
fig_4.Name='Ciclo a 2 livelli'; fig_4.MenuBar='none';
fig_4.NumberTitle='off'; fig_4.Resize='off'; movegui(fig_4,'center')

g_4=axes(fig_4); g_4.Parent=fig_4;
g_4.Units='normalized'; g_4.Position=[.05 .1 .4 .85];

pm_g_4=uicontrol('Parent',fig_4,'Style','popupmenu');
pm_g_4.Units='normalized'; pm_g_4.Position=[.46 .8 .06 .15];
pm_g_4.FontSize=11; pm_g_4.FontWeight='bold';
pm_g_4.String={'p_h','T_s','h_s','p_v'}; pm_g_4.Callback=@cambia_grafico_4;

%TABELLE RISULTATI
%energie/efficienze
utl_4=uitable('Parent',fig_4);
utl_4.FontSize=10; utl_4.ColumnName={'Risultati'};
utl_4.ColumnEditable=true;
    
```

```

%punti ciclo 1
ut2_4=uitable('Parent',fig_4);
ut2_4.FontSize=10; ut2_4.ColumnName={'P1','P2','P7','P8'};
ut2_4.ColumnEditable=true;
ut2_4.RowName={'Pressione (Pa)','Temperatura (K)','Densità (kg/m^3)',...
    'Volume specifico (kg/m^3)','Entalpia (J/kg)','Entropia (J/(kg*K))',...
    'Titolo'};
ut2_4.Units='normalized';
ut2_4.Position=[.46 .375 ut2_4.Extent(3) ut2_4.Extent(4)];

%punti ciclo 2
ut3_4=uitable('Parent',fig_4);
ut3_4.FontSize=10; ut3_4.ColumnName={'P3','P4','P5','P6'};
ut3_4.ColumnEditable=true;
ut3_4.RowName={'Pressione (Pa)','Temperatura (K)','Densità (kg/m^3)',...
    'Volume specifico (kg/m^3)','Entalpia (J/kg)','Entropia (J/(kg*K))',...
    'Titolo'};
ut3_4.Units='normalized';
ut3_4.Position=[.46 .11 ut3_4.Extent(3) ut3_4.Extent(4)];

%PULSANTI
pb_mod_4=uicontrol('Parent',fig_4,'Style','pushbutton');
pb_mod_4.Units='normalized'; pb_mod_4.Position=[.46 .025 .255 .08];
pb_mod_4.FontSize=12; pb_mod_4.FontWeight='bold';
pb_mod_4.String='Modifica secondo livello'; pb_mod_4.Callback=@mod_4;

pb_ann_4=uicontrol('Parent',fig_4,'Style','pushbutton');
pb_ann_4.Units='normalized'; pb_ann_4.Position=[.72 .025 .132 .08];
pb_ann_4.FontSize=12; pb_ann_4.FontWeight='bold';
pb_ann_4.String='Annulla'; pb_ann_4.Callback=@ann_4;

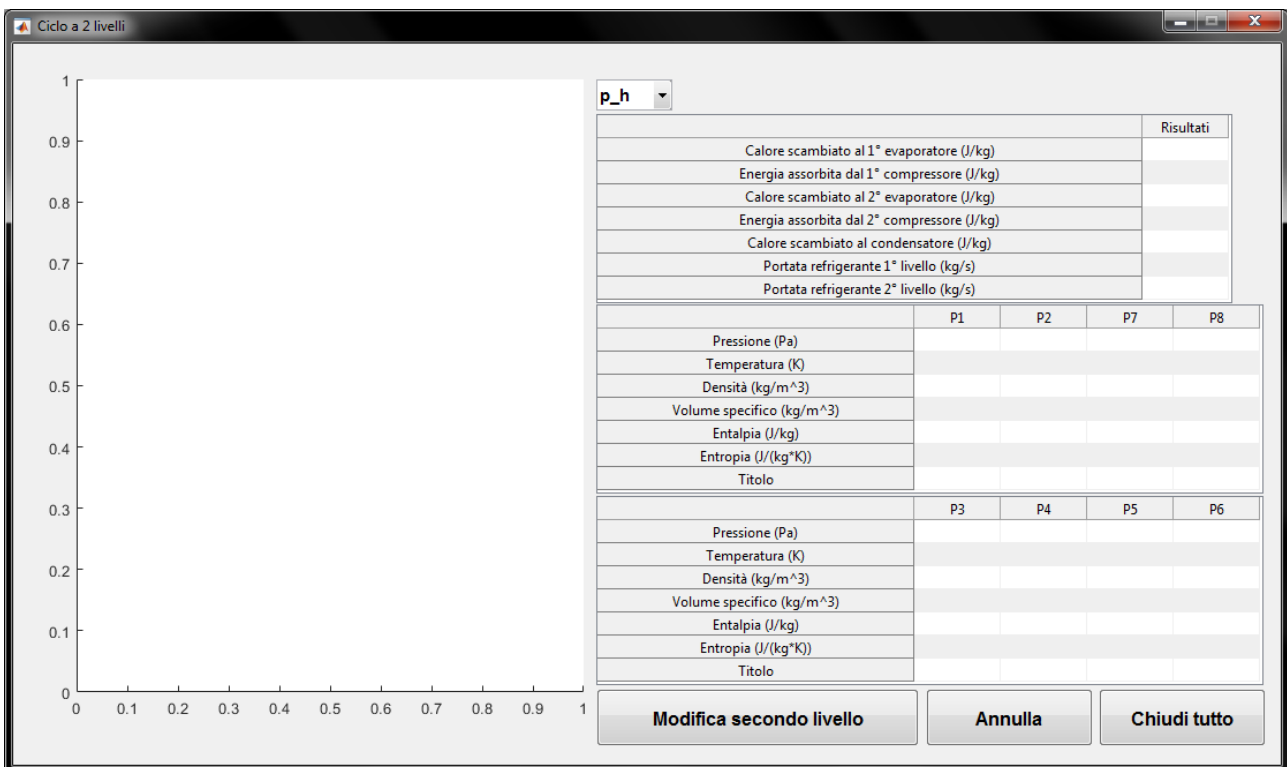
pb_chi_4=uicontrol('Parent',fig_4,'Style','pushbutton');
pb_chi_4.Units='normalized'; pb_chi_4.Position=[.857 .025 .132 .08];
pb_chi_4.FontSize=12; pb_chi_4.FontWeight='bold';
pb_chi_4.String='Chiudi tutto'; pb_chi_4.Callback=@chiudi;

```

Viene quindi innanzitutto creata la figura, denominata “fig_4”, dopodiché viene predisposto lo spazio “g_4” per inserire il grafico e viene creato il menu a tendina “pm_g_4” che servirà all’utente per selezionare che tipo di diagramma visualizzare (p-h, T-s, h-s o p-v).

In seguito avviene la creazione delle tabelle “ut1_4”, “ut2_4” e “ut3_4” che raccoglieranno i risultati di output (rispettivamente i parametri del ciclo, le coordinate termodinamiche dei 4 punti del livello inferiore e quelle dei 4 punti del livello superiore). Infine vengono creati i pulsanti “pb_mod_4” (Modifica secondo livello), “pb_ann_4” (Annulla) e “pb_chi_4” (Chiudi tutto).

Questo è il risultato:



4.1.9fig_4a

È la figura che mostra lo schema dell'impianto frigorifero a due livelli.

```
fig_4a=figure; fig_4a.Visible='off';
fig_4a.Name='Schema 2 livelli'; fig_4a.MenuBar='none';
fig_4a.NumberTitle='off'; fig_4a.Resize='off';
```

```

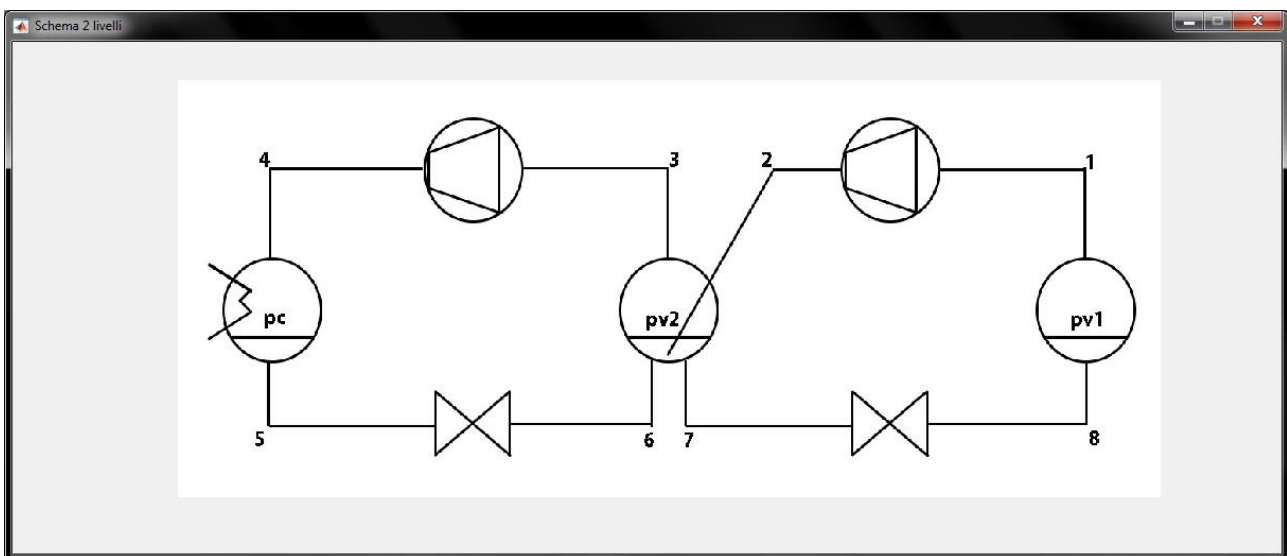
schema_2lv=imread('schema_2lv.jpg');
[r,c,d]=size(schema_2lv); fig_4a.Position=[0 0 c r];
movegui(fig_4a,'center')

g_4a=axes(fig_4a); g_4a.Parent=fig_4a;
image(schema_2lv,'Parent',g_4a); axis(g_4a,'off')
    
```

Viene quindi innanzitutto creata la figura, denominata “fig_4a”, dopodiché viene creata la matrice 3d “schema_2lv” contenente l’immagine “schema_2lv.jpg” (contenente una rappresentazione dello schema dell’impianto); le dimensioni di tale immagine (lunghezza “r” e altezza “c”) vengono utilizzate per dimensionare la figura.

Infine all’interno della figura viene predisposto lo spazio “g_4a”, nel quale viene visualizzata la matrice precedentemente creata.

Questo è il risultato:



4.2 Callback Functions

4.2.1 curve_ciclo

Questa funzione non è una vera e propria *Callback Function* in quanto viene invocata in una *Callback Function* successiva (vedi sotto-sotto-paragrafo 4.2.3.1 *calcola*); abbiamo scelto di estrarre questa funzione da quella in cui viene invocata unicamente per rendere più leggibile il codice.

Tale funzione, dopo aver determinato il fluido che si sta utilizzando (memorizzandolo nella variabile “Fluid”), calcola - mediante la funzione “PropsSI” - le coordinate termodinamiche dei 4 punti fondamentali del ciclo frigorifero semplice (prima considerando una compressione ideale e poi utilizzando il rendimento di compressione per calcolare le coordinate del punto di fine compressione reale).

Dopodiché - scelte come variabili indipendenti la pressione per le curve 1-2, 3-4 e l’entalpia per le curve 2-3, 4-1 - vengono calcolate le coordinate termodinamiche dei punti sulle varie curve.

Nota: vengono determinate prima le coordinate p ed h mancanti, poi T ed s, p e v ed infine le coordinate delle isoterme ambiente esterno e ambiente frigorifero.

I risultati vengono ordinati e memorizzati in vettori; infine tali vettori vengono posti nello *struct* Ris1, il quale a sua volta è salvato nello *struct* “Ris”, inizializzato precedentemente.

```
function Ris1=curve_ciclo(source,evendata)
    %determinazione 4 pti fondamentali
    Fluid=cell2mat(pm_fl_1.String(pm_fl_1.Value)); %nome del fluido
    coord={'P','T','Q','D','S','H'};

    for i=1:6
        %Punto 1
        if pm1_1.Value==1
            p1(i)=CoolProp.PropsSI(cell2mat(coord(i)),'P',...
                str2num(e1_1.String),'Q',1,Fluid);
        else
            p1(i)=CoolProp.PropsSI(cell2mat(coord(i)),'T',...
                str2num(e1_1.String),'Q',1,Fluid);
        end
        %Punto 3
        if pm2_1.Value==1
            p3(i)=CoolProp.PropsSI(cell2mat(coord(i)),'P',...
                str2num(e2_1.String),'Q',0,Fluid);
```

```

else
    p3(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'T', ...
        str2num(e2_1.String), 'Q', 0, Fluid);
end
end

for i=1:6
    %Punto 4
    p4(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p1(1), 'H', ...
        p3(6), Fluid);
    %Punto 2i
    p2i(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p3(1), 'S', ...
        p1(5), Fluid); %espansione ideale
end
%Punto 2
p2(6)=p1(6)+(p2i(6)-p1(6))/str2num(e_n_1.String); %h pto 2 reale
for i=1:5
    p2(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p3(1), 'H', ...
        p2(6), Fluid); %espansione reale
end

n=10; %num punti curve

%CURVE
%Variabili indipendenti
p1_2=linspace(p1(1), p3(1), n); p3_4=fliplr(p1_2);
h2_3=linspace(p2(6), p3(6), n); h4_1=linspace(p4(6), p1(6), n);
for i=1:n
    %Curva p-h
    h1_2i(i)=CoolProp.PropsSI('H', 'P', p1_2(i), 'S', p1(5), Fluid);
    h1_2(i)=p1(6)+(h1_2i(i)-p1(6))/str2num(e_n_1.String);
    p2_3(i)=p3(1);
    h3_4(i)=p3(6);
    p4_1(i)=p1(1);

```

```

%Curva T-s
s1_2i(i)=p1(5);
T1_2i(i)=CoolProp.PropsSI('T','P',p1_2(i),'S',p1(5),Fluid);
s1_2(i)=CoolProp.PropsSI('S','P',p1_2(i),'H',h1_2(i),Fluid);
T1_2(i)=CoolProp.PropsSI('T','P',p1_2(i),'H',h1_2(i),Fluid);
s2_3(i)=CoolProp.PropsSI('S','P',p3(1),'H',h2_3(i),Fluid);
T2_3(i)=CoolProp.PropsSI('T','P',p3(1),'H',h2_3(i),Fluid);
s3_4(i)=CoolProp.PropsSI('S','P',p3_4(i),'H',p3(6),Fluid);
T3_4(i)=CoolProp.PropsSI('T','P',p3_4(i),'H',p3(6),Fluid);
s4_1(i)=CoolProp.PropsSI('S','P',p1(1),'H',h4_1(i),Fluid);
T4_1(i)=CoolProp.PropsSI('T','P',p1(1),'H',h4_1(i),Fluid);

%Curva p-v
d1_2i(i)=CoolProp.PropsSI('D','P',p1_2(i),'H',h1_2i(i),Fluid);
v1_2i(i)=1/d1_2i(i);
d1_2(i)=CoolProp.PropsSI('D','P',p1_2(i),'H',h1_2(i),Fluid);
v1_2(i)=1/d1_2(i);
d2_3(i)=CoolProp.PropsSI('D','P',p3(1),'H',h2_3(i),Fluid);
v2_3(i)=1/d2_3(i);
d3_4(i)=CoolProp.PropsSI('D','P',p3_4(i),'H',h3_4(i),Fluid);
v3_4(i)=1/d3_4(i);
d4_1(i)=CoolProp.PropsSI('D','P',p1(1),'H',h4_1(i),Fluid);
v4_1(i)=1/d4_1(i);

%Isoterme amb e frigo
Tamb(i)=p3(2)-str2num(e_dta_1.String); %T3-DTamb
pamb(i)=CoolProp.PropsSI('P','T',Tamb(i),'S',s2_3(i),Fluid);
hamb(i)=CoolProp.PropsSI('H','T',Tamb(i),'S',s2_3(i),Fluid);
damb(i)=CoolProp.PropsSI('D','T',Tamb(i),'S',s2_3(i),Fluid);
Tl(i)=p1(2)+str2num(e_dtf_1.String); %T1+DTf
pf(i)=CoolProp.PropsSI('P','T',Tl(i),'Q',0,Fluid);
hf(i)=CoolProp.PropsSI('H','T',Tl(i),'S',s4_1(i),Fluid);
df(i)=CoolProp.PropsSI('D','T',Tl(i),'S',s4_1(i),Fluid);

end

```



```

vamb=1./damb; samb=s2_3;
vf=1./df; sf=s4_1;
pc=[p1_2,p2_3,p3_4,p4_1];
hc=[h1_2i,h2_3,h3_4,h4_1];
Tc=[T1_2i,T2_3,T3_4,T4_1];
sc=[s1_2i,s2_3,s3_4,s4_1];
vc=[v1_2i,v2_3,v3_4,v4_1];

%creazione struct risultati
c1={pc, hc, Tc, sc, vc, p1_2, h1_2, T1_2, s1_2, v1_2, p1, p2, p2i, p3, p4, pf, ...
    pamb, hf, hamb, Tl, Tamb, sf, samb, vf, vamb};
campil={'pc', 'hc', 'Tc', 'sc', 'vc', 'p1_2', 'h1_2', 'T1_2', 's1_2', ...
    'v1_2', 'p1', 'p2', 'p2i', 'p3', 'p4', 'pf', 'pamb', 'hf', 'hamb', ...
    'Tl', 'Tamb', 'sf', 'samb', 'vf', 'vamb'};
Ris1=cell2struct(c1', campil, 1);
save('Ris.mat', 'Ris1') %salva lo struct Ris1 nello struct Ris.mat
end

```

4.2.2curve_ciclo2

Questa funzione non è una vera e propria *Callback Function* in quanto viene invocata in una *Callback Function* successiva (vedi sotto-sotto-paragrafo 4.2.7.3 *calcola2*); abbiamo scelto di estrarre questa funzione da quella in cui viene invocata unicamente per rendere più leggibile il codice.

Tale funzione, dopo aver determinato il fluido che si sta utilizzando (memorizzandolo nella variabile “Fluid”), calcola - mediante la funzione “PropsSI” - le coordinate termodinamiche degli 8 punti fondamentali del ciclo frigorifero a 2 livelli (prima considerando le compressioni ideali e poi utilizzando il rendimento di compressione per calcolare le coordinate dei punti di fine compressione reale); dopodiché - scelte come variabili indipendenti la pressione per le curve 1-2, 3-4, 5-6, 7-8 e l’entalpia per le curve 2-7, 8-1, 4-5, 6-3 - vengono calcolate le coordinate termodinamiche dei punti sulle varie curve del 1° ciclo (a pressione/temperatura minore), caratterizzato dai punti 1, 2, 8, 1, e del 2° ciclo (a pressione/temperatura maggiore), caratterizzato dai punti 3, 4, 5, 6.

Nota: vengono determinate prima le coordinate p ed h mancanti, poi T ed s, p e v ed infine le coordinate delle isoterme ambiente esterno, ambiente frigorifero a temperatura minore e ambiente frigorifero a temperatura maggiore.

I risultati vengono ordinati e memorizzati in vettori; infine tali vettori vengono posti nello *struct* Ris2, il quale a sua volta è salvato nello *struct* “Ris”, inizializzato precedentemente.

Nota: non vengono memorizzate le coordinate termodinamiche per il plot della curva 6-3 perché si sovrappone esattamente alla curva 2-7.

```
%calcolo curve 2 livelli
function Ris2=curve_ciclo2(source,evendata)
    %determinazione 8 pti fondamentali
    Fluid=cell2mat(pm_fl_1.String(pm_fl_1.Value)); %nome del fluido
    coord={'P','T','Q','D','S','H'};
    for i=1:6
        %Punto 1
        if pm1_1.Value==1
            p1(i)=CoolProp.PropsSI(cell2mat(coord(i)),'P',...
                str2num(e1_1.String),'Q',1,Fluid);
        else
            p1(i)=CoolProp.PropsSI(cell2mat(coord(i)),'T',...
                str2num(e1_1.String),'Q',1,Fluid);
        end
        %Punto 5
        if pm2_1.Value==1
            p5(i)=CoolProp.PropsSI(cell2mat(coord(i)),'P',...
                str2num(e2_1.String),'Q',0,Fluid);
        else
            p5(i)=CoolProp.PropsSI(cell2mat(coord(i)),'T',...
                str2num(e2_1.String),'Q',0,Fluid);
        end
    end
```

```

    %Punto 3
    if pm_3.Value==1
        p3(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', ...
            str2num(e_3.String), 'Q', 1, Fluid);
    else
        p3(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'T', ...
            str2num(e_3.String), 'Q', 1, Fluid);
    end
end

for i=1:6
    %Punto 6
    p6(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p3(1), 'H', ...
        p5(6), Fluid);
    %Punto 7
    p7(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p3(1), 'Q', 0, ...
        Fluid);
end

for i=1:6
    %Punto 8
    p8(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p1(1), 'H', ...
        p7(6), Fluid);
    %Punto 2i
    p2i(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p3(1), 'S', ...
        p1(5), Fluid); %espansione ideale
    %Punto 4i
    p4i(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p5(1), 'S', ...
        p3(5), Fluid); %espansione ideale
end

%Punti 2 e 4
p2(6)=p1(6)+(p2i(6)-p1(6))/str2num(e_n_1.String); %h pto 2 reale
p4(6)=p3(6)+(p4i(6)-p3(6))/str2num(e_n_1.String); %h pto 4 reale
for i=1:5
    p2(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p3(1), 'H', ...
        p2(6), Fluid); %espansione reale
    p4(i)=CoolProp.PropsSI(cell2mat(coord(i)), 'P', p5(1), 'H', ...
        p4(6), Fluid); %espansione reale
end

```

```

n=10; %num punti curve

%CURVE
%Variabili indipendenti
p1_2=linspace(p1(1),p7(1),n); p3_4=linspace(p3(1),p5(1),n);
p5_6=fliplr(p3_4); p7_8=fliplr(p1_2);
h2_7=linspace(p2(6),p7(6),n); h8_1=linspace(p8(6),p1(6),n);
h4_5=linspace(p4(6),p5(6),n); h6_3=linspace(p6(6),p3(6),n);
for i=1:n
%Curva p-h
    %1° ciclo
    h1_2i(i)=CoolProp.PropsSI('H','P',p1_2(i),'S',p1(5),Fluid);
    h1_2(i)=p1(6)+(h1_2i(i)-p1(6))/str2num(e_n_1.String);
    p2_7(i)=p7(1);
    h7_8(i)=p7(6);
    p8_1(i)=p1(1);
    %2° ciclo
    h3_4i(i)=CoolProp.PropsSI('H','P',p3_4(i),'S',p3(5),Fluid);
    h3_4(i)=p3(6)+(h3_4i(i)-p3(6))/str2num(e_n_1.String);
    p4_5(i)=p5(1);
    h5_6(i)=p5(6);

%Curva T-s
    %1°ciclo
    s1_2i(i)=p1(5);
    T1_2i(i)=CoolProp.PropsSI('T','P',p1_2(i),'S',p1(5),Fluid);
    s1_2(i)=CoolProp.PropsSI('S','P',p1_2(i),'H',h1_2(i),Fluid);
    T1_2(i)=CoolProp.PropsSI('T','P',p1_2(i),'H',h1_2(i),Fluid);
    s2_7(i)=CoolProp.PropsSI('S','P',p7(1),'H',h2_7(i),Fluid);
    T2_7(i)=CoolProp.PropsSI('T','P',p7(1),'H',h2_7(i),Fluid);
    s7_8(i)=CoolProp.PropsSI('S','P',p7_8(i),'H',p7(6),Fluid);
    T7_8(i)=CoolProp.PropsSI('T','P',p7_8(i),'H',p7(6),Fluid);
    s8_1(i)=CoolProp.PropsSI('S','P',p1(1),'H',h8_1(i),Fluid);
    T8_1(i)=CoolProp.PropsSI('T','P',p1(1),'H',h8_1(i),Fluid);

```

```

%2° ciclo
s3_4i(i)=p3(5);
T3_4i(i)=CoolProp.PropsSI('T','P',p3_4(i),'S',p3(5),Fluid);
s3_4(i)=CoolProp.PropsSI('S','P',p3_4(i),'H',h3_4(i),Fluid);
T3_4(i)=CoolProp.PropsSI('T','P',p3_4(i),'H',h3_4(i),Fluid);
s4_5(i)=CoolProp.PropsSI('S','P',p5(1),'H',h4_5(i),Fluid);
T4_5(i)=CoolProp.PropsSI('T','P',p5(1),'H',h4_5(i),Fluid);
s5_6(i)=CoolProp.PropsSI('S','P',p5_6(i),'H',p5(6),Fluid);
T5_6(i)=CoolProp.PropsSI('T','P',p5_6(i),'H',p5(6),Fluid);
s6_3(i)=CoolProp.PropsSI('S','P',p3(1),'H',h6_3(i),Fluid); %serve
%per hf2

%Curva p-v
%1° ciclo
d1_2i(i)=CoolProp.PropsSI('D','P',p1_2(i),'H',h1_2i(i),Fluid);
v1_2i(i)=1/d1_2i(i);
d1_2(i)=CoolProp.PropsSI('D','P',p1_2(i),'H',h1_2(i),Fluid);
v1_2(i)=1/d1_2(i);
d2_7(i)=CoolProp.PropsSI('D','P',p7(1),'H',h2_7(i),Fluid);
v2_7(i)=1/d2_7(i);
d7_8(i)=CoolProp.PropsSI('D','P',p7_8(i),'H',h7_8(i),Fluid);
v7_8(i)=1/d7_8(i);
d8_1(i)=CoolProp.PropsSI('D','P',p1(1),'H',h8_1(i),Fluid);
v8_1(i)=1/d8_1(i);
%2° ciclo
d3_4i(i)=CoolProp.PropsSI('D','P',p3_4(i),'H',h3_4i(i),Fluid);
v3_4i(i)=1/d3_4i(i);
d3_4(i)=CoolProp.PropsSI('D','P',p3_4(i),'H',h3_4(i),Fluid);
v3_4(i)=1/d3_4(i);
d4_5(i)=CoolProp.PropsSI('D','P',p5(1),'H',h4_5(i),Fluid);
v4_5(i)=1/d4_5(i);
d5_6(i)=CoolProp.PropsSI('D','P',p5_6(i),'H',h5_6(i),Fluid);
v5_6(i)=1/d5_6(i);

```

```

%Isoterme amb e f
    Tamb(i)=p5(2)-str2num(e_dta_1.String); %T5-DTamb
    pamb(i)=CoolProp.PropsSI('P','T',Tamb(i),'S',s4_5(i),Fluid);
    hamb(i)=CoolProp.PropsSI('H','T',Tamb(i),'S',s4_5(i),Fluid);
    damb(i)=CoolProp.PropsSI('D','T',Tamb(i),'S',s4_5(i),Fluid);
    Tl1(i)=p1(2)+str2num(e_dtf_1.String); %T1+DTf1
    pf1(i)=CoolProp.PropsSI('P','T',Tl1(i),'Q',0,Fluid);
    hf1(i)=CoolProp.PropsSI('H','T',Tl1(i),'S',s8_1(i),Fluid);
    df1(i)=CoolProp.PropsSI('D','T',Tl1(i),'S',s8_1(i),Fluid);
    Tl2(i)=p3(2)+str2num(e_dtf2_3.String); %T3+DTf2
    pf2(i)=CoolProp.PropsSI('P','T',Tl2(i),'Q',0,Fluid);
    hf2(i)=CoolProp.PropsSI('H','T',Tl2(i),'S',s6_3(i),Fluid);
    df2(i)=CoolProp.PropsSI('D','T',Tl2(i),'S',s6_3(i),Fluid);
end

vamb=1./damb; samb=s4_5;
vf1=1./df1; sf1=s8_1;
vf2=1./df2; sf2=s6_3;
pc1=[p1_2,p2_7,p7_8,p8_1]; pc2=[p3_4,p4_5,p5_6];
hc1=[h1_2i,h2_7,h7_8,h8_1]; hc2=[h3_4i,h4_5,h5_6];
Tc1=[T1_2i,T2_7,T7_8,T8_1]; Tc2=[T3_4i,T4_5,T5_6];
sc1=[s1_2i,s2_7,s7_8,s8_1]; sc2=[s3_4i,s4_5,s5_6];
vc1=[v1_2i,v2_7,v7_8,v8_1]; vc2=[v3_4i,v4_5,v5_6];
%non memorizzo tutti i pti in un unico vettore altrimenti ci sono
%problemi nel plot

%creazione struct risultati
c2={pc1,hc1,Tc1,sc1,vc1,pc2,hc2,Tc2,sc2,vc2,p1_2,h1_2,T1_2,s1_2,...
    v1_2,p3_4,h3_4,T3_4,s3_4,v3_4,p1,p2,p2i,p7,p8,p3,p4,p4i,p5,...
    p6,pf1,pf2,pamb,hf1,hf2,hamb,Tl1,Tl2,Tamb,sf1,sf2,samb,vf1,...
    vf2,vamb};
campi2={'pc1','hc1','Tc1','sc1','vc1','pc2','hc2','Tc2','sc2',...
    'vc2','p1_2','h1_2','T1_2','s1_2','v1_2','p3_4','h3_4',...
    'T3_4','s3_4','v3_4','p1','p2','p2i','p7','p8','p3','p4',...
    'p4i','p5','p6','pf1','pf2','pamb','hf1','hf2','hamb','Tl1',...
    'Tl2','Tamb','sf1','sf2','samb','vf1','vf2','vamb'};
Ris2=cell2struct(c2',campi2,1);
save('Ris.mat','-append','Ris2') %aggiunge lo struct Ris1 a Ris.mat
end

```

4.2.3fig_1

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_1”.

4.2.3.1 calcola

Viene invocata alla pressione del pulsante “pb_cal_1”; rende visibili le finestre “fig_2” e “fig_2a”, nasconde “fig_1” e, dopo aver determinato il fluido che si sta utilizzando (memorizzandolo nella variabile “Fluid”), carica i dati (dalla cartella “Curve_limite”⁹ del programma) relativi alla curva limite di tale fluido, memorizzandoli nella variabile “CLim”; dopodiché viene invocata la funzione “curve_ciclo” - i cui risultati vengono memorizzati nello *struct* “Ris1” - e in “g_2” vengono graficate nel piano p-h (p è in scala logaritmica):

- la curva limite;
- la curva del ciclo (con compressione ideale);
- la compressione reale;
- l’isoterma ambiente esterno;
- l’isoterma ambiente frigorifero.

Di default vengono calcolate direttamente nella tabella “ut1_2”:

- l’energia scambiata all’evaporatore $Q_f = h_1 - h_4$;
- l’efficienza frigorifera $E_f = \frac{Q_f}{L_m} = \frac{h_1 - h_4}{h_2 - h_1}$;
- il lavoro assorbito dal compressore $L_m = h_2 - h_1$;
- l’energia scambiata al condensatore $Q_c = h_2 - h_3$;
- l’efficienza della pompa di calore $E_c = \frac{Q_c}{L_m} = \frac{h_2 - h_3}{h_1 - h_1} = E_f + 1$.

Mentre nella tabella “ut2_2” vengono inserite le coordinate termodinamiche dei 4 punti fondamentali del ciclo semplice.

⁹ Vedi capitolo 6 Appendice

Si controlla infine se sono state inserite (direttamente o tramite i parametri superficie e coefficiente globale di scambio termico) la potenza scambiata all'evaporatore e/o al condensatore; in caso affermativo è possibile determinare la portata \dot{m}_f di fluido circolante e la potenza assorbita dal compressore $P_m = L_m \cdot \dot{m}_f$; i risultati ottenuti sovrascrivono parte dei precedenti in "ut1_2":

- se sono state inserite entrambe le potenze scambiate o se è stata inserita solo quella scambiata all'evaporatore, quest'ultima viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri: $P_f = \alpha \cdot S \cdot (T_1 - T_l)$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_l è la temperatura dell'ambiente frigorifero).

Dopodiché viene calcolata la portata di fluido circolante $\dot{m}_f = \frac{P_f}{Q_f}$ ed infine la potenza scambiata al condensatore (anche se era stata precedentemente inserita) $P_c = Q_c \cdot \dot{m}_f$;

- se è stata inserita solo la potenza scambiata al condensatore, questa viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri: $P_c = \frac{\alpha \cdot S \cdot (\tau_2 - \tau_1)}{\log_{\tau_1} \tau_2}$,

con $\tau_1 = T_3 - T_{amb}$ e $\tau_2 = T_2 - T_{amb} = T_2 - T_3 + T_3 - T_{amb} = T_2 - T_3 + \tau_1$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{amb} è la temperatura dell'ambiente esterno).

Dopodiché viene calcolata la portata di fluido circolante $\dot{m}_f = \frac{P_f}{Q_f}$ ed infine la potenza scambiata all'evaporatore $P_f = Q_f \cdot \dot{m}_f$.


```

function calcola(source,eventdata)
    fig_2.Visible='on'; fig_2a.Visible='on';
    fig_1.Visible='off';

    Fluid=cell2mat(pm_fl_1.String(pm_fl_1.Value)); %nome del fluido
    fl_mat=[Fluid, '.mat']; %nome matrice curva limite fluido
    CLim=load(fl_mat); %carica i dati della curva limite del fluido
    Ris1=curve_ciclo;
    semilogy(g_2,CLim.hcl,CLim.pcl, 'b'), hold(g_2, 'on')
    %plot di default curva p-h
    semilogy(g_2,Ris1.hc,Ris1.pc, 'r')
    semilogy(g_2,Ris1.h1_2,Ris1.p1_2, 'r--')
    semilogy(g_2,Ris1.hamb,Ris1.pamb, 'm--')
    semilogy(g_2,Ris1.hf,Ris1.pf, 'c--'), hold(g_2, 'off')
    xlabel(g_2, 'Entalpia specifica (J/kg)');
    ylabel(g_2, 'Pressione (Pa)');
    legend(g_2, 'Curva limite', 'Curva ciclo ideale', ...
           'Compressione reale', 'Tamb esterno', 'Tamb frigorifero', ...
           'location', 'northwest')
    zoom(g_2, 'on'), grid(g_2, 'on')

    %energie/efficienze
    ut1_2.Data={Ris1.p1(6)-Ris1.p4(6), ...
               (Ris1.p1(6)-Ris1.p4(6))/(Ris1.p2(6)-Ris1.p1(6)), ...
               Ris1.p2(6)-Ris1.p1(6), Ris1.p2(6)-Ris1.p3(6), ...
               (Ris1.p2(6)-Ris1.p3(6))/(Ris1.p2(6)-Ris1.p1(6)), ''};
    %Qf=h1-h4, Ef=Qf/Lm=(h1-h4)/(h2-h1), Lm=h2-h1 Qc=h2-h3
    %Ec=Qc/Lm=(h2-h3)/(h2-h1)=Ef+1

    ut2_2.Data={Ris1.p1(1), Ris1.p2(1), Ris1.p3(1), Ris1.p4(1); ...
               Ris1.p1(2), Ris1.p2(2), Ris1.p3(2), Ris1.p4(2); Ris1.p1(4), ...
               Ris1.p2(4), Ris1.p3(4), Ris1.p4(4); 1/Ris1.p1(4), 1/Ris1.p2(4), ...
               1/Ris1.p3(4), 1/Ris1.p4(4); Ris1.p1(6), Ris1.p2(6), Ris1.p3(6), ...
               Ris1.p4(6); Ris1.p1(5), Ris1.p2(5), Ris1.p3(5), Ris1.p4(5); ...
               1, '', 0, Ris1.p4(3)};
    %p; T; D; v; h; s; Q n.b. Q(1)=1, Q(2)='', Q(3)=0

```

```

ut1_2.RowName={'Calore scambiato all''evaporatore (J/kg)',...
    'Efficienza frigorifera',...
    'Energia assorbita dal compressore (J/kg)',...
    'Calore scambiato al condensatore (J/kg)',...
    'Efficienza pompa di calore','Portata refrigerante (kg/s)'};
ut1_2.Units='normalized';
ut1_2.Position=[.46 .58 ut1_2.Extent(3) ut1_2.Extent(4)];

%potenze/portata
lpf=~(isempty(e_p_1b.String) && isempty(e_a_1b.String));
    %Pf è inserita/detrminabile
lpc=~(isempty(e_p_1a.String) && isempty(e_a_1a.String));
    %Pc è inserita/detrminabile
if lpf || lpc %se sono state inserite Pf o Pc
    ut1_2.RowName(1)={'Potenza scambiata all''evaporatore (W)'};
    ut1_2.RowName(3:4)={'Potenza assorbita dal compressore (W)',...
        'Potenza scambiata al condensatore (W)'};
    if (lpf && lpc) || (lpf && ~lpc) %(Pf e Pc) o solo Pf
        if ~isempty(e_p_1b.String) %Pf è inserita
            Pf=str2num(e_p_1b.String);
        else %Pf è detrminabile
            Pf=str2num(e_a_1b.String)*str2num(e_s_1b.String)*...
                str2num(e_dtf_1.String);%Pf=a*S*(T1-T1)
        end
        mf=Pf/cell2mat(ut1_2.Data(1)); %mf=Pf/Qf
        Pc=cell2mat(ut1_2.Data(4))*mf; %Pc=Qc*mf

    else %solo Pc
        if ~isempty(e_p_1a.String) %Pc è inserita
            Pc=str2num(e_p_1a.String);
        else %Pc è determinabile
            t1=str2num(e_dta_1.String); %t1=T3-Tamb
            t2=Ris1.p2(2)-Ris1.p3(2)+t1; %t2=T2-Tamb=T2-T3+t1
            Pc=str2num(e_a_1a.String)*str2num(e_s_1a.String)*...
                (t2-t1)/log(t2/t1); %Pc=a*S*(t2-t1)/ln(t2/t1)
        end
        mf=Pc/cell2mat(ut1_2.Data(4)); %mf=Pc/Qc
        Pf=cell2mat(ut1_2.Data(1))*mf; %Pf=Qf*mf
    end
end

```

```

Pm=cell2mat(ut1_2.Data(3))*mf; %Pm=Lm*mf

ut1_2.Data(1)={Pf}; ut1_2.Data(3:4)={Pm,Pc}';
ut1_2.Data(6)={mf};

end

ut1_2.Position=[.46 .58 ut1_2.Extent(3) ut1_2.Extent(4)];

end

```

4.2.3.2 *reset*

Viene invocata alla pressione del pulsante “pb_res_1”; elimina tutti i valori inseriti nelle caselle editabili di “fig_1”, “fig_1a” e “fig_1b”.

```

function reset(source,eventdata)
    e1_1.String=''; e2_1.String=''; e_n_1.String='';
    e_dtf_1.String=''; e_dta_1.String='';
    e_a_1a.String=''; e_s_1a.String=''; e_p_1a.String='';
    e_a_1b.String=''; e_s_1b.String=''; e_p_1b.String='';

end

```

4.2.3.3 *chiudi*

Viene invocata alla pressione del pulsante “pb_chi_1”; chiude tutte le finestre create ed elimina il file “Ris.mat”.

```

function chiudi(source,eventdata)
    close all, delete Ris.mat

end

```

Nota: Questa funzione è invocata anche dai pulsanti “pb_chi_2” di “fig_2” e “pb_chi_4” di “fig_4”.

4.2.3.4 *evap*

Viene invocata alla pressione del pulsante “pb_evap_1”; rende visibile la finestra “fig_1b”.

```
function evap(source,eventdata)
    fig_1b.Visible='on';
end
```

4.2.3.5 *cond*

Viene invocata alla pressione del pulsante “pb_cond_1”; rende visibile la finestra “fig_1a”.

```
function cond(source,eventdata)
    fig_1a.Visible='on';
end
```

4.2.4 *fig_1a*

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_1a”.

4.2.4.1 *popm_1a*

Viene invocata dal menu a tendina “pm_1a”:

- se il suo valore è 2 (è selezionata l’opzione “Parametri α -S”) vengono rese visibili le caselle di testo “t_a_1a” e “t_s_1a”, e le caselle editabili “e_a_1a” ed “e_s_1a”, mentre vengono nascoste la casella di testo “t_p_1a” e la casella editabile “e_p_1a”; inoltre viene eliminato il valore inserito nella casella editabile “e_p_1a”;
- se il suo valore è 1 (è selezionata l’opzione “Potenza scambiata”) vengono rese visibili la casella di testo “t_p_1a” e la casella editabile “e_p_1a”, mentre vengono nascoste le caselle di testo “t_a_1a” e “t_s_1a”, e le caselle editabili “e_a_1a” ed “e_s_1a”; inoltre vengono eliminati i valori inseriti nelle caselle editabili “e_a_1a” e “e_s_1a”.

Nota: L'eliminazione dei valori nelle caselle editabili che vengono nascoste previene errori di ridondanza.

```
function popm_1a(source,eventdata)
    if pm_1a.Value==2
        t_a_1a.Visible='on'; e_a_1a.Visible='on';
        t_s_1a.Visible='on'; e_s_1a.Visible='on';
        t_p_1a.Visible='off'; e_p_1a.Visible='off'; e_p_1a.String='';
    else
        t_a_1a.Visible='off'; e_a_1a.Visible='off'; e_a_1a.String='';
        t_s_1a.Visible='off'; e_s_1a.Visible='off'; e_s_1a.String='';
        t_p_1a.Visible='on'; e_p_1a.Visible='on';
    end
end
```

4.2.4.2 *ok_1a*

Viene invocata alla pressione del pulsante “pb_ok_1a”:

- se è stata compilata una sola tra le caselle editabili “e_a_1a” ed “e_s_1a” non avviene nulla;

Nota: Ciò previene errori dovuti al mancato inserimento di uno tra i parametri coefficiente globale e superficie di scambio termico.

- in tutti gli altri casi nasconde la figura “fig_1a”.

```
function ok_1a(source,eventdata)
    %se alfa è inserito ma S no (e viceversa) -> non fare niente
    if ~xor(isempty(e_a_1a.String), isempty(e_s_1a.String))
        fig_1a.Visible='off';
    end
end
```

4.2.4.3 *ann_1a*

Viene invocata alla pressione del pulsante “pb_ann_1a”; nasconde la finestra “fig_1a” ed elimina gli eventuali valori inseriti nelle caselle editabili in essa create.

```
function ann_1a(source,eventdata)
    fig_1a.Visible='off';
    e_a_1a.String=''; e_s_1a.String=''; e_p_1a.String='';
end
```

4.2.5fig_1b

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_1b”.

4.2.5.1 popm_1b

Viene invocata dal menu a tendina “pm_1b”:

- se il suo valore è 2 (è selezionata l’opzione “Parametri α -S”) vengono rese visibili le caselle di testo “t_a_1b” e “t_s_1b”, e le caselle editabili “e_a_1b” ed “e_s_1b”, mentre vengono nascoste la casella di testo “t_p_1b” e la casella editabile “e_p_1b”; inoltre viene eliminato il valore inserito nella casella editabile “e_p_1b”;
- se il suo valore è 1 (è selezionata l’opzione “Potenza scambiata”) vengono rese visibili la casella di testo “t_p_1b” e la casella editabile “e_p_1b”, mentre vengono nascoste le caselle di testo “t_a_1b” e “t_s_1b”, e le caselle editabili “e_a_1b” ed “e_s_1b”; inoltre vengono eliminati i valori inseriti nelle caselle editabili “e_a_1b” e “e_s_1b”.

Nota: L’eliminazione dei valori nelle caselle editabili che vengono nascoste previene errori di ridondanza.

```
function popm_1b(source,eventdata)
    if pm_1b.Value==2
        t_a_1b.Visible='on'; e_a_1b.Visible='on';
        t_s_1b.Visible='on'; e_s_1b.Visible='on';
        t_p_1b.Visible='off'; e_p_1b.Visible='off'; e_p_1b.String='';
    end
end
```

```
else
    t_a_1b.Visible='off'; e_a_1b.Visible='off'; e_a_1b.String='';
    t_s_1b.Visible='off'; e_s_1b.Visible='off'; e_s_1b.String='';
    t_p_1b.Visible='on'; e_p_1b.Visible='on';
end
end
```

4.2.5.2 *ok_1b*

Viene invocata alla pressione del pulsante “pb_ok_1b”:

- se è stata compilata una sola tra le caselle editabili “e_a_1b” ed “e_s_1b” non avviene nulla;

Nota: Ciò previene errori dovuti al mancato inserimento di uno tra i parametri coefficiente globale e superficie di scambio termico.

- in tutti gli altri casi nasconde la figura “fig_1b”.

```
function ok_1b(source,eventdata)
    %se alfa è inserito ma S no (e viceversa) -> non fare niente
    if ~xor(isempty(e_a_1b.String), isempty(e_s_1b.String))
        fig_1b.Visible='off';
    end
end
```

4.2.5.3 *ann_1b*

Viene invocata alla pressione del pulsante “pb_ann_1b”; nasconde la finestra “fig_1b” ed elimina gli eventuali valori inseriti nelle caselle editabili in essa create.

```
function ann_1b(source,eventdata)
    fig_1b.Visible='off';
    e_a_1b.String=''; e_s_1b.String=''; e_p_1b.String='';
end
```

4.2.6fig_2

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_2”.

4.2.6.1 cambia_grafico

Viene invocata dal menu a tendina “pm_g_2”; dopo aver determinato il fluido che si sta utilizzando (memorizzandolo nella variabile “Fluid”), carica i dati (dalla cartella “Curve_limite” del programma) relativi alla curva limite di tale fluido, memorizzandoli nella variabile “CLim”; dopodiché viene caricato lo *struct* “Ris” (contenente i risultati dei calcoli termodinamici¹⁰), memorizzandolo nell’omonima variabile.

Infine in “g_2” vengono graficate la curva limite del fluido, la curva del ciclo (con compressione ideale), la compressione reale, l’isoterma ambiente esterno e l’isoterma ambiente frigorifero:

- Se il valore di “pm_g_2” è 1 (è selezionata l’opzione “p_h”) nel piano “p-h” (p è in scala logaritmica);
- Se il valore di “pm_g_2” è 2 (è selezionata l’opzione “T_s”) nel piano “T-s”;
- Se il valore di “pm_g_2” è 3 (è selezionata l’opzione “h_s”) nel piano “h-s”;
- Se il valore di “pm_g_2” è 4 (è selezionata l’opzione “p_v”) nel piano “p-v” (v è in scala logaritmica).

```
function cambia_grafico(source,eventdata)
    Fluid=cell2mat(pm_fl_1.String(pm_fl_1.Value)); %nome del fluido
    fl_mat=[Fluid, '.mat']; %nome matrice curva limite fluido
    CLim=load(fl_mat); %carica i dati della curva limite del fluido
    Ris=load('Ris.mat'); %caricamento dati ciclo
```

¹⁰ Vedi sotto-paragrafo 4.2.1 curve_ciclo


```

if pm_g_2.Value==1 %p-h
    semilogy(g_2,CLim.hcl,CLim.pcl,'b'), hold(g_2,'on')
    semilogy(g_2,Ris.Ris1.hc,Ris.Ris1.pc,'r')
    semilogy(g_2,Ris.Ris1.h1_2,Ris.Ris1.p1_2,'r--')
    semilogy(g_2,Ris.Ris1.hamb,Ris.Ris1.pamb,'m--')
    semilogy(g_2,Ris.Ris1.hf,Ris.Ris1.pf,'c--'), hold(g_2,'off')
    xlabel(g_2,'Entalpia specifica (J/kg)');
    ylabel(g_2,'Pressione (Pa)');
    legend(g_2,'Curva limite','Curva ciclo ideale',...
        'Compressione reale','Tamb esterno','Tamb frigorifero',...
        'location','northwest')
    zoom(g_2,'on'), grid(g_2,'on')
elseif pm_g_2.Value==2 %T-s
    plot(g_2,CLim.scl,CLim.Tcl,'b'), hold(g_2,'on')
    plot(g_2,Ris.Ris1.sc,Ris.Ris1.Tc,'r')
    plot(g_2,Ris.Ris1.s1_2,Ris.Ris1.T1_2,'r--')
    plot(g_2,Ris.Ris1.samb,Ris.Ris1.Tamb,'m--')
    plot(g_2,Ris.Ris1.sf,Ris.Ris1.Tl,'c--'), hold(g_2,'off')
    xlabel(g_2,'Entropia specifica (J/(kg*K))');
    ylabel(g_2,'Temperatura (K)');
    legend(g_2,'Curva limite','Curva ciclo ideale',...
        'Compressione reale','Tamb esterno','Tamb frigorifero',...
        'location','northwest')
    zoom(g_2,'on'), grid(g_2,'on')
elseif pm_g_2.Value==3 %h-s
    plot(g_2,CLim.scl,CLim.hcl,'b'), hold(g_2,'on')
    plot(g_2,Ris.Ris1.sc,Ris.Ris1.hc,'r')
    plot(g_2,Ris.Ris1.s1_2,Ris.Ris1.h1_2,'r--')
    plot(g_2,Ris.Ris1.samb,Ris.Ris1.hamb,'m--')
    plot(g_2,Ris.Ris1.sf,Ris.Ris1.hf,'c--'), hold(g_2,'off')
    xlabel(g_2,'Entropia specifica (J/(kg*K))');
    ylabel(g_2,'Entalpia specifica (J/kg)');
    legend(g_2,'Curva limite','Curva ciclo ideale',...
        'Compressione reale','Tamb esterno','Tamb frigorifero',...
        'location','northwest')
    zoom(g_2,'on'), grid(g_2,'on')

```

```

else %p-v
    semilogx(g_2,CLim.vcl,CLim.pcl,'b'), hold(g_2,'on')
    semilogx(g_2,Ris.Ris1.vc,Ris.Ris1.pc,'r')
    semilogx(g_2,Ris.Ris1.v1_2,Ris.Ris1.p1_2,'r--')
    semilogx(g_2,Ris.Ris1.vamb,Ris.Ris1.pamb,'m--')
    semilogx(g_2,Ris.Ris1.vf,Ris.Ris1.pf,'c--'), hold(g_2,'off')
    xlabel(g_2,'Volume specifico(m^3/kg)');
    ylabel(g_2,'Pressione (Pa)');
    legend(g_2,'Curva limite','Curva ciclo ideale',...
        'Compressione reale','Tamb esterno','Tamb frigorifero')
    zoom(g_2,'on'), grid(g_2,'on')
end
end

```

4.2.6.2 *ind_2*

Viene invocata alla pressione del pulsante “pb_ind_2”; rende visibile la finestra “fig_1”, nasconde le finestre “fig_2” e “fig_2a”, e riporta il menu a tendina “pm_g_2” al valore di default (è selezionata l’opzione “p_h”).

```

function ind_2(source,eventdata)
    fig_1.Visible='on'; fig_2.Visible='off'; fig_2a.Visible='off';
    pm_g_2.Value=1;
end

```

4.2.6.3 *ins_2*

Viene invocata alla pressione dei pulsanti “pb_ins_2” e “pb_mod_4” (di “fig_4”); rende visibile la finestra “fig_3”, dopodiché viene effettuato un controllo per determinare se la funzione sia stata invocata dal pulsante “Inserisci secondo livello” (ovvero “pb_ins_2”) oppure dal pulsante “Modifica secondo livello” (ovvero “pb_mod_4”).

Nel primo caso vengono nascoste le finestre “fig_2” e “fig_2a”; per praticità il valore del menu a tendina “pm_3” viene reso uguale a quello del menu a tendina “pm1_1” (se inizialmente l’utente aveva inserito una pressione di evaporazione è probabile che inserirà una pressione anche per il secondo livello, analogamente se inizialmente aveva inserito una temperatura è molto probabile che per il secondo livello inserirà nuovamente una temperatura). Analogamente il valore della casella editabile “e_3” è reso uguale a quello inserito in “e1_1” ed il valore dello *slider* “sl_3” è riportato a quello di default.

Nelle caselle di testo “t_sx_3” e “t_dx_3” vengono mostrati i valori limite per il secondo livello (pressione/temperatura di evaporazione/condensazione a seconda del valore assunto dal menu a tendina “pm_3”), ovvero quelli calcolati in precedenza per i punti 1 e 3 e memorizzati in “Ris1” all’interno del file “Ris.mat”. Infine anche il valore inserito nella casella editabile “e_dtf2_3” è reso uguale a quello inserito nella casella editabile “e_dtf_1”.

Nel secondo caso invece vengono nascoste le finestre “fig_4” e “fig_4a”, e il menu a tendina “pm_g_4” viene riportato al valore di default.

```
function ins_2(source,eventdata)
    fig_3.Visible='on';
    if isequal(fig_2.Visible,'on') %"Inserisci II livello"
        fig_2.Visible='off'; fig_2a.Visible='off';
        pm_3.Value=pm1_1.Value; e_3.String=e1_1.String; sl_3.Value=0;
        Ris=load('Ris.mat');
        if pm_3.Value==1 %pressione
            t_sx_3.String=num2str(Ris.Ris1.p1(1));
            t_dx_3.String=num2str(Ris.Ris1.p3(1));
        else %temperatura
            t_sx_3.String=num2str(Ris.Ris1.p1(2));
            t_dx_3.String=num2str(Ris.Ris1.p3(2));
        end
        e_dtf2_3.String=e_dtf_1.String;
```

```

else %"Modifica II livello"
    fig_4.Visible='off'; fig_4a.Visible='off'; pm_g_4.Value=1;
end
end
end

```

4.2.7fig_3

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_3”.

4.2.7.1 pop_3

Viene invocata dal menu a tendina “pm_3”; serve a mostrare nelle caselle di testo “t_sx_3” e “t_dx_3” i valori limite per il secondo livello (pressione/temperatura di evaporazione/condensazione): vengono usati i valori calcolati precedentemente e salvati in “Ris1” all’interno del file “Ris.mat”.

- Se il valore di “pm_3” è 1 (ovvero se l’utente ha selezionato come input per il secondo livello la pressione di evaporazione) i valori mostrati nelle caselle di testo corrispondono alle pressioni rispettivamente dei punti 1 e 3;
- Altrimenti (ovvero se l’utente ha selezionato come input per il secondo livello la temperatura di evaporazione) i valori mostrati nelle caselle di testo corrispondono alle temperature rispettivamente dei punti 1 e 3.

Infine il valore mostrato nella casella editabile “e_3” è calcolato tramite l’espressione $min + (max - min) \cdot x$, dove *min* è il valore minimo (memorizzato in “t_sx_3”), *max* è il valore massimo (memorizzato in “t_dx_3”) ed *x* è il valore (compreso fra 0 e 1) dato dalla posizione dello slider.

```

function pop_3(source,eventdata)
Ris=load('Ris.mat');
    if pm_3.Value==1 %pressione
        t_sx_3.String=num2str(Ris.Ris1.p1(1));
        t_dx_3.String=num2str(Ris.Ris1.p3(1));
    end
end

```

```

else %temperatura
    t_sx_3.String=num2str(Ris.Ris1.p1(2));
    t_dx_3.String=num2str(Ris.Ris1.p3(2));
end
e_3.String=num2str(str2num(t_sx_3.String)+(str2num(t_dx_3.String)-...
    str2num(t_sx_3.String))*sl_3.Value); %min+(max-min)*x
end

```

4.2.7.2 slider

Viene invocata dallo *slider* “sl_3”; aggiorna il valore mostrato nella casella editabile “e_3”, calcolandolo tramite l’espressione $min + (max - min) \cdot x$, dove min è il valore minimo (mostrato nella casella di testo “t_sx_3”), max è il valore massimo (mostrato nella casella di testo “t_dx_3”) ed x è il valore (compreso fra 0 e 1) dato dalla posizione dello *slider*.

```

function slider(source,eventdata)
e_3.String=num2str(str2num(t_sx_3.String)+(str2num(t_dx_3.String)-...
    str2num(t_sx_3.String))*sl_3.Value); %min+(max-min)*x
end

```

4.2.7.3 calcola2

Viene invocata alla pressione del pulsante “pb_cal_3”; rende visibili le finestre “fig_4” e “fig_4a”, nasconde “fig_3” e - dopo aver determinato il fluido che si sta utilizzando (memorizzandolo nella variabile “Fluid”) - carica i dati (dalla cartella “Curve_limite” del programma) relativi alla curva limite di tale fluido, memorizzandoli nella variabile “CLim”; dopodiché viene invocata la funzione “curve_ciclo2”, i cui risultati vengono memorizzati nello *struct* “Ris2” e in “g_4” vengono graficate nel piano p-h (p è in scala logaritmica):

- la curva limite;
- la curva del ciclo (con compressioni ideali);
- le compressioni reali;
- l’isoterma ambiente esterno;

- le isoterme ambienti frigoriferi.

Di default vengono calcolate direttamente nella tabella “ut1_4”:

- l’energia scambiata al primo evaporatore $Q_{f1} = h_1 - h_8$;
- il lavoro assorbito dal primo compressore $L_{m1} = h_2 - h_1$;
- il lavoro assorbito dal secondo compressore $L_{m2} = h_4 - h_3$;

Nota: l’energia scambiata al secondo evaporatore non è calcolabile perché viene attraversato da 2 portate diverse di fluido, quindi dal bilancio delle potenze entranti/uscenti non è possibile passare al bilancio delle energie.

- l’energia scambiata al condensatore $Q_c = h_4 - h_5$.

Mentre nella tabella “ut2_4” vengono inserite le coordinate termodinamiche dei 4 punti fondamentali del ciclo a pressione/temperatura minore (punti 1, 2, 7 e 8) e nella tabella “ut3_4” vengono inserite le coordinate termodinamiche dei 4 punti fondamentali del ciclo a pressione/temperatura maggiore (punti 3, 4, 5 e 6).

Si controlla infine se sono state inserite (direttamente o tramite i parametri superficie e coefficiente globale di scambio termico) le potenze scambiate agli evaporatori e/o al condensatore; in caso affermativo:

- Se ne sono state inserite almeno 2 è possibile determinare le potenze assorbite dai compressori P_{m1} e P_{m2} ed entrambe le portate circolanti \dot{m}_{f1} e \dot{m}_{f2} ;
- Se sono state inserite la potenza scambiata al primo evaporatore e quella scambiata al condensatore, quest’ultima viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri:

$$P_c = \frac{\alpha \cdot S \cdot (\tau_2 - \tau_1)}{\log_{\tau_1} \tau_2}, \text{ con } \tau_1 = T_5 - T_{amb} \text{ e}$$

$$\tau_2 = T_4 - T_{amb} = T_4 - T_5 + T_5 - T_{amb} =$$

$= T_4 - T_5 + \tau_1$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{amb} è la temperatura dell'ambiente esterno); dopodiché vengono calcolate le portate di fluido circolante al livello superiore $\dot{m}_{f2} = \frac{P_c}{Q_c}$ ed al livello inferiore $\dot{m}_{f1} = \frac{P_{f1}}{Q_{f1}}$, le potenze assorbite dai compressori $P_{m1} = L_{m1} \cdot \dot{m}_{f1}$, $P_{m2} = L_{m2} \cdot \dot{m}_{f2}$ ed infine la potenza scambiata al secondo condensatore (anche se era stata precedentemente inserita)

$$P_{f2} = P_c - P_{f1} - P_{m1} - P_{m2};$$

Nota: La potenza scambiata al primo evaporatore P_{f1} viene semplicemente presa dal calcolo precedentemente eseguito per il ciclo semplice.

- Se sono state inserite la potenza scambiata al secondo evaporatore e quella scambiata al condensatore, quest'ultima viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri:

$$P_c = \frac{\alpha \cdot S \cdot (\tau_2 - \tau_1)}{\log \frac{\tau_2}{\tau_1}}, \text{ con } \tau_1 = T_5 - T_{amb} \text{ e}$$

$$\tau_2 = T_4 - T_{amb} = T_4 - T_5 + T_5 - T_{amb} =$$

$= T_4 - T_5 + \tau_1$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{amb} è la temperatura dell'ambiente esterno); dopodiché viene calcolata la portata di fluido circolante al livello

superiore $\dot{m}_{f2} = \frac{P_c}{h_4 - h_6}$ e la potenza assorbita dal relativo compressore $P_{m2} = L_{m2} \cdot \dot{m}_{f2}$.

Poi la potenza scambiata al secondo evaporatore viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri: $P_{f2} = \alpha \cdot S \cdot (T_{l2} - T_7)$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{l2} è la temperatura del secondo ambiente frigorifero); dopodiché viene calcolata la portata di fluido circolante al livello inferiore

$\dot{m}_{f1} = \frac{P_c - P_{f2} - P_{m2}}{h_2 - h_8}$ e la potenza assorbita dal relativo compressore $P_{m1} = L_{m1} \cdot \dot{m}_{f1}$;

- Se sono state inserite le potenze scambiate agli evaporatori, vengono calcolate dapprima la portata di fluido circolante al livello inferiore $\dot{m}_{f1} = \frac{P_{f1}}{Q_{f1}}$ e la potenza assorbita dal relativo compressore $P_{m1} = L_{m1} \cdot \dot{m}_{f1}$; dopodiché la potenza scambiata al secondo evaporatore viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri: $P_{f2} = \alpha \cdot S \cdot (T_{l2} - T_7)$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{l2} è la temperatura del secondo ambiente frigorifero), quindi si calcolano la portata di fluido circolante al livello superiore $\dot{m}_{f2} = \frac{P_{f2} + P_{f1} + P_{m1}}{h_3 - h_6}$, la potenza scambiata al condensatore $P_c = \dot{m}_{f2} \cdot (h_4 - h_5)$ e la potenza assorbita dal secondo compressore $P_{m2} = L_{m2} \cdot \dot{m}_{f2}$.

Nota: La potenza scambiata al primo evaporatore P_{f1} viene semplicemente presa dal calcolo precedentemente eseguito per il ciclo semplice.

- Se è stata inserita solo la potenza scambiata al primo evaporatore, questa viene semplicemente presa dal calcolo precedentemente eseguito per il ciclo semplice; dopodiché vengono calcolate la portata di fluido circolante al livello inferiore $\dot{m}_{f1} = \frac{P_{f1}}{Q_{f1}}$ e la potenza assorbita dal relativo compressore $P_{m1} = L_{m1} \cdot \dot{m}_{f1}$;
- Se è stata inserita solo la potenza scambiata al condensatore, questa viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri: $P_c = \frac{\alpha \cdot S \cdot (\tau_2 - \tau_1)}{\log \frac{\tau_2}{\tau_1}}$, con $\tau_1 = T_5 - T_{amb}$ e $\tau_2 = T_4 - T_{amb} = T_4 - T_5 + T_5 - T_{amb} = T_4 - T_5 + \tau_1$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{amb} è la temperatura dell'ambiente esterno); dopodiché viene calcolata la portata di fluido circolante al livello superiore $\dot{m}_{f2} = \frac{P_c}{h_4 - h_6}$ e la potenza assorbita dal relativo compressore $P_{m2} = L_{m2} \cdot \dot{m}_{f2}$;
- Se è stata inserita solo la potenza scambiata al secondo evaporatore, questa viene o letta (se è stata inserita direttamente) o calcolata (se sono stati inseriti i parametri: $P_{f2} = \alpha \cdot S \cdot (T_{l2} - T_7)$, dove α è il coefficiente globale di scambio termico, S è la superficie dello scambiatore e T_{l2} è la temperatura del secondo ambiente frigorifero).

Nota: I risultati ottenuti sovrascrivono sempre parte dei precedenti in “ut1_4”.

```

function calcola2(source,eventdata)
    fig_4.Visible='on';    fig_4a.Visible='on';
    fig_3.Visible='off';

    Fluid=cell2mat(pm_fl_1.String(pm_fl_1.Value)); %nome del fluido
    fl_mat=[Fluid, '.mat']; %nome matrice curva limite fluido
    CLim=load(fl_mat); %carica i dati della curva limite del fluido
    Ris2=curve_ciclo2;
    %plot di default curva p-h
    semilogy(g_4,CLim.hc1,CLim.pc1, 'b'), hold(g_4, 'on')
    semilogy(g_4,Ris2.hc1,Ris2.pc1, 'r')
    semilogy(g_4,Ris2.h1_2,Ris2.p1_2, 'r--')
    semilogy(g_4,Ris2.hamb,Ris2.pamb, 'm--')
    semilogy(g_4,Ris2.hf1,Ris2.pf1, 'c--')
    semilogy(g_4,Ris2.hc2,Ris2.pc2, 'r')
    semilogy(g_4,Ris2.h3_4,Ris2.p3_4, 'r--')
    semilogy(g_4,Ris2.hf2,Ris2.pf2, 'c--'), hold(g_4, 'off')
    xlabel(g_4, 'Entalpia specifica (J/kg)');
    ylabel(g_4, 'Pressione (Pa)');
    legend(g_4, 'Curva limite', 'Curva ciclo ideale', ...
        'Compressione reale', 'Tamb esterno', 'Tamb frigorifero', ...
        'location', 'northwest')
    zoom(g_4, 'on'), grid(g_4, 'on')

    %energie/efficienze
    ut1_4.Data={Ris2.p1(6)-Ris2.p8(6),Ris2.p2(6)-Ris2.p1(6), ...
        'N/D',Ris2.p4(6)-Ris2.p3(6),Ris2.p4(6)-Ris2.p5(6), '', ''}';
    %Qf1=h1-h8, Lm1=h2-h1, Qf2, Lm2=h4-h3, Qc=h4-h5

    %ciclo1 (1,2,7,8)
    ut2_4.Data={Ris2.p1(1),Ris2.p2(1),Ris2.p7(1),Ris2.p8(1); ...
        Ris2.p1(2),Ris2.p2(2),Ris2.p7(2),Ris2.p8(2);Ris2.p1(4), ...
        Ris2.p2(4),Ris2.p7(4),Ris2.p8(4);1/Ris2.p1(4),1/Ris2.p2(4), ...
        1/Ris2.p7(4),1/Ris2.p8(4);Ris2.p1(6),Ris2.p2(6),Ris2.p7(6), ...
        Ris2.p8(6);Ris2.p1(5),Ris2.p2(5),Ris2.p7(5),Ris2.p8(5); ...
        1, '', 0,Ris2.p8(3)};
    %p; T; D; v; h; s; Q n.b. Q(1)=1, Q(7)=0, Q(2)=''

```

```

%ciclo2 (3,4,5,6)
ut3_4.Data={Ris2.p3(1),Ris2.p4(1),Ris2.p5(1),Ris2.p6(1);...
    Ris2.p3(2),Ris2.p4(2),Ris2.p5(2),Ris2.p6(2);Ris2.p3(4),...
    Ris2.p4(4),Ris2.p5(4),Ris2.p6(4);1/Ris2.p3(4),1/Ris2.p4(4),...
    1/Ris2.p5(4),1/Ris2.p6(4);Ris2.p3(6),Ris2.p4(6),Ris2.p5(6),...
    Ris2.p6(6);Ris2.p3(5),Ris2.p4(5),Ris2.p5(5),Ris2.p6(5);...
    1, '', 0, Ris2.p6(3)};
%p; T; D; v; h; s; Q n.b. Q(3)=1, Q(5)=0, Q(4)=''

ut1_4.RowName={'Calore scambiato al 1° evaporatore (J/kg)',...
    'Energia assorbita dal 1° compressore (J/kg)',...
    'Calore scambiato al 2° evaporatore (J/kg)',...
    'Energia assorbita dal 2° compressore (J/kg)',...
    'Calore scambiato al condensatore (J/kg)',...
    'Portata refrigerante 1° livello (kg/s)',...
    'Portata refrigerante 2° livello (kg/s)'};
ut1_4.Units='normalized';
ut1_4.Position=[.46 .64 ut1_4.Extent(3) ut1_4.Extent(4)];

%potenze/portate
lpf1=~(isempty(e_p_1b.String) && isempty(e_a_1b.String));
    %Pf1 è inserita/determinabile
lpf2=~(isempty(e_p_3a.String) && isempty(e_a_3a.String));
    %Pf2 è inserita/determinabile
lpc=~(isempty(e_p_1a.String) && isempty(e_a_1a.String));
    %Pc è inserita/determinabile
if (lpc && (lpf1 || lpf2)) || (lpf1 && lpf2) %almeno 2
    %ciclo 1 e 2
    ut1_4.RowName(1:5)={...
        'Potenza scambiata al 1° evaporatore (W)',...
        'Potenza assorbita dal 1° compressore (W)',...
        'Potenza scambiata al 2° evaporatore (W)',...
        'Potenza assorbita dal 2° compressore (W)',...
        'Potenza scambiata al condensatore (W)'};
    if (lpc && lpf1 && lpf2) || (lpc && lpf1) %tutte o (Pc e Pf1)
        %Pc
        if ~isempty(e_p_1a.String) %Pc è inserita
            Pc=str2num(e_p_1a.String);
        end
    end
end

```

```

else %Pc è determinabile
    t1=str2num(e_dta_1.String); %t1=T5-Tamb
    t2=Ris2.p4(2)-Ris2.p5(2)+t1; %t2=T4-Tamb=T4-T5+t1
    Pc=str2num(e_a_1a.String)*str2num(e_s_1a.String)*...
        (t2-t1)/log(t2/t1); %Pc=a*S*(t2-t1)/ln(t2/t1)
end
mf2=Pc/(Ris2.p4(6)-Ris2.p6(6)); %mf2=Pc/(h4-h6)
Pf1=cell2mat(ut1_2.Data(1)); %Pf1 (già calcolata)
mf1=Pf1/cell2mat(ut1_4.Data(1)); %mf1=Pf1/Qf1
Pm1=cell2mat(ut1_4.Data(2))*mf1; %Pm1=Lm1*mf1
Pm2=cell2mat(ut1_4.Data(4))*mf2; %Pm2=Lm2*mf2
Pf2=Pc-Pf1-Pm1-Pm2; %Pf2=Pc-Pf1-Pm1-Pm2
elseif lpc && lpf2 %Pc e Pf2
    %Pc
    if ~isempty(e_p_1a.String) %Pc è inserita
        Pc=str2num(e_p_1a.String);
    else %Pc è determinabile
        t1=str2num(e_dta_1.String); %t1=T5-Tamb
        t2=Ris2.p4(2)-Ris2.p5(2)+t1; %t2=T4-Tamb=T4-T5+t1
        Pc=str2num(e_a_1a.String)*str2num(e_s_1a.String)*...
            (t2-t1)/log(t2/t1); %Pc=a*S*(t2-t1)/ln(t2/t1)
    end
    mf2=Pc/(Ris2.p4(6)-Ris2.p6(6)); %mf2=Pc/(h4-h6)
    Pm2=cell2mat(ut1_4.Data(4))*mf2; %Pm2=Lm2*mf2
    %Pf2
    if ~isempty(e_p_3a.String) %Pf2 inserita
        Pf2=str2num(e_p_3a.String);
    elseif ~isempty(e_a_3a.String) %Pf2 determinabile
        Pf2=str2num(e_a_3a.String)*str2num(e_s_3a.String)*...
            str2num(e_dtf2_3.String); %Pf2=a*S*(T12-T7)
    end
    mf1=(Pc-Pf2-Pm2)/(Ris2.p2(6)-Ris2.p8(6));
        %mf1=(Pc-Pf2-Pm2)/(h2-h8)
    Pf1=mf1*cell2mat(ut1_4.Data(1)); %Pf1=mf1*Qf1
    Pm1=cell2mat(ut1_4.Data(2))*mf1; %Pm1=Lm1*mf1
else %Pf1 e Pf2
    Pf1=cell2mat(ut1_2.Data(1)); %Pf1 (già calcolata)
    mf1=Pf1/cell2mat(ut1_4.Data(1)); %mf1=Pf1/Qf1
    Pm1=cell2mat(ut1_4.Data(2))*mf1; %Pm1=Lm1*mf1

```

```

    %Pf2
    if ~isempty(e_p_3a.String) %Pf2 inserita
        Pf2=str2num(e_p_3a.String);
    elseif ~isempty(e_a_3a.String) %Pf2 determinabile
        Pf2=str2num(e_a_3a.String)*str2num(e_s_3a.String)*...
            str2num(e_dtf2_3.String); %Pf2=a*S*(T12-T7)
    end
    mf2=(Pf2+Pf1+Pm1)/(Ris2.p3(6)-Ris2.p6(6));
    %mf2=(Pf2+Pf1+Pm1)/(h3-h6)
    Pc=mf2*(Ris2.p4(6)-Ris2.p5(6)); %Pc=mf2*(h4-h5)
    Pm2=cell2mat(ut1_4.Data(4))*mf2; %Pm2=Lm2*mf2
end

    ut1_4.Data={Pf1,Pm1,Pf2,Pm2,Pc,mf1,mf2}';
elseif lpf1 && ~lpc && ~lpf2 %solo Pf1
    %Pf1,mf1,Pm1
    ut1_4.RowName(1:2)={...
        'Potenza scambiata al 1° evaporatore (W)',...
        'Potenza assorbita dal 1° compressore (W)'}';
    Pf1=cell2mat(ut1_2.Data(1)); %Pf1 (già calcolata)
    mf1=Pf1/cell2mat(ut1_4.Data(1)); %mf1=Pf1/Qf1
    Pm1=cell2mat(ut1_4.Data(2))*mf1; %Pm1=Lm1*mf1

    ut1_4.Data(1:2)={Pf1,Pm1}'; ut1_4.Data(6)={mf1}';
elseif lpc && ~lpf1 && ~lpf2 %solo Pc
    %Pc,mf2,Pm2
    ut1_4.RowName(4:5)={...
        'Potenza assorbita dal 2° compressore (W)',...
        'Potenza scambiata al condensatore (W)'}';
    %Pc
    if ~isempty(e_p_1a.String) %Pc è inserita
        Pc=str2num(e_p_1a.String);
    else %Pc è determinabile
        t1=str2num(e_dta_1.String); %t1=T5-Tamb
        t2=Ris2.p4(2)-Ris2.p5(2)+t1; %t2=T4-Tamb=T4-T5+t1
        Pc=str2num(e_a_1a.String)*str2num(e_s_1a.String)*...
            (t2-t1)/log(t2/t1); %Pc=a*S*(t2-t1)/ln(t2/t1)
    end
end

```

```

mf2=Pc/(Ris2.p4(6)-Ris2.p6(6)); %mf2=Pc/(h4-h6)
Pm2=cell2mat(ut1_4.Data(4))*mf2; %Pm2=Lm2*mf2

ut1_4.Data(4:5)={Pm2,Pc}'; ut1_4.Data(7)={mf2};
elseif ~lpc && ~lpf1 && lpf2 %solo Pf2
    ut1_4.RowName(3)={'Potenza scambiata al 2° evaporatore (W)'};
    %Pf2
    if ~isempty(e_p_3a.String) %Pf2 inserita
        Pf2=str2num(e_p_3a.String);
    elseif ~isempty(e_a_3a.String) %Pf2 determinabile
        Pf2=str2num(e_a_3a.String)*str2num(e_s_3a.String)*...
            str2num(e_dtf2_3.String); %Pf2=a*S*(T12-T7)
    end
    ut1_4.Data(3)={Pf2};
end
ut1_4.Position=[.46 .64 ut1_4.Extent(3) ut1_4.Extent(4)];
end

```

4.2.7.4 *ann_3*

Viene invocata alla pressione del pulsante “pb_ann_3”; rende visibile la finestra “fig_2” e nasconde “fig_3”.

Dopodiché vengono eliminati gli eventuali valori inseriti nelle caselle editabili “e_a_3a”, “e_s_3a” e “e_p_3a” di “fig_3a”.

```

function ann_3(source,eventdata)
    fig_2.Visible='on'; fig_3.Visible='off';
    e_a_3a.String=''; e_s_3a.String=''; e_p_3a.String='';
end

```

4.2.7.5 *evap2*

Viene invocata alla pressione del pulsante “pb_evap2_3”; rende visibile la finestra “fig_3a”.

```

function evap2(source,eventdata)
    fig_3a.Visible='on';
end

```

4.2.8fig_3a

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_3a”.

4.2.8.1 popm_3a

Viene invocata dal menu a tendina “pm_3a”:

- se il suo valore è 2 (è selezionata l’opzione “Parametri α -S”) vengono rese visibili le caselle di testo “t_a_3a” e “t_s_3a”, e le caselle editabili “e_a_3a” ed “e_s_3a”, mentre vengono nascoste la casella di testo “t_p_3a” e la casella editabile “e_p_3a”; inoltre viene eliminato il valore inserito nella casella editabile “e_p_3a”;
- se il suo valore è 1 (è selezionata l’opzione “Potenza scambiata”) vengono rese visibili la casella di testo “t_p_3a” e la casella editabile “e_p_3a”, mentre vengono nascoste le caselle di testo “t_a_3a” e “t_s_3a”, e le caselle editabili “e_a_3a” ed “e_s_3a”; inoltre vengono eliminati i valori inseriti nelle caselle editabili “e_a_3a” ed “e_s_3a”.

Nota: L’eliminazione dei valori nelle caselle editabili che vengono nascoste previene errori di ridondanza.

```
function popm_3a(source,eventdata)
    if pm_3a.Value==2
        t_a_3a.Visible='on'; e_a_3a.Visible='on';
        t_s_3a.Visible='on'; e_s_3a.Visible='on';
        t_p_3a.Visible='off'; e_p_3a.Visible='off'; e_p_3a.String='';
    else
        t_a_3a.Visible='off'; e_a_3a.Visible='off'; e_a_3a.String='';
        t_s_3a.Visible='off'; e_s_3a.Visible='off'; e_s_3a.String='';
        t_p_3a.Visible='on'; e_p_3a.Visible='on';
    end
end
```

4.2.8.2 ok_3a

Viene invocata alla pressione del pulsante “pb_ok_3a”:

- se è stata compilata una sola tra le caselle editabili “e_a_3a” ed “e_s_3a” non avviene nulla;

Nota: Ciò previene errori dovuti al mancato inserimento di uno tra i parametri coefficiente globale e superficie di scambio termico.

- in tutti gli altri casi nasconde la figura “fig_3a”.

```
function ok_3a(source,eventdata)
    %se alfa è inserito ma S no (e viceversa) -> non fare niente
    if ~xor(isempty(e_a_3a.String), isempty(e_s_3a.String))
        fig_3a.Visible='off';
    end
end
```

4.2.8.3 ann_3a

Viene invocata alla pressione del pulsante “pb_ann_3a”; nasconde la finestra “fig_3a” ed elimina gli eventuali valori inseriti nelle caselle editabili in essa create.

```
function ann_3a(source,eventdata)
    fig_3a.Visible='off';
    e_a_3a.String=''; e_s_3a.String=''; e_p_3a.String='';
end
```

4.2.9fig_4

Verranno di seguito analizzate le *Callback Functions* invocate dagli oggetti creati in “fig_4”.

4.2.9.1 cambia_grafico_4

Viene invocata dal menu a tendina “pm_g_4”; dopo aver determinato il fluido che si sta utilizzando (memorizzandolo nella variabile “Fluid”), carica i dati (dalla cartella “Curve_limite” del programma) relativi alla curva limite di tale fluido, memorizzandoli nella variabile “CLim”; dopodiché viene caricato lo *struct* “Ris” (contenente i risultati dei calcoli termodinamici¹¹), memorizzandolo nell’omonima variabile.

Infine in “g_4” vengono graficate la curva limite del fluido, la curva del ciclo (con compressioni ideali), le compressioni reali, l’isoterma ambiente esterno e le isoterme ambienti frigoriferi:

- Se il valore di “pm_g_4” è 1 (è selezionata l’opzione “p_h”) nel piano “p-h” (p è in scala logaritmica);
- Se il valore di “pm_g_4” è 2 (è selezionata l’opzione “T_s”) nel piano “T-s”;
- Se il valore di “pm_g_4” è 3 (è selezionata l’opzione “h_s”) nel piano “h-s”;
- Se il valore di “pm_g_4” è 4 (è selezionata l’opzione “p_v”) nel piano “p-v” (v è in scala logaritmica).

```
function cambia_grafico_4(source,eventdata)
    Fluid=cell2mat(pm_fl_1.String(pm_fl_1.Value)); %nome del fluido
    fl_mat=[Fluid, '.mat']; %nome matrice curva limite fluido
    CLim=load(fl_mat); %carica i dati della curva limite del fluido
    Ris=load('Ris.mat'); %caricamento dati ciclo
```

¹¹ Vedi sotto-paragrafo 4.2.2 curve_ciclo2

```

if pm_g_4.Value==1 %p-h
    semilogy(g_4,CLim.hc1,CLim.pcl,'b'), hold(g_4,'on')
    semilogy(g_4,Ris.Ris2.hc1,Ris.Ris2.pcl,'r')
    semilogy(g_4,Ris.Ris2.h1_2,Ris.Ris2.p1_2,'r--')
    semilogy(g_4,Ris.Ris2.hamb,Ris.Ris2.pamb,'m--')
    semilogy(g_4,Ris.Ris2.hf1,Ris.Ris2.pf1,'c--')
    semilogy(g_4,Ris.Ris2.hc2,Ris.Ris2.pc2,'r')
    semilogy(g_4,Ris.Ris2.h3_4,Ris.Ris2.p3_4,'r--')
    semilogy(g_4,Ris.Ris2.hf2,Ris.Ris2.pf2,'c--'), hold(g_4,'off')
    xlabel(g_4,'Entalpia specifica (J/kg)');
    ylabel(g_4,'Pressione (Pa)');
    legend(g_4,'Curva limite','Curva ciclo ideale',...
        'Compressione reale','Tamb esterno','Tamb frigorifero',...
        'location','northwest')
    zoom(g_4,'on'), grid(g_4,'on')
elseif pm_g_4.Value==2 %T-s
    plot(g_4,CLim.scl,CLim.Tcl,'b'), hold(g_4,'on')
    plot(g_4,Ris.Ris2.scl,Ris.Ris2.Tcl,'r')
    plot(g_4,Ris.Ris2.s1_2,Ris.Ris2.T1_2,'r--')
    plot(g_4,Ris.Ris2.samb,Ris.Ris2.Tamb,'m--')
    plot(g_4,Ris.Ris2.sf1,Ris.Ris2.Tl1,'c--')
    plot(g_4,Ris.Ris2.sc2,Ris.Ris2.Tc2,'r')
    plot(g_4,Ris.Ris2.s3_4,Ris.Ris2.T3_4,'r--')
    plot(g_4,Ris.Ris2.sf2,Ris.Ris2.Tl2,'c--'), hold(g_4,'off')
    xlabel(g_4,'Entropia specifica (J/(kg*K))');
    ylabel(g_4,'Temperatura (K)');
    legend(g_4,'Curva limite','Curva ciclo ideale',...
        'Compressione reale','Tamb esterno','Tamb frigorifero',...
        'location','northwest')
    zoom(g_4,'on'), grid(g_4,'on')

```

```

elseif pm_g_4.Value==3 %h-s
    plot(g_4,CLim.scl,CLim.hcl,'b'), hold(g_4,'on')
    plot(g_4,Ris.Ris2.scl,Ris.Ris2.hcl,'r')
    plot(g_4,Ris.Ris2.s1_2,Ris.Ris2.h1_2,'r--')
    plot(g_4,Ris.Ris2.samb,Ris.Ris2.hamb,'m--')
    plot(g_4,Ris.Ris2.sf1,Ris.Ris2.hf1,'c--')
    plot(g_4,Ris.Ris2.sc2,Ris.Ris2.hc2,'r')
    plot(g_4,Ris.Ris2.s3_4,Ris.Ris2.h3_4,'r--')
    plot(g_4,Ris.Ris2.sf2,Ris.Ris2.hf2,'c--'), hold(g_4,'off')
    xlabel(g_4,'Entropia specifica (J/(kg*K))');
    ylabel(g_4,'Entalpia specifica (J/kg)');
    legend(g_4,'Curva limite','Curva ciclo ideale',...
           'Compressione reale','Tamb esterno','Tamb frigorifero',...
           'location','northwest')
    zoom(g_4,'on'), grid(g_4,'on')
else %p-v
    semilogx(g_4,CLim.vcl,CLim.pcl,'b'), hold(g_4,'on')
    semilogx(g_4,Ris.Ris2.vcl,Ris.Ris2.pcl,'r')
    semilogx(g_4,Ris.Ris2.v1_2,Ris.Ris2.p1_2,'r--')
    semilogx(g_4,Ris.Ris2.vamb,Ris.Ris2.pamb,'m--')
    semilogx(g_4,Ris.Ris2.vf1,Ris.Ris2.pf1,'c--')
    semilogx(g_4,Ris.Ris2.vc2,Ris.Ris2.pc2,'r')
    semilogx(g_4,Ris.Ris2.v3_4,Ris.Ris2.p3_4,'r--')
    semilogx(g_4,Ris.Ris2.vf2,Ris.Ris2.pf2,'c--'), hold(g_4,'off')
    xlabel(g_4,'Volume specifico(m^3/kg)');
    ylabel(g_4,'Pressione (Pa)');
    legend(g_4,'Curva limite','Curva ciclo ideale',...
           'Compressione reale','Tamb esterno','Tamb frigorifero')
    zoom(g_4,'on'), grid(g_4,'on')
end
end
end

```

4.2.9.2 ann_4

Viene invocata alla pressione del pulsante “pb_ann_4”; rende visibili le finestre “fig_2” e “fig_2a”, nasconde le finestre “fig_4” e “fig_4a”, riporta il menu a tendina “pm_g_4” al valore di default ed elimina gli eventuali valori inseriti nelle caselle editabili “e_p_3a”, “e_a_3a” ed “e_s_3a” di “fig_3a”.

```
function ann_4(source,eventdata)
    fig_2.Visible='on'; fig_4.Visible='off'; pm_g_4.Value=1;
    fig_2a.Visible='on'; fig_4a.Visible='off';
    e_p_3a.String=''; e_a_3a.String=''; e_s_3a.String='';
end
```

4.3 Esempio

Come parametri di input per il ciclo frigorifero semplice sono stati inseriti il fluido (R134a), le pressioni di evaporazione (0,8 bar) e di condensazione (3 bar), il rendimento del compressore (80%) ed i salti temperatura negli scambiatori (entrambi di 5°C).

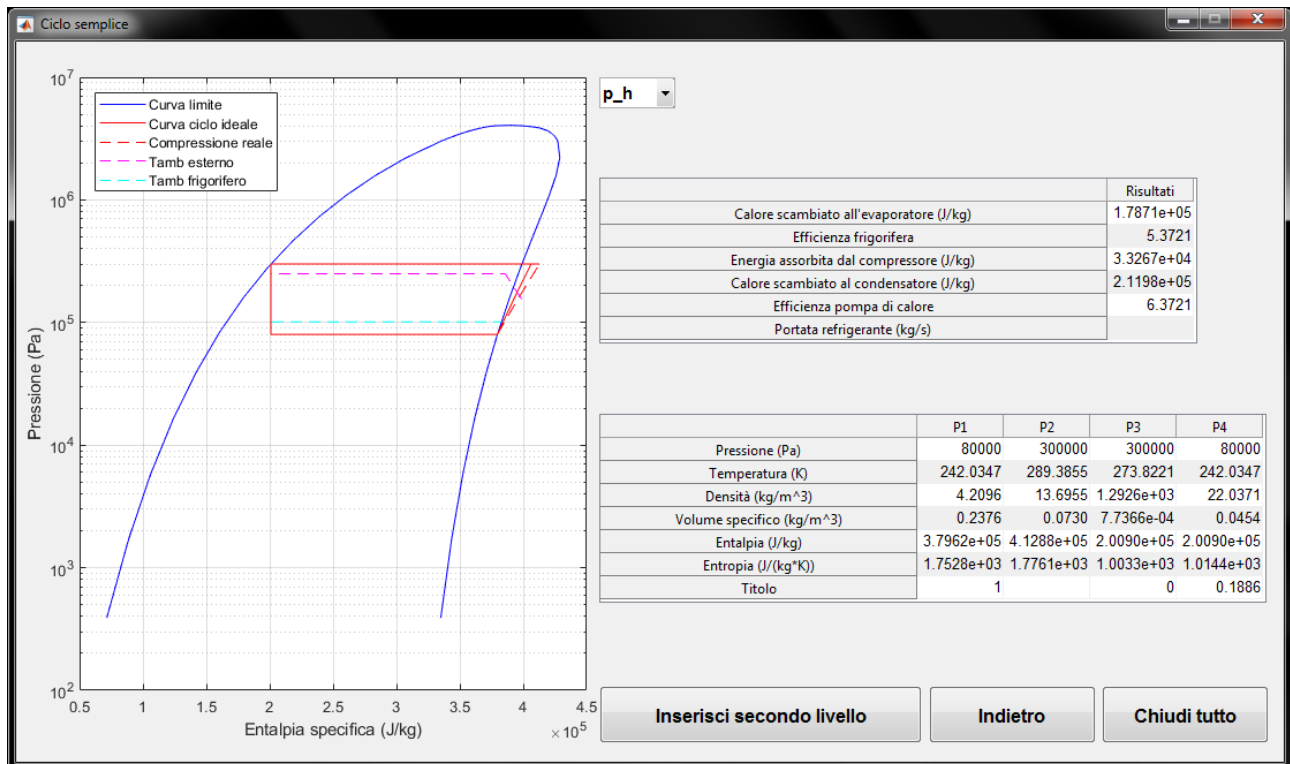
Nota: In realtà questi ultimi due parametri sono inutili ai fini dell'esempio in quanto servono per calcolare la potenza termica sottratta o ceduta dagli scambiatori stessi nel caso in cui vengano inseriti il coefficiente globale di scambio termico (α) e la superficie di scambio (S).

The screenshot shows a software interface for a simple refrigeration cycle. The window title is "Ciclo frigorifero". Inside, there is a section titled "Ciclo semplice" with the following parameters and controls:

- Fluid:** A dropdown menu set to "R134a".
- Pressione di evaporazione (Pa):** A dropdown menu with a value of "0.8e+5".
- Pressione di condensazione (Pa):** A dropdown menu with a value of "3e+5".
- Rendimento compressore:** A text input field with the value "0.8".
- ΔT ambiente frigorifero (K):** A text input field with the value "5".
- ΔT ambiente esterno (K):** A text input field with the value "5".

On the right side of the parameter area, there are three buttons: "Calcola", "Reset", and "Chiudi tutto". At the bottom of the window, there are two buttons: "Caratteristiche evaporatore" and "Caratteristiche condensatore".

Premendo il pulsante “Calcola” si ottengono i seguenti risultati:



Nota: La casella relativa al calcolo della portata di fluido frigorifero circolante è vuota in quanto precedentemente non è stata inserita la potenza termica scambiata con l'ambiente esterno o quello frigorifero.

Inoltre si vede come l'isoterma a temperatura ambiente cominci a deviare dall'orizzontale prima della curva limite superiore a causa del ridotto numero di punti utilizzati per calcolare tale curva; il problema può essere risolto aumentandone il numero (definito dalla variabile “n” nelle *Callback Functions* “curve_ciclo” e “curve_ciclo2”) ma ciò aumenta il tempo di calcolo necessario.

Inserendo la potenza termica scambiata all'evaporatore e/o quella scambiata al condensatore (nell'esempio rispettivamente 1 MW e 2 MW) è possibile calcolare la portata circolante.

Caratteristiche evaporatore

Potenza scambiata

Potenza scambiata (W)

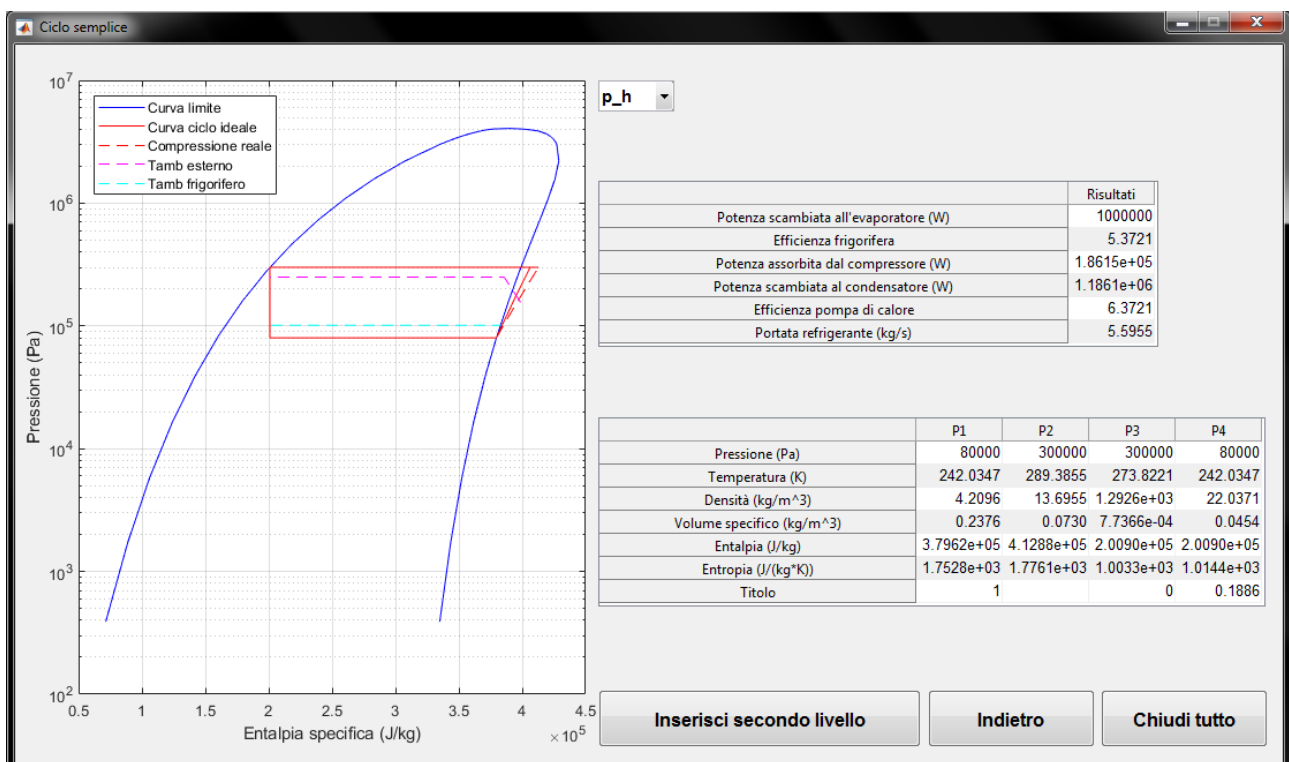
Ok Annulla

Caratteristiche condensatore

Potenza scambiata

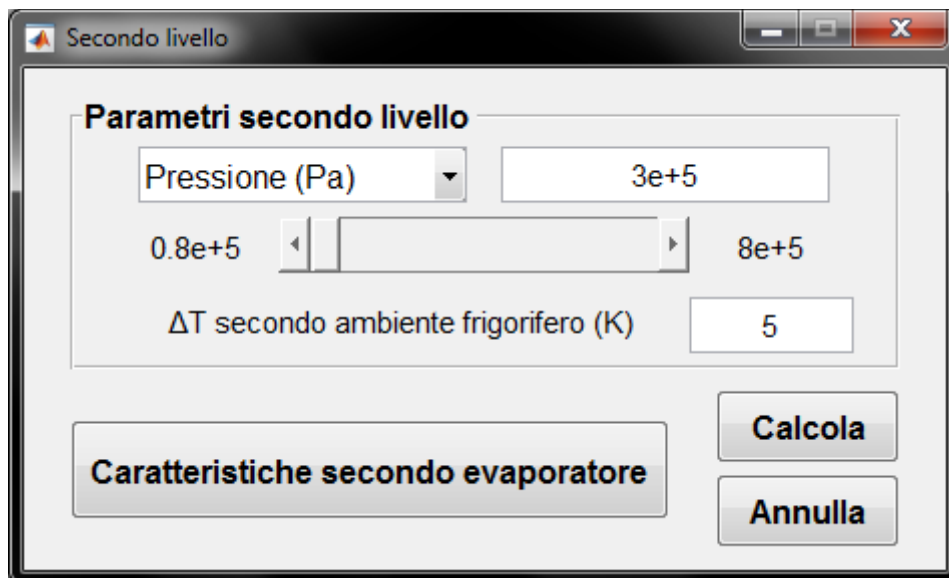
Potenza scambiata (W)

Ok Annulla



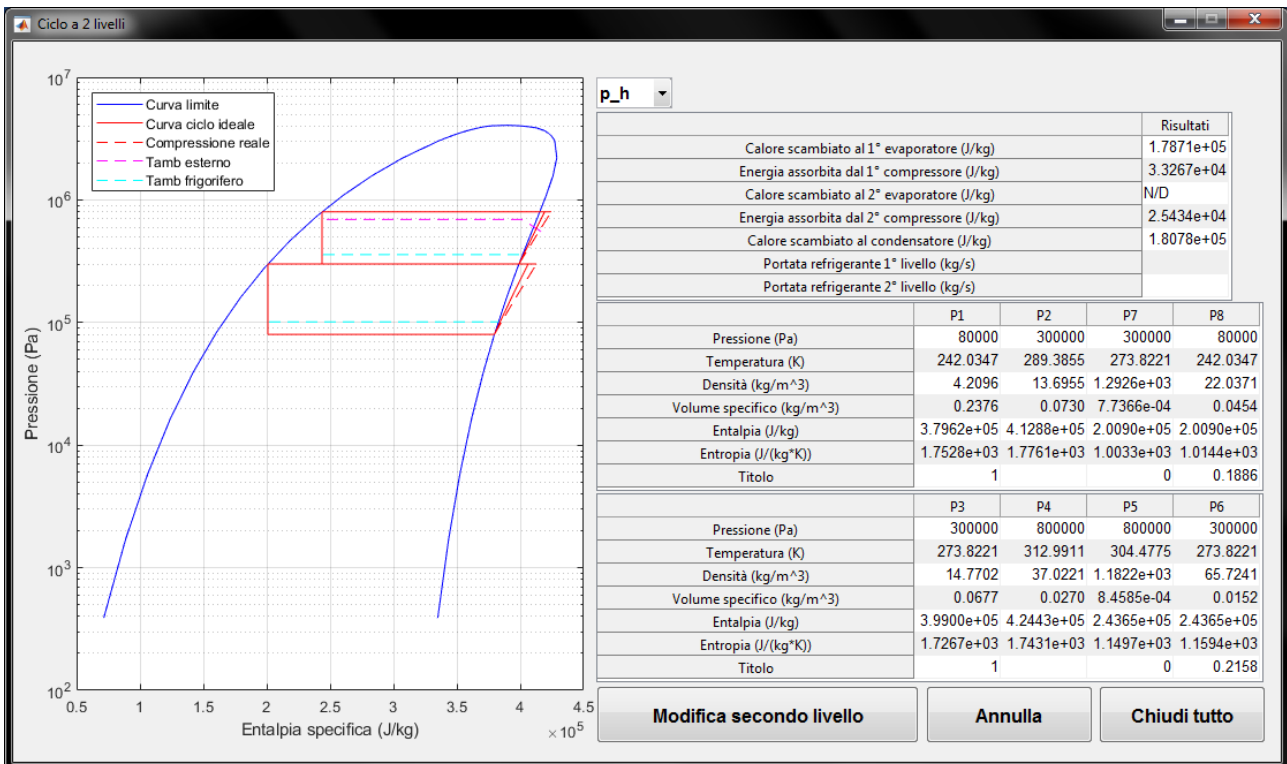
Nota: La potenza termica scambiata al condensatore inserita in input è stata ignorata.

Come parametri di input per il ciclo frigorifero a 2 livelli sono stati inseriti il fluido (R134a), le pressioni estreme di evaporazione (0,8 bar) e di condensazione (8 bar) - come se si trattasse di un ciclo semplice -, il rendimento del compressore (80%) ed i salti temperatura negli scambiatori (entrambi di 5°C). Dopo aver premuto il pulsante “Calcola” nella prima finestra è necessario premere il pulsante “Inserisci secondo livello” in quella con i risultati del ciclo semplice, inserire la pressione del secondo evaporatore (nel nostro caso 3 bar) ed eventualmente il relativo salto di temperatura con l’esterno (se si vuole calcolare la potenza termica sottratta dal secondo evaporatore attraverso il coefficiente globale di scambio termico α e la superficie di scambio S).

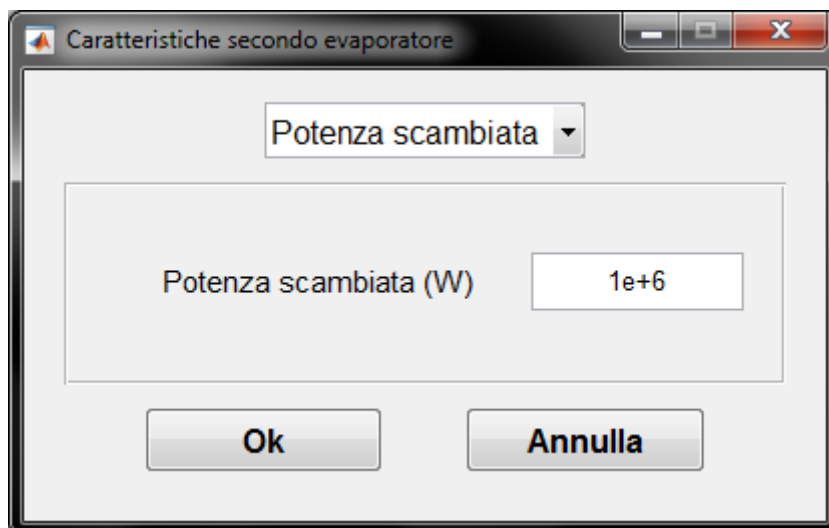


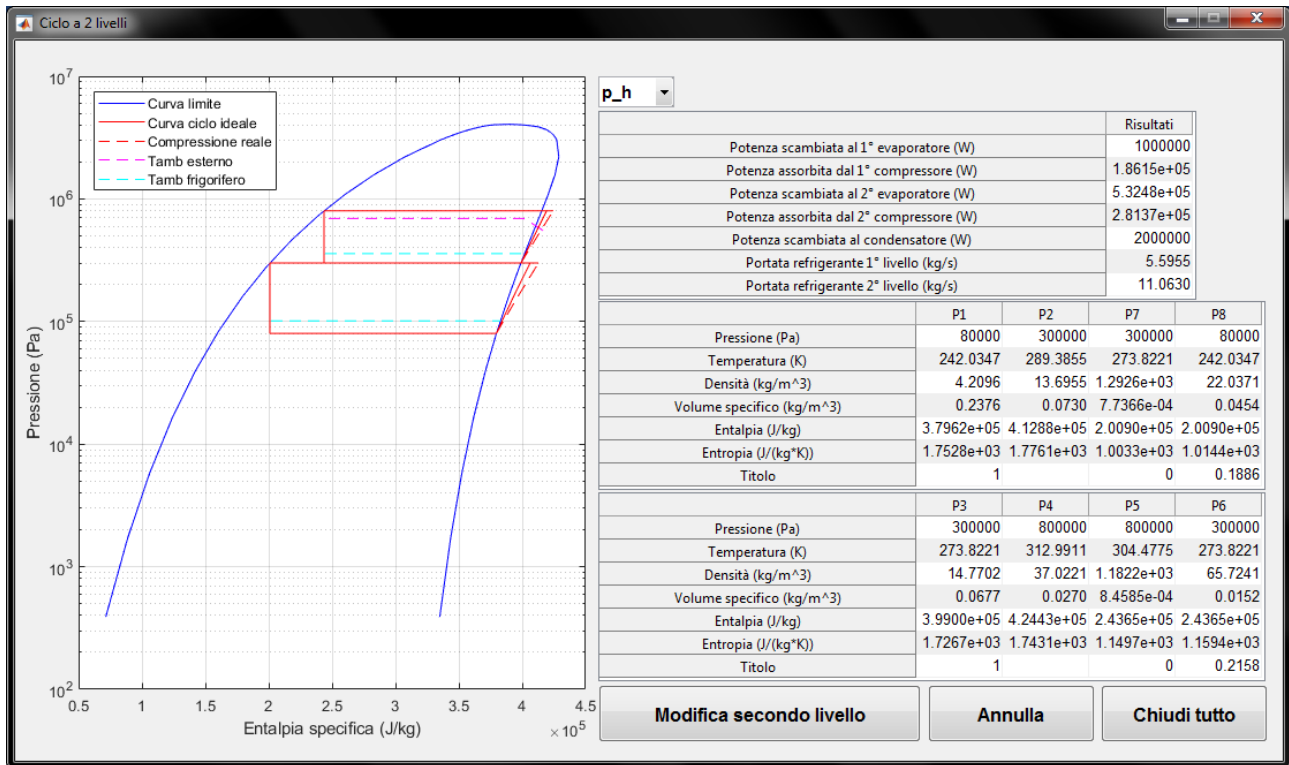
Nota: Ai lati dello slider vengono visualizzate le pressioni estreme del ciclo; se nel menu a tendina venisse selezionata l’opzione “Temperatura (K)” sarebbero invece presenti le temperature estreme del ciclo.

Premendo il pulsante “Calcola” si ottengono i seguenti risultati:



Inserendo la potenza termica assorbita o ceduta in almeno due scambiatori (nell'esempio è stato inserito 1 MW per entrambi gli evaporatori e 2 MW per il condensatore) è possibile calcolare le portate circolanti nei due livelli.





Nota: La potenza termica scambiata al secondo evaporatore inserita in input è stata ignorata.

5 Conclusioni

Questi applicativi consentono un notevole risparmio di tempo nella determinazione delle coordinate che identificano lo stato termodinamico di un fluido e dei parametri che caratterizzano un ciclo frigorifero (semplice o a doppio stadio) poiché si evita la consultazione di tabelle e diagrammi.

Nota: I valori calcolati per via numerica attraverso le funzioni “PropsSI” e “Props1SI” presentano uno scostamento trascurabile rispetto a quelli rilevati con i metodi tradizionali.

L’elevato numero di fluidi disponibili permette inoltre l’utilizzo di queste procedure per un’ampia gamma di scenari, tuttavia questi *software* sono stati concepiti per un uso prevalentemente accademico, ossia come strumenti di ausilio al docente, ad esempio nella formulazione di esempi od esercizi didattici.

Sono da segnalare alcuni errori riscontrati durante il *debug* di alcune “Funzioni speciali” dell’applicativo “coord_term”, quali:

- “DIPOLE_MOMENT” (momento di dipolo);
- “ODP” (potenziale esaurimento di ozono);
- “DELTA” (densità ridotta);
- “TAU” (temperatura ridotta reciproca);
- “ISENTROPIC_EXPANSION_COEFFICIENT” (coefficiente di espansione isoentropica).

Tuttavia, essendo CoolProp un pacchetto *open source*, è auspicabile che tali problematiche vengano risolte in future *release* delle librerie.

Possibili migliorie alle procedure potrebbero riguardare l'implementazione della possibilità di inserire come input - oltre ai fluidi standard - miscele personalizzate definite dall'utente; più specificamente per l'applicativo "ciclo_frigo" si potrebbe raffinare la modellazione del circuito includendo perdite di carico, dissipazioni di calore e simulazione di comportamenti dinamici, oltre a permettere la selezione di diverse tipologie di componenti (come scambiatori ad acqua o ad aria e compressori a capacità variabile a gradini discreti o in modo continuo) per avvicinare l'impiego di tale *software* al campo della progettazione impiantistica.

Infine per, ampliare il bacino di utenti, risulterebbe sicuramente utile la creazione di un eseguibile *stand-alone* che svincoli gli applicativi dall'ambiente MATLAB[®], ad esempio attraverso un compilatore come MATLAB Compiler[™].

6 Appendice

6.1 MATLAB®

6.1.1 *User Interface*

Per creare un'interfaccia grafica in MATLAB® è necessario innanzitutto creare una finestra, all'interno della quale è possibile inserire vari oggetti, tra cui quelli proposti nei paragrafi a seguire:

```
fig=figure;
```

Crea una figura, identificata con “fig”.

Nota: È possibile modificare le proprietà di ogni oggetto andando a variarne i valori o i parametri in questo modo:

```
fig.Proprietà=val;
```

Dove “Proprietà” va sostituito con la sigla che identifica la proprietà specifica che si vuole modificare, mentre “val” è il nuovo valore che questa deve assumere.

Nota: A seconda del tipo di proprietà “val” può essere un numero, un vettore, una stringa o un cell-array; le proprietà secondo noi fondamentali sono “Position” (vedi anche “Units”) - comune a tutti gli oggetti - “String”, “Value” e “Callback” (di cui si parlerà in seguito) - a seconda del tipo di oggetto.

Esempio:

```
fig.Position=[300 200 800 400];
```

Modifica la posizione di “fig” (i primi 2 numeri indicano rispettivamente le coordinate x e y dell'angolo sud-ovest dell'oggetto, mentre gli ultimi 2 indicano la lunghezza dei lati della figura rispettivamente lungo x e lungo y).

Nota: L'origine del sistema di riferimento è l'angolo sud-ovest dello schermo (0,0) e l'unità di misura è impostata di default in pixel (ma può essere modificata intervenendo sulla proprietà “Units”).

Per avere un elenco delle proprietà di cui dispone l'oggetto è sufficiente, dopo averlo creato, digitare il suo nome identificativo nel *Command Window* di MATLAB®.

Esempio:

```
fig
```

Mostra le proprietà di base dell'oggetto "fig".

Nota: Si consiglia di non eseguire pedissequamente le istruzioni che seguiranno poiché in tal caso gli oggetti che verranno creati andranno a sovrapporsi; gli esempi sono quindi da considerarsi separati l'uno dall'altro e il loro scopo è dare indicazioni assolutamente generiche sulla creazione di tali oggetti e sulla loro reciproca relazione *Parent-Child*.

6.1.1.1 *uipanel*

Serve a creare un pannello contenitore di oggetti, è utile per delimitare specifiche aree della figura.

```
pan=uipanel('Parent',fig);
```

Crea il pannello "pan" all'interno di "fig".

Nota: Sicuramente la sua posizione e le sue dimensioni andranno modificate (tramite "pan.Position"); di default occupa l'intero spazio disponibile nell'oggetto "Parent".

Nota: È importante definire sempre una relazione di parentela tra gli oggetti, in modo da evitare effetti indesiderati.

6.1.1.2 *uitable*

Serve a creare una tabella in cui inserire dati manualmente (ad esempio se servono come input per delle funzioni) oppure in cui mostrare dati di output.

```
tab=uitable ('Parent',pan);
```

Crea la tabella “tab” all’interno di “pan”.

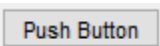
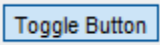
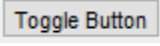
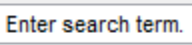
Nota: Sicuramente la sua posizione e le sue dimensioni andranno modificate (tramite “tab.Position”); si tratta infatti per ora di una tabella vuota (non contiene né righe né colonne). Andranno quindi definite anche le righe (“tab.RowName”) e le colonne (“tab.ColumnName”).


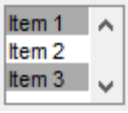
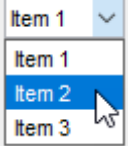
6.1.1.3 *uicontrol*

Serve a creare gli oggetti dell’interfaccia utente.

```
obj=uicontrol ('Parent',pan, 'Style',Style_value);
```

Crea l’oggetto “obj” all’interno di “pan”; il tipo di oggetto dipende da “Style_value” (vedi tabella).

Style_value	Aspetto	Descrizione
'pushbutton'		Pulsante.
'togglebutton'	 	Pulsante (visivamente come “pushbutton”) che indica il suo stato (on/off).
'checkboxbox'	<input checked="" type="checkbox"/> Check Box <input type="checkbox"/> Check Box	Casella che può essere selezionata/deselezionata.
'radiobutton'	<input checked="" type="radio"/> Radio Button <input type="radio"/> Radio Button	Come “checkboxbox”, ma - se fa parte di un “uibuttongroup” - alla spunta di un radio button corrisponde la deselegione degli altri radio button del gruppo.
'edit'		Casella editabile.
'text'	Select an item below:	Testo statico.

Style_value	Aspetto	Descrizione
'slider'		Barra che l'utente può muovere: "slider.Value" dipenderà dalla posizione della barra (0 se completamente a sinistra, 1 se completamente a destra).
'listbox'		Lista di opzioni che possono essere selezionate (più di una contemporaneamente).
'popupmenu'		Menu a tendina (solo un'opzione può essere selezionata).

6.1.1.4 axes

Serve a predisporre uno spazio in cui inserire un grafico.

```
graf=axes('Parent',fig);
```

Crea il piano cartesiano "graf" all'interno di "fig"; di default è posizionato al centro di "Parent".

Nota: Questo oggetto è fondamentalmente usato come "Parent" di comandi quali "plot", "semilogx", "semilogy", "loglog" ecc.

Esempio:

```
plot(graf,X,Y)
```

Plotta all'interno di "graf" i punti identificati dai vettori "X" e "Y".

6.1.1.5 Callback Functions

Sono funzioni che vengono associate agli oggetti creati nella *User Interface* e vengono eseguite in risposta a specifici eventi, ad esempio quando viene premuto un pulsante.

Per associare una *Callback Function* ad un oggetto:

```
obj.Callback=@nome_funz;
```

Dove “obj” è un oggetto precedentemente creato e “nome_funz” è il nome con cui viene identificata la funzione, che a sua volta deve essere implementata in questo modo:

```
function nome_funz(source,eventdata)
    pos=[0.1 0.1 0.8 0.3];
    obj.Units='normalized'; obj.Position=pos;
end
```

In questo caso viene prima creato un vettore “pos”, il quale poi viene usato per modificare la posizione di “obj” (la cui unità di misura è stata preventivamente impostata in “normalized”, ovvero in percentuale delle dimensioni del “Parent” di “obj”).

Nota: È anche possibile creare interattivamente tali interfacce tramite “App Designer” o, in alternativa, digitando il comando:

```
guide
```

6.2 COOLPROP

Nota: prima di poter invocare le funzioni della libreria è necessario il comando

```
addpath(genpath(pwd));
```

In questo modo tutte le sottocartelle della cartella di lavoro (comprese quelle con i file sorgente del pacchetto) vengono incluse nel *path* di ricerca di MATLAB®.

6.2.1 PropsSI

Questa funzione calcola le coordinate termodinamiche del fluido di interesse.

La funzione va invocata in questo modo:

```
CoolProp.PropsSI('OUT','IN1',in1,'IN2',in2,'Fluido')
```

Dove “OUT” (stringa) è il parametro che si desidera calcolare, “IN1” (stringa) è il primo parametro di input, “in1” (numero) è il valore di “IN1”, “IN2” (stringa) è il secondo parametro di input, “in2” (numero) è il valore di “IN2” e “Fluido” (stringa) è il nome del fluido di interesse (vedi sotto-paragrafo 6.2.3 Lista fluidi per i fluidi disponibili).

Questa è la lista dei parametri inseribili come input e output:

Descrizione	Parametro	Input/ Output
Densità (kg/m ³)	D	I/O
Densità molare (mol/m ³)	DMOLAR	I/O
Densità ridotta (rho/rhoc)	DELTA	I/O
Energia interna (J/kg)	UMOLAR	I/O
Energia interna molare (J/mol)	U	I/O
Entalpia molare (J/mol)	HMOLAR	I/O
Entalpia specifica (J/kg)	H	I/O
Entropia molare (J/(mol*K))	SMOLAR	I/O
Entropia specifica (J/kg*K)	S	I/O
Pressione (Pa)	P	I/O
Temperatura (K)	T	I/O
Temperatura ridotta reciproca (Tc/T)	TAU	I/O
Titolo ¹²	Q	I/O
Coefficiente di espansione isobara (1/K)	ISOBARIC_EXPANSION_COEFFICIENT	O

¹² Il valore di tale parametro deve essere compreso tra 0 e 1

Descrizione	Parametro	Input/ Output
Coefficiente di espansione isoentropica	ISENTROPIC_EXPANSION_COEFFICIENT	O
Comprimibilità isoterma (1/Pa)	ISOTHERMAL_COMPRESSIBILITY	O
Conducibilità termica (W/(m*K))	L	O
Calore specifico a p costante (J/(kg*K))	C	O
Cp ideale (J/(kg*K))	CP0MASS	O
Cp molare (J/(mol*K))	CPMOLAR	O
Cp molare ideale (J/(mol*K))	CP0MOLAR	O
Calore specifico a v costante (J/(kg*K))	O	O
Cv molare (J/(mol*K))	CVMOLAR	O
Derivata del secondo coefficiente viriale rispetto a T	DBVIRIAL_DT	O
Derivata del terzo coefficiente viriale rispetto a T	DCVIRIAL_DT	O
Derivata dell'energia ideale di Helmholtz con delta	DALPHA0_DDELTA_CONSTTAU	O
Derivata dell'energia ideale di Helmholtz con tau	DALPHA0_DTAU_CONSTDELTA	O
Derivata dell'energia residua di Helmholtz con delta	DALPHAR_DDELTA_CONSTTAU	O

Descrizione	Parametro	Input/ Output
Derivata fondamentale della dinamica del gas	FUNDAMENTAL_DERIVATIVE_OF_GAS_DYNAMICS	O
Energia di Gibbs (J/kg)	G	O
Energia di Helmholtz (J/kg)	HELMHOLTZMASS	O
Energia ideale di Helmholtz	ALPHA0	O
Energia molare di Gibbs (J/mol)	GMOLAR	O
Energia molare di Helmholtz (J/mol)	HELMHOLTZMOLAR	O
Energia residua di Helmholtz	ALPHAR	O
Entropia molare residua (J/(mol*K))	SMOLAR_RESIDUAL	O
Fase ¹³	PHASE	O
Fattore di comprimibilità	Z	O
Numero di Prandtl	PRANDTL	O
Parametro di identificazione della fase ¹⁴	PIP	O
Secondo coefficiente viriale	BVIRIAL	O
Tensione superficiale	I	O
Terzo coefficiente viriale	CVIRIAL	O
Velocità del suono (m/s)	A	O
Viscosità (Pa*s)	V	O

¹³ Output: 0=Liquido sottoraffreddato, 1=Fluido supercritico, 2=Gas supercritico, 3=Liquido supercritico, 4=Fluido al punto critico, 5=Gas surriscaldato, 6=vapore

¹⁴ Phase Identification Parameter di G. Venkatarathnam and L.R. Oellrich; usa le derivate parziali di pressione, volume e temperatura, non si riferisce alle proprietà di saturazione

Nota: Le equazioni di stato sono basate su temperatura e densità, perciò “T” e “D” sono gli input che consentono una maggiore velocità di calcolo. Quando tra gli input compare solo uno dei due il tempo di calcolo sarà un po’ maggiore (3-10 volte), e molto maggiore se nessuno dei due parametri è presente.

6.2.2 Props1SI

Questa funzione calcola le grandezze proprie del fluido in sé, e non del suo stato.

La funzione va invocata in questo modo:

```
CoolProp.Props1SI('OUT', 'Fluido')
```

Dove “OUT” (stringa) è il parametro che si desidera calcolare e “Fluido” (stringa) è il nome del fluido di interesse.

Questa è la lista dei parametri inseribili come output:

Descrizione	Parametro
Costante del gas (J/(mol*K))	GAS_CONSTANT
Concentrazione massima per soluzioni incompressibili ¹⁵	FRACTION_MAX
Concentrazione minima per soluzioni incompressibili ¹⁵	FRACTION_MIN
Densità al punto di riduzione (kg/m ³)	RHOMASS_REDUCING
Densità critica (kg/m ³)	RHOCRIT
Densità molare al punto di riduzione (mol/m ³)	RHOMOLAR_REDUCING
Densità molare critica (mol/m ³)	RHOMOLAR_CRITICAL
Fattore acentrico	ACENTRIC
Massa molare (kg/mol)	M
Momento di dipolo (C*m)	DIPOLE_MOMENT

¹⁵ Funzioni non inserite nell’applicativo perché relative a miscele di fluidi

Descrizione	Parametro
Potenziale di riscaldamento globale (20 anni)	GWP20
Potenziale di riscaldamento globale (100 anni)	GWP100
Potenziale di riscaldamento globale (500 anni)	GWP500
Potenziale esaurimento di Ozono	ODP
Pressione al punto di riduzione (Pa)	P_REDUCE
Pressione al punto triplo (Pa)	PTRIPLE
Pressione critica (Pa)	PCRIT
Pressione massima (Pa)	PMAX
Pressione minima (Pa)	PMIN
Rischio di infiammabilità	FH
Rischio fisico	PH
Rischio per la salute	HH
Temperatura al punto di riduzione (K)	T_REDUCE
Temperatura al punto triplo (K)	TTRIPLE
Temperatura critica (K)	TCRIT
Temperatura massima (K)	TMAX
Temperatura minima (K)	TMIN
Temperatura di solidificazione per soluzioni incompressibili (K) ¹⁶	T_FREEZE

Nota: È possibile evitare l'uso della funzione "PropsSI" e utilizzare solamente la funzione "PropsSI" inserendo argomenti input vuoti e valori di input nulli:

```
CoolProp.PropsSI('OUT','','0','','0','Fluido')
```

¹⁶ Funzione non inserita nell'applicativo perché relativa a miscele di fluidi

Nota: Se non si dispone della cartella con i file sorgente del pacchetto, è possibile installare CoolProp su Python (è richiesta la versione a 64-bit):

```
[v,e]=pyversion;
system([e,' -m pip install --user -U CoolProp']);
```

E successivamente richiamare la funzione “PropsSI” (che contiene in questo caso anche le funzionalità di “Props1SI”) tramite i comandi (da eseguire all’inizio di ogni funzione/script.):

```
import py.CoolProp.CoolProp.*;
import py.CoolProp.CoolProp.PropsSI.*;
```

La funzione “PropsSI” può quindi essere invocata in questo modo:

```
PropsSI('OUT','IN1',in1,'IN2',in2,'Fluido') %input non trivial
PropsSI('OUT','Fluido') %input trivial
```

6.2.3 Lista fluidi

Questa è la lista dei 122 fluidi disponibili:

- 1-Butene
- Acetone
- Air
- Ammonia
- Argon
- Benzene
- CarbonDioxide
- CarbonMonoxide
- CarbonylSulfide
- CycloHexane
- CycloPropane
- Cyclopentane
- D4
- D5
- D6
- Deuterium
- Dichloroethane
- DiethylEther
- DimethylCarbonate
- DimethylEther
- Ethane
- Ethanol
- EthylBenzene
- Ethylene
- EthyleneOxide
- Fluorine
- HFE143m
- HeavyWater

- Helium
- Hydrogen
- HydrogenChloride
- HydrogenSulfide
- IsoButane
- IsoButene
- Isohexane
- Isopentane
- Krypton
- MD2M
- MD3M
- MD4M
- MDM
- MM
- Methane
- Methanol
- MethylLinoleate
- MethylLinolenate
- MethylOleate
- MethylPalmitate
- MethylStearate
- Neon
- Neopentane
- Nitrogen
- NitrousOxide
- Novec649
- OrthoDeuterium
- OrthoHydrogen
- Oxygen
- ParaDeuterium
- ParaHydrogen
- Propylene
- Propyne
- R11
- R113
- R114
- R115
- R116
- R12
- R123
- R1233zd(E)
- R1234yf
- R1234ze(E)
- R1234ze(Z)
- R124
- R125
- R13
- R134a
- R13I1
- R14
- R141b
- R142b
- R143a
- R152A
- R161
- R21
- R218
- R22
- R227EA
- R23

- R236EA
- R236FA
- R245ca
- R245fa
- R32
- R365MFC
- R40
- R404A
- R407C
- R41
- R410A
- R507A
- RC318
- SES36
- SulfurDioxide
- SulfurHexafluoride
- Toluene
- Water
- Xenon
- cis-2-Butene
- m-Xylene
- n-Butane
- n-Decane
- n-Dodecane
- n-Heptane
- n-Hexane
- n-Nonane
- n-Octane
- n-Pentane
- n-Propane
- n-Undecane
- o-Xylene
- p-Xylene
- trans-2-Butene

Nota: Nell’inserimento del fluido come argomento delle funzioni “PropsSI” e “Props1SI” tali fluidi vanno inseriti fra apici e va rispettata la formattazione (maiuscole/minuscole).

6.3 FluidsList.mat

È il *cell-array* nel quale sono memorizzati tutti fluidi disponibili in CoolProp.

È stato creato a partire da un file Excel (“FluidsList.xls”), nel quale la prima colonna contiene il nome di ogni fluido, uno per riga.

Procedura per la creazione del file “.mat”:

```
[A,FluidsList]=xlsread('FluidsList.xls');
```

Legge il file e memorizza i valori nel nell *cell-array* “FluidsList”.


```
save('FluidsList.mat','FluidsList')
```

Salva il *cell-array* “FluidsList” con il nome di “FluidsList.mat”.

6.4 Funzioni.mat

È il *cell-array* nel quale sono memorizzate tutti gli argomenti di output implementati nelle funzioni “PropsSI” e “Props1SI”.

È stato creato a partire da un file Excel (“Funzioni.xls”), nel quale la prima colonna contiene l’argomento da inserire nelle funzioni, mentre la seconda ne contiene la descrizione; gli elementi della prima riga (1[^] e 2[^] colonna) contengono il simbolo “/” ma dovranno essere vuoti; gli elementi sono stati ordinati in modo che dalla 2[^] alla 26-esima riga siano presenti gli argomenti “*trivial*” (invocati nella funzione “Props1SI”).

Procedura per la creazione del file “.mat”:

```
[A, Funzioni]=xlsread('Funzioni.xls');
```

Legge il file e memorizza i valori nel nell *cell-array* “Funzioni” (che ha 2 colonne).

```
Funzioni(1,:)={' '};
```

Rende vuoti gli elementi della prima riga (1[^] e 2[^] colonna).

```
save('Funzioni.mat','Funzioni')
```

Salva il *cell-array* “Funzioni” con il nome di “Funzioni.mat”.

6.5 Curve limite

Questo script calcola le coordinate termodinamiche (p,T,h,s,v) dei punti sulle curve limite di ognuno dei fluidi disponibili e salva i risultati nella matrice “Fluid.mat”, dove “Fluid” è il nome del fluido utilizzato.

```

close all; clear all; clc
addpath(genpath(pwd))
load FluidsList.mat
for i=1:length(FluidsList) %scorre tutti gli elementi di FluidsList
    Fluid=cell2mat(FluidsList(i)); %imposta il fluido da utilizzare dall'elenco
    n=15; %numero di pti della curva limite inferiore/superiore

    %Variabile indipendente: Temperatura
    Tmax=CoolProp.Props1SI('T_critical',Fluid); %T punto critico
    Tmin=CoolProp.Props1SI('T_triple',Fluid); %T punto triplo
    T=linspace(Tmin,Tmax,n); %campionamento del vettore T con n elementi
spaziati equamente
    T=[T(1:n-2),linspace(T(n-1),Tmax,n/2)]; %infittire i pti da T(n-1) a T(n)
    T=[T(1:length(T)-2),linspace(T(length(T)-1),Tmax,n/2)];
    T0=T; T1=T;

    for j=1:length(T)
        p0(j)=CoolProp.PropsSI('P','T',T(j),'Q',0,Fluid); %p curva limite
inferiore
        p1(j)=CoolProp.PropsSI('P','T',T(j),'Q',1,Fluid); %p curva limite
superiore

        s0(j)=CoolProp.PropsSI('S','T',T(j),'Q',0,Fluid); %s curva limite
inferiore
        s1(j)=CoolProp.PropsSI('S','T',T(j),'Q',1,Fluid); %s curva limite
superiore

        rho0(j)=CoolProp.PropsSI('D','T',T(j),'Q',0,Fluid); %rho curva limite
inferiore
        rho1(j)=CoolProp.PropsSI('D','T',T(j),'Q',1,Fluid); %rho curva limite
superiore

        h0(j)=CoolProp.PropsSI('H','T',T(j),'Q',0,Fluid); %h curva limite
inferiore
        h1(j)=CoolProp.PropsSI('H','T',T(j),'Q',1,Fluid); %h curva limite
superiore
    end

```

```
Tcl=[T0,fliplr(T1)]; %Temperatura (K)
pcl=[p0,fliplr(p1)]; %Pressione (Pa)
scl=[s0,fliplr(s1)]; %Entropia specifica (J/(kg*K))
rhocl=[rho0,fliplr(rho1)]; %Densità (kg/(m^3))
vcl=1./rhocl; %Volume specifico (m^3/kg)
hcl=[h0,fliplr(h1)]; %Entalpia specifica (J/kg)

%Salvataggio dati
CLim=[Fluid, '.mat'];
save(CLim, 'Tcl', 'pcl', 'scl', 'vcl', 'hcl')
end
```

Il file risultante è un *cell*-array per ogni fluido contenente i vettori “Tcl”, “pcl”, “scl”, “vcl” e “hcl”, ovvero le coordinate dei punti sulla curva limite, ordinati dal punto triplo al punto critico sulla curva limite inferiore e dal punto critico al punto triplo sulla curva limite superiore.

7 Bibliografia

7.1 Sistemi Energetici

- G. Negri di Montenegro, M. Bianchi, A. Peretto. *Sistemi Energetici e loro componenti*. Pitagora.

7.2 MATLAB[®]

- S. Attaway. *MATLAB - A Practical Introduction to Programming and Problem Solving*. Elsevier.
- MathWorks. (n.d.). *MATLAB*. Retrieved from <https://it.mathworks.com/help/matlab/>
- MathWorks. (n.d.). *uipanel*. Retrieved from <https://it.mathworks.com/help/matlab/ref/uipanel.html>
- MathWorks. (n.d.). *uitable*. Retrieved from <https://it.mathworks.com/help/matlab/ref/uitable.html>
- MathWorks. (n.d.). *uicontrol*. Retrieved from <https://it.mathworks.com/help/matlab/ref/uicontrol.html>
- MathWorks. (n.d.). *axes*. Retrieved from <https://it.mathworks.com/help/matlab/ref/axes.html>
- MathWorks. (n.d.). *Create and Execute Callback Functions*. Retrieved from <https://it.mathworks.com/help/opc/ug/create-and-execute-callback-functions.html>
- MathWorks. (n.d.). *uibuttongroup*. Retrieved from <https://it.mathworks.com/help/matlab/ref/uibuttongroup.html>

7.3 CoolProp

- CoolProp Team. (n.d.). *Welcome to CoolProp*. Retrieved from <http://www.coolprop.org/>
- CoolProp Team. (n.d.). *Pure and Pseudo-Pure fluid properties*. Retrieved from http://www.coolprop.org/fluid_properties/PurePseudoPure.html

- CoolProp Team. (n.d.). *High-Level Interface*. Retrieved from www.coolprop.org/coolprop/HighLevelAPI.html
- Ian H. Bell, Jorrit Wronski, Sylvain Quoilin, Vincent Lemort. 2014. *Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp*. *Industrial & Engineering Chemistry Research*. 53 (6), 2498-2508. Retrieved from <http://pubs.acs.org/doi/abs/10.1021/ie4033999>