

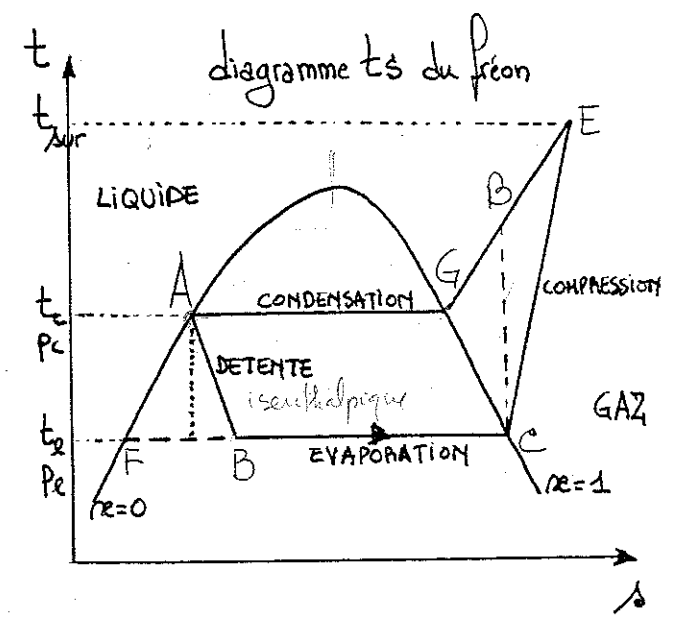
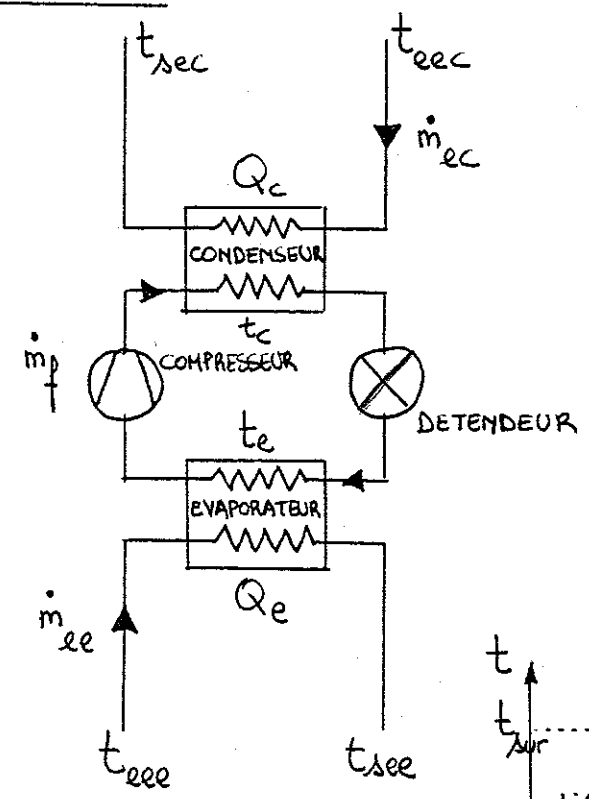
PAC

SIMULATION DE
POMPE A CHALEUR

FUL - ARLON
1989

PAC: SIMULATION DE POMPE A CHALEUR

1. Schemas



2. NOMENCLATURE

2

Symbole	Signification	Unités	Correspondance dans le Programme
t_{sec}	température de l'eau à l'entrée du condenseur	°C	TEECO
$t_{séc}$	température de l'eau à la sortie du condenseur	°C	TSECO
t_{eel}	température de l'eau à l'entrée de l'évaporateur	°C	TEEEY
t_{see}	température de l'eau à la sortie de l'évaporateur	°C	TSEY
t_e	température d'évaporation du fréon	°C	TE
t_c	température de condensation du fréon	°C	TC
t_{sur}	température de surchauffe	°C	TPE
$t_{A...G}$	température au point A...G du cycle	°C	TPA...TPG
T	température absolue	K	-
Q_c	puissance échangée au condenseur	W	QC
Q_e	puissance échangée à l'évaporateur	W	QE
\dot{m}_f	débit massique du fréon	kg/s	RHF
\dot{m}_{ec}	débit massique de l'eau au condenseur	kg/s	RMECO
\dot{m}_{ee}	débit massique de l'eau à l'évaporateur	kg/s	RMEY
P_c	pression en sortie du compresseur	Pa	PC
P_e	pression à l'entrée du compresseur	Pa	PE
K_c	coefficient d'échange du condenseur	W/K	RKCO
K_e	coefficient d'échange de l'évaporateur	W/K	RKEY
L_{vc}	chaleur sensible de condensation du fréon	J/kg	-
L_{ve}	chaleur sensible d'évaporation du fréon	J/kg	-
C_{pf}	chaleur spécifique du fréon	J/kg K	CPFREON
C_{pe}	chaleur spécifique du fluide caloporteur (eau)	J/kg K	CPEAU
X_B	titre en vapeur au point B du diagramme	-	XE

3

Symbole	Signification	Unités	Correspondance
$i(t)$	enthalpie du fréon à la température t	J/kg	-
$i'(t)$	enthalpie du fréon liquide saturé	J/kg	RILIQSAT(T)
$i''(t)$	enthalpie de la vapeur saturante de fréon	J/kg	RIVAPSAT(T)
$s(t)$	entropie du fréon à la température t	J/kg K	-
$s'(t)$	entropie du fréon liquide saturé	J/kg K	SLIQSAT(T)
$s''(t)$	entropie de la vapeur saturante de fréon	J/kg K	SVAPSAT(T)
$i_{A...G}$	enthalpie au point A...G du cycle	J/kg	RIA...RIG
$s_{A...G}$	entropie au point A...G du cycle	J/kg K	SA...SG
N	vitesses de rotation du compresseur	s ⁻¹	RNCOMP
V_c	volume massique du fréon à l'entrée du compresseur	m ³ /kg	RYPL - RVAPSAT
V_p	volume balayé par un piston du compresseur	m ³	RVP
M_{th}	rendement thermique du compresseur	-	RENDCOMP
COP_{th}	coefficient de performance théorique de la pompe à chaleur	-	{RENDEMENT COP
M	nombre de divisions dans l'intervalle de variation des paramètres	-	NINT

3. RELATIONS FONDAMENTALES

3.1. Echange fluide caloporteur ("eau") / fluide frigorigène ("fréon") au condenseur

La puissance échangée au condenseur dépend du coefficient d'échange K_c et de la différence moyenne logarithmique entre la température de condensation du fréon, t_c , et les températures d'entrée et de sortie de l'eau (échangeur à contre-courant):

$$Q_c = K_c \Delta t_{mc} \quad (1)$$

$$\Delta t_{mc} = \frac{t_{eec} - t_{sec}}{\ln\left(\frac{t_c - t_{sec}}{t_c - t_{eec}}\right)} \quad (2)$$

Cette puissance est fournie, d'une part, par la condensation du fréon à la température constante t_c et, d'autre part, par la chute de chaleur sensible entre la température de surchauffe t_{sur} et la température de condensation t_c :

$$Q_c = \dot{m}_f L_{vc} + C_{pf} \dot{m}_f (t_{sur} - t_c) \quad (3)$$

Cette même puissance est récupérée par l'eau qui s'échauffe d'une température d'entrée t_{eec} à une température de sortie t_{sec} avec un débit \dot{m}_{ec} :

$$Q_c = \dot{m}_{ec} C_{pe} (t_{sec} - t_{eec}) \quad (4)$$

3.2. Echange fluide caloporteur ("eau") / fluide frigorigène ("fréon") à l'évaporateur

De la même façon, on a, à l'évaporateur, les 4 relations suivantes:

$$Q_e = K_e \Delta t_{me} \quad (5)$$

$$\Delta t_{me} = \frac{t_{eee} - t_{see}}{\ln\left(\frac{t_e - t_{see}}{t_e - t_{eee}}\right)} \quad (6)$$

$$Q_e = \dot{m}_f (1 - x_B) L_{ve} = \dot{m}_f [i''(t_e) - i'(t_c)] \quad (7)$$

$$Q_e = \dot{m}_{ee} C_{pe} (t_{eee} - t_{see}) \quad (8)$$

Explication de la relation (7):

La chaleur Q_e fournie au fréon sert à l'évaporer: elle est donc équivalente à la chaleur latente d'évaporation du fréon, multipliée par le débit de fréon et par le titre $(1 - x_B)$ de fréon liquide au point B. Si on considère que la détente $A \rightarrow B$ est isenthalpique (on néglige le sous-refroidissement), on a aussi que l'enthalpie au point B est égale à l'enthalpie au point A pour $x=0$ (indice ') à la température t_c .

Comme par ailleurs l'enthalpie au point C est caractéristique d'une situation où le fréon est en état de vapeur saturante ($x=1$, indice'') à la température t_e , on a:

$$Q_e = \dot{m}_f (i_C - i_B) = \dot{m}_f [i''(t_e) - i'(t_c)]$$

3.3. Détente du fluide frigorigène ("fréon") liquide jusqu'à la pression d'évaporation #6

Supposons un diaphragme dans un tube : le débit volumique du fréon sera proportionnel à la racine carrée de la différence de pression ($P_c - P_e$) de part et d'autre de ce diaphragme.

+ isenthalpique

Dans notre analyse nous n'avons pas besoin de l'équation du détenteur.

3.4. Compression du fluide frigorigène ("fréon") gazeux

Pour un compresseur à piston, on peut utiliser la relation suivante [1] :

$$N = \frac{\dot{m}_f V_c}{2(1 - 0.05 \frac{P_c}{P_e}) V_p} \quad (9)$$

4. REGIMES

4.1. Régime 1 : La vitesse du compresseur, N , et le diamètre du détenteur sont commandables

Pour cette simulation, on s'impose les températures d'entrée et de sortie, ainsi que le débit de l'eau au condenseur, c'est-à-dire la puissance Q_c . Le but est de régler la vitesse N du compresseur de manière à satisfaire la puissance Q_c au condenseur. On suppose que le diamètre du diaphragme du détenteur est réglé en conséquence.

[1] Minimisation de la consommation d'énergie d'une pompe à chaleur par l'emploi d'une commande à microprocesseur - J. EHRHART
ENERGY HEATING ON HEAT PUMPS RESEARCH, DEVELOPMENT AND APPLICATIONS - BRUSSELS 27-28 Sept 78

#7
Par la relation (4), on connaît Q_c .
Puisque K_e est connu, on peut évaluer Δt_{me} par la relation (1) et enfin calculer la température de condensation t_c par (2).

Si on néglige, dans un premier temps la chaleur sensible entre la surchauffe et la condensation, la puissance Q_c est aussi le débit de fréon \dot{m}_f multiplié par la différence d'enthalpie $i''(t_c) - i'(t_c)$ entre les points G ($x=1$) et A ($x=0$).

On connaît donc le débit de fréon :

$$\dot{m}_f = \frac{Q_c}{i''(t_c) - i'(t_c)}$$

Le point A étant fixé, la détente A-B doit se réaliser de telle manière
- qu'elle soit isenthalpique
- qu'au point C on obtienne de la vapeur saturée ($x=1$)

Supposons que l'on impose la température d'entrée et le débit de l'eau à l'évaporateur. On peut connaître la puissance absorbée à l'évaporateur, Q_e , en éliminant des équations la température de sortie de l'eau, t_{see} .

En effet, par la relation (8), on a :

$$t_{see} = \frac{-Q_e}{\dot{m}_{ee} c_{pe}} + t_{see} \quad (10)$$

On injectant cette expression dans (6), on élimine t_{see} du calcul de Q_e par la relation (5).

#8

On obtient ainsi :

$$Q_e = \dot{m}_{ee} c_{pe} (t_{see} - t_e) \left[1 - \exp\left(\frac{-K_e}{\dot{m}_{ee} c_{pe}}\right) \right] \quad (11)$$

Dans cette expression, seule t_e n'est pas connue pour permettre d'évaluer Q_e .

Cependant, par la relation (7), on sait que :

$$Q_e = \dot{m}_f [i''(t_e) - i'(t_e)]$$

et t_e et donc $i'(t_e)$ sont connues.

Comme on ne sait pas extraire t_e de cette expression de façon explicite, on résout ce système de deux équations à deux inconnues (Q_e et t_e) par itérations.

On détermine donc t_e et on en déduit Q_e par (7).

La température de sortie de l'eau à l'évaporateur peut alors être déterminée, puisqu'elle est dès lors la seule inconnue de la relation (8).

Les points A, B et C du cycle sont ainsi déterminés, il reste à déterminer les caractéristiques du point de surchauffe (D ou E) et le point de départ de la condensation (G).

#9

* Considérons d'abord la compression adiabatique, isentropique (point D). C'est-à-dire que l'on néglige le travail de frottement du compresseur dont on suppose le rendement thermique unitaire.

Dans ce cas,

$$s_D = s_C = s''(t_e)$$

Pour déterminer le point G, on considère l'expression de l'entropie pour un gaz parfait :

$$ds = c_p \frac{dT}{T} - R \frac{dp}{p} \quad (12)$$

Comme les pertes de charge dans les tuyauteries du condenseur sont faibles par rapport aux différences de pression du cycle, on peut considérer le refroidissement D \rightarrow G comme isobare.

Dès lors :

$$ds = c_p \frac{dT}{T}$$

$$\text{et } s_D - s_G = c_{pf} \ln \frac{T_D}{T_G} \quad (13)$$

$$\text{ou } T_D = T_G \exp\left(\frac{s_D - s_G}{c_{pf}}\right) = T_e \exp\left[\frac{s''(t_e) - s''(t_e)}{c_{pf}}\right] \quad (14)$$

#10

* Dans la simulation, nous considérerons une transformation polytropique, plus générale, en tenant compte du rendement thermique du compresseur, que l'on imposera.

Le rendement thermique est défini comme le rapport du travail idéal (isentropique) au travail actuel, soit :

$$\eta_{th} = \frac{i_D - i_C}{i_E - i_C} \quad (15)$$

Dans cette expression, nous nous imposons η_{th} et nous connaissons $i_C = i''(t_e)$.

L'enthalpie en D peut être connue en partant du point G, puisque t_D peut être connue par la relation (14), que $t_G = t_e$, que $i_G = i''(t_e)$ et que, pour la transformation isobare $D \rightarrow G$,

$$i_D = i_G + c_{Pf}(t_D - t_G) \quad (16)$$

On peut ainsi extraire i_E de (15), les autres paramètres étant connus. Quant à la température au point E, on l'extrait d'une relation similaire à (16), soit :

$$t_E = \frac{i_E - i_G}{c_{Pf}} + t_G \quad (17)$$

Enfin, l'entropie en E est déduite d'une loi de type (13), soit :

$$s_E - s_D = c_{Pf} \ln \frac{T_E}{T_D} \quad (18)$$

#11

Dans un premier temps, nous avons négligé, dans l'évaluation de la puissance échangée au condenseur, Q_c , l'influence de la chaleur sensible entre la surchauffe et la condensation ($E \rightarrow G$). Maintenant que l'enthalpie du point E est connue, on peut corriger \dot{m}_f par $\dot{m}_f = \frac{Q_c}{i_E - i_A}$ (19).

On itère ensuite jusqu'au moment où la variation relative de ce débit soit inférieure à 0.1%.

Le COP théorique (ne tenant pas compte du rendement mécanique du compresseur) peut enfin être déterminé comme le rapport de cette puissance Q_c fournie au condenseur à l'énergie dépensée pour la compression, soit :

$$COP_{th} = \frac{Q_c}{\dot{m}_f (i_E - i_C)} \quad (20)$$

4.2. Régime 2 : la vitesse du compresseur, N, et le diamètre du détendeur sont imposés

Au condenseur, comme à l'évaporateur, on ne s'impose que le débit et la température d'entrée de l'eau. On ne connaît donc, ni Q_c , ni Q_e .

On élimine l'inconnue t_{see} des équations (5), (6) et (8).

#12

On a:

$$Q_e = \dot{m}_{ee} c_{pe} (t_e - t_{eee}) \left[-1 + \exp\left(\frac{-K_e}{\dot{m}_{ee} c_{pe}}\right) \right] \quad (21)$$

On a par la relation (7):

$$Q_e = \dot{m}_f [i'(t_e) - i'(t_c)] \quad (22)$$

De la même façon, au condenseur, on a:

$$Q_c = \dot{m}_{ec} c_{pe} (t_c - t_{eec}) \left[1 - \exp\left(\frac{-K_c}{\dot{m}_{ec} c_{pe}}\right) \right] \quad (23)$$

et:

$$Q_c = \dot{m}_f [i_E - i_A] \quad (24)$$

Enfin, le débit de fréon est connu par l'équation du compresseur:

$$\dot{m}_f = 2N \left[1 - 0.05 \frac{P_c}{P_e} \right] \cdot \frac{V_p}{V_c} \quad (25)$$

Dans la relation (24), les enthalpies aux points E et A sont évaluées par des équations du type (14), (15) et (16).

On obtient donc 5 équations (21 à 25) à 5 inconnues (Q_e, Q_c, t_e, t_c et \dot{m}_f).

#13

On résout le système par itérations:

Au départ, on s'impose t_e et t_c , ainsi que \dot{m}_f , et on calcule la différence $\varepsilon = (21) - (22)$ qui doit tendre vers zéro.

Ensuite, on boucle:

- calcul de \dot{m}_f par (25),
- calcul d'une nouvelle température t_e par (23) et (24),
- calcul de la nouvelle différence ε ,
- correction de t_e en fonction de l'écart avec l'ancienne différence ε calculée,
- correction de \dot{m}_f , de t_c et de la différence ε jusqu'au moment où ε est inférieur à 10^{-4} .

Les 5 inconnues sont dès lors estimées.

Pour connaître les températures, entropies et enthalpies aux 6 points du cycle (A, B, C, D, E et G), on utilise les mêmes relations que pour le premier régime:

- transformations isobares $D \rightarrow G$ et $E \rightarrow D$.
- règle de proportionnalité pour le point B:

$$x_B = \frac{s_B - s_F}{s_C - s_F} = \frac{i_B - i_F}{i_C - i_F}$$

- relation (15) qui tient compte du rendement thermique du compresseur.

4.3. Régime 3 : Variation d'un paramètre

Ce régime n'est en fait qu'une variante du régime 2 où l'on autorise l'évolution d'une des variables t_{ee} , t_{ee} , m_{ee} ou m_{ee} entre deux bornes.

On divise cet intervalle en m et on effectue m simulations de type 2, en calculant chaque fois l'état d'un des indices suivants : COP, Q_c ou Q_e .

Ce régime permet de visualiser l'évolution de ces indices en fonction de la variable qui évolue.

5. INFORMATIQUE

Le programme est disponible sur un PC de type XT, AT ou 386 sous MS-DOS.

La carte graphique couleur EGA est indispensable pour l'interface utilisateur et le co-processeur mathématique est fort utile pour assurer une bonne vitesse d'exécution.

La source du logiciel est écrite en FORTRAN 77, la version actuelle (version 1.0) comprend environ 1300 lignes de code, sans les commentaires, ni les bibliothèques.

La version exécutable occupe environ 80 Koctets. Elle a été obtenue par la compilation du code avec le "MICROSOFT FORTRAN OPTIMIZING COMPILER", version 4.01, avec l'option /FP:87 qui génère des instructions tenant compte du co-processeur mathématique 8087/80287/80387.

L'édition de liens a été réalisée avec le "MICROSOFT OVERLAY LINKER" en ajoutant aux modules "objet" la bibliothèque FORTRAN et une bibliothèque graphique.

Cette bibliothèque graphique comprend les primitives du "GRAPHICS DEVELOPMENT TOOLKIT" d'IBM et un ensemble de fonctions évoluées écrites par nous, en FORTRAN, dans GLIB.

Les deux parties (FORVDI et GLIB) ont été réunies dans une bibliothèque regroupable : GRAPH.LIB.

La partie graphique du programme exige l'installation du driver VDIDY010.SYS (IBM Enhanced Graphics Adapter - Enhanced Color Display) et de VDI.SYS qui contient la partie résidente du "Toolkit". Enfin, il est utile d'ajouter également au fichier de boot, COMMAND.SYS, l'installation du driver DMDRVR.BIN qui permet notamment d'accéder à d'autres partitions (D: par exemple).

Les sources du programme sont réparties en 7 blocs, qui constituent 7 fichiers .FOR.

* PAC.FOR constitue la source du programme principal qui gère l'ensemble des simulations et des entrées et sorties.

#16

- * SIMUL.FOR contient les trois routines de simulation pour les 3 régimes envisagés,
- * MENUS.FOR contient tous les menus de choix d'options qui seront affichés, en mode graphique, sur l'écran,
- * FICHIERS.FOR contient les routines d'ouverture, d'écriture, de lecture et de fermeture des fichiers.
- * DESSIN.FOR contient tous les dessins qui apparaîtront sur l'écran (PAC, diagramme TS, ...)
- * FREON12.FOR contient les 6 fonctions ajustées définissant les caractéristiques du fréon 12.
- * GLIB.FOR contient la source des fonctions graphiques évoluées faisant appel aux primitives de GDT. Ce fichier n'a normalement rien à voir avec la simulation proprement dite.

Ces sources sont commentées et ne nécessitent pas d'informations plus détaillées.

Un certain nombre de fichiers sont utilisés par le programme :

- * PACDONSI.DAT est un fichier d'entrée il contient
 - le coefficient d'échange thermique au condenseur, $RKCO$, en hJ/Ks (hW/K)
 - le coefficient d'échange thermique à l'évaporateur, $RKEV$, en hJ/Ks (hW/K)
 - la chaleur spécifique du fluide caloporteur, $CPEAU$, en hJ/kgK
 - la chaleur spécifique du fréon, $CPFREON$, en hJ/kgK
 - le rendement thermique du compresseur, $RENDCHP$, entre 0 et 1.

- * PACSORSI.DAT est un fichier de sortie, reprenant
 - les températures et débits des fluides aux échangeurs
 - les caractéristiques des points du cycle
 - les puissances et rendements

#17

- * CYCLE.DAT est un fichier de travail, qui peut être utile en sortie pour un traitement ultérieur, et qui reprend les entropies aux 6 points du cycle pour la simulation envisagée.
- * TABLE.DAT est un fichier produit par la simulation et qui fournit les caractéristiques du fréon entre $-100^{\circ}C$ et $110^{\circ}C$ de 5 en 5 degrés. IP peut être utilisé à des fins de vérification.

Le fonctionnement du programme est très simple :

- * après l'affichage et le dessin d'un logo de départ, l'utilisateur est invité à frapper la touche "ENTER" pour démarrer le programme.
- * le menu principal de choix de type de simulation apparaît alors :
 1. simulation
 2. évolution des paramètres
 3. quitter
- * si la "simulation" est choisie, l'utilisateur sera confronté au choix suivant :
 1. vitesse du compresseur libre
 2. vitesse du compresseur imposée
- * dans les 2 cas il entrera les paramètres notés sur le schéma de la pompe à chaleur : températures d'entrée et de sortie de l'eau au condenseur, température d'entrée de l'eau à l'évaporateur, débits d'eau aux deux échangeurs, ainsi que le volume balayé par le piston pour le compresseur dans le cas où la vitesse est libre.
- Dans le cas où la vitesse est imposée, on imposera pas la température de l'eau en sortie du condenseur, mais on imposera la vitesse de rotation du compresseur.

#18

- * dans les deux cas, le programme sauvera les données sur fichiers et dessinera sur le même écran la pompe à chaleur avec la valeur des différentes variables et le diagramme t_s du fluon avec le schéma du cycle.
- * l'utilisateur pourra éventuellement visualiser le diagramme t_s sur toute la surface de l'écran
- * il pourra ensuite revenir au menu principal ou recommencer un nouveau cycle dont le parcours sur le diagramme t_s se superposera au premier, ce qui permet les comparaisons.
- * si l'utilisateur choisit l'option "évolution des paramètres", il pourra choisir en début de simulation le paramètre qu'il désire voir évoluer et en fin de simulation la variable à visualiser en fonction de ce paramètre.

6. EXEMPLE

Afin de vérifier si le programme fonctionne correctement, voici à titre d'exemple les données d'entrée et les résultats pour une simulation simple, avec vitesse du compresseur imposée.

En entrée, on impose:

$$K_e (RKCO) = 4 \text{ kJ/Ks} = 4 \text{ kW/K}$$

$$K_e (RKEY) = 4 \text{ kW/K}$$

$$C_{pe} (CPEAU) = 4.18 \text{ J/kg K}$$

$$C_{pf} (CPFREON) = 0.84 \text{ J/kg K}$$

#19

$$\eta_{th} (RENDCOMP) = 0.8$$

$$t_{eec} = 10^\circ\text{C} \quad \dot{m}_{ec} = 1 \text{ kg/s} \quad t_{eee} = 0^\circ\text{C} \quad \dot{m}_{ee} = 1 \text{ kg/s}$$

$$V_b = 200 \text{ cm}^3 \quad N = 100 \text{ tours/s}$$

En sortie, on obtient:

$$t_{sec} = 22^\circ\text{C} \quad t_{see} = -9.5^\circ\text{C}$$

$$t_c = 29.4^\circ\text{C} \quad t_e = -15.4^\circ\text{C}$$

$$P_c = 0.734 \text{ MPa} \quad P_e = 0.180 \text{ MPa}$$

$$Q_c = 50.06 \text{ kW} \quad Q_e = 39.62 \text{ kW}$$

$$\text{COP}_{th} = 4.713$$

PAC.FOR

```

implicit integer*2 (v)
COMMON/INOUT/QC, TC, TSEEV, TEEV, RMEEV, TSECO, TEECO, RMECO
COMMON/INOUT2/TEEV2, RMEEV2, TEECO2, RMECO2
COMMON/SIMUL/COP (100), QCOND (100), QEVAP (100)
COMMON/CONSTHER/RKCO, RKEV, CPEAU, CPFREON
COMMON/COMPRESSEUR/RVP, RENDCOMP
COMMON/REGIME/IQCOND, IREGIME
COMMON/CYCLE/TPA, TPB, TPC, TPD, TPE, TPG,
*          RIA, RIB, RIC, RID, RIE, RIG,
*          SA, SB, SC, SD, SE, SG,
*          PC, PE
COMMON/PUISS/QEV, QCO, TCOMP, RENDEMENT, RMF, RNCOMP
common /graph/ vxmax, vymax,
*          devhandle
common/absrel/x1a, x2a, y1a, y2a, x1r, x2r, y1r, y2r
dimension xy (100), vxy (100), vaxe (4)
character*80 rep, choix

C
C INITIALISATION
C
      call init_graph
      call logo
      call openfic
      call lecpadon
      call cretable

C
C CHOIX : 1.SIMULATION 2.EVOLUTION DES PARAMETRES 3.QUITTER
C
5      call menu3 (rep)
      if (rep(1:1).eq.'1') goto 2
      if (rep(1:1).eq.'2') goto 3000
      IF (REP(1:1).EQ.'3') GOTO 2020
      goto 5

C
C SIMULATION SIMPLE
C
2      nsimul=0
1      nsimul=nsimul+1
      call dessin

C
C CHOIX : VITESSE DU COMPRESSEUR : 1.LIBRE 2.IMPOSEE
C
10     call qregime (REP)
C
C VITESSE LIBRE
      IF (REP(1:1).EQ.'1') THEN
      call dessin
      call entrel
      call simull
      GOTO 2000

C
C VITESSE IMPOSEE

```

```

ELSEIF (REP(1:1).EQ.'2') THEN
  call dessin
  call entre2
  call simul2
  GOTO 2000

ELSE
  GOTO 10
ENDIF

C
C SORTIE, AFFICHAGE, DESSIN, SAUVETAGE DES DONNEES
C
2000 call sortie
    call savcycl

2010 call dessin
    call sortgr
    call diagts (0,0,NSIMUL,70.,100.,25.,75.)

C
C CHOIX : 1.NOUVEAU CYCLE 2.DESSIN DU DIAGRAMME TS 3.RETOUR AU MENU
C
2017 call menud(REP)
    IF (REP(1:1).EQ.'1') GOTO 1
    IF (REP(1:1).EQ.'3') GOTO 5
    if (rep(1:1).eq.'2') goto 2018
    goto 2017

C
C DIAGRAMME TS
C
2018 call diagts(7,5,nsimul,10.,90.,10.,86.)

C
C CHOIX : 1.NOUVEAU CYCLE 2.DESSIN DE LA PAC 3.QUITTER
C
2019 call menuc(REP)
    IF (REP(1:1).EQ.'1') GOTO 1
    IF (REP(1:1).EQ.'3') GOTO 2020
    IF (REP(1:1).EQ.'2') GOTO 2010
    goto 2019

C
C COURBES COP OU PUISSANCES LORSQU'UN PARAMETRE VARIE
C
3000 call dessin

C
C CHOIX DU PARAMETRE VARIABLE
C
    call menu4(rep)
    choix=rep

C
C SIMULATION DANS LE CAS D'UN PARAMETRE VARIABLE
C
    call entre3(rep)
    call simul3(rep)

C
C CHOIX DE LA VARIABLE A VISUALISER : 1.COP 2.W COND. 3.W EVAP
4.RETOUR
C
3010 call menu5(rep)
    call varicurv(choix,rep)
    if (rep(1:1).eq.'4') go to 5
    go to 3010

```

```

C
C ARRET DU PROGRAMME
C
2020 call closfic
    call close_graph
    stop
end

```

SIMUL.FOR

```

-----
subroutine simull
-----
c
c Routine de simulation dans le cas du regime :
c "Vitesse du compresseur libre"
c
COMMON/INOUT/QC, TC, TSEEV, TEEV, RMEEV, TSECO, TEECO, RMECO
COMMON/CONSTHER/RKCO, RKEV, CPEAU, CPFREON
COMMON/COMPRESSEUR/RVP, RENDCOMP
COMMON/REGIME/IQCOND, IREGIME
COMMON/CYCLE/TPA, TPB, TPC, TPD, TPE, TPG,
* RIA, RIB, RIC, RID, RIE, RIG,
* SA, SB, SC, SD, SE, SG,
* PC, PE
COMMON/PUISS/QEV, QCO, TCOMPTH, RENDEMENT, RMF, RNCOMP
c
c#####
c
c principe de resolution :
c Au condenseur, on connait le debit (RMECO) et les temperatures
c entree (TSECO) et sortie (TEECO) de l'eau
c => Puissance QC connue => le rapport de QC au coefficient
c de transfert du condenseur fournit le delta Tm et donc la
c temperature de condensation
c Connaissant cette temperature, on evalue les enthalpies
c du freon dans les etats liquide et gaz et on en deduit le
c debit du freon (RMF)
c
500 QC=RMECO*CPEAU*(TSECO-TEECO)

DTMEEV=QC/RKCO
TERCAL=EXP((TEECO-TSECO)/DTMEEV)
TC=(TSECO-TEECO*TERCAL)/(1-TERCAL)

RLVC=RIVAPSAT(TC)-RILIQSAT(TC)
RMF=QC/RLVC

```

```

c
c -----
c A l'evaporateur, on connait le debit (RMEEV) et la temperature
c d'entree (TEEV) de l'eau, en eliminant dans les equations la
c temperature de sortie de l'eau, on peut calculer le rapport RH1
c entre la puissance echangee a l'evaporateur (QE) et la difference
c de temperature (TEEV-TE) avec TE=temperature d'evaporation du freon.
c La puissance QE=RH1*(TEEV-TE) doit aussi egaler la difference
c d'enthalpie du freon (RIVAPSAT(TE)-RILIQSAT(TC)) multipliee par le
c debit RMF du freon
c On doit donc annuler la difference :
c FTE=RH1*(TEEV-TE)-(RIVAPSAT(TE)-RILIQSAT(TC))*RMF
c Ainsi, par iterations, on parvient a determiner TE
c
1 RH1=RMEEV*CPEAU*(1-EXP(-RKEV/RMEEV/CPEAU))
TE=2.
FTE=RH1*(TEEV-TE)-(RIVAPSAT(TE)-RILIQSAT(TC))*RMF

c
c debut des iterations
105 IF (ABS(FTE).LE.0.001) GOTO 110
TE1=TE+1.
FTE1=RH1*(TEEV-TE1)-(RIVAPSAT(TE1)-RILIQSAT(TC))*RMF
TE=-FTE/(FTE1-FTE)+TE
FTE=RH1*(TEEV-TE)-(RIVAPSAT(TE)-RILIQSAT(TC))*RMF
GOTO 105

c
c fin des iterations
110 CONTINUE
XE=(RILIQSAT(TC)-RILIQSAT(TE))/(RIVAPSAT(TE)-RILIQSAT(TE))
RLVE=RIVAPSAT(TE)-RILIQSAT(TE)
QE=RMF*(1-XE)*RLVE

c
c Calcul de la temperature de sortie de l'eau a l'evaporateur
c et de la vitesse du compresseur
c
TSEEV=TEEV-QE/(RMEEV*CPEAU)
PE=PSAT(TE)
RVPC=RVAPSAT(TE)
PC=PSAT(TC)
RNCOMP=RMF*RVPC/2/(1-0.05*PC/PE)/RVP

c
c
c Determination des points inconnus dans le cycle du freon
c
SA=SLIQSAT(TC)
SC=SVAPSAT(TE)
SF=SLIQSAT(TE)
SB=SF+XE*(SC-SF)
SD=SC
SG=SVAPSAT(TC)

TPD=(TC+273)*EXP((SD-SG)/CPFREON)-273
TPA=TC
TPB=TE
TPC=TE
TPG=TC

RIA=RILIQSAT(TPA)
RIF=RILIQSAT(TE)
RIC=RIVAPSAT(TPC)
RIB=RIF+XE*(RIC-RIF)
RIG=RIVAPSAT(TPG)
RID=RIG+CPFREON*(TPD-TPG)

```

```

RIE=RIC+(RID-RIC)/RENDCOMP
TPE=(RIE-RIG)/CPFREON+TPG
SE=SD+CPFREON*ALOG((TPE+273)/(TPD+273))
QEV=RMF*(RIC-RIB)

```

```

c
c En fait QC=RMF*(RIE-RIA) et non QC=RMF*(RIG-RIA)
c Afin de tenir compte dans QC de la chaleur sensible entre E et G
c on corrige le debit de freon calcule et on itere jusqu'au
c moment ou l'evolution relative du debit est inferieure au pourmille
c

```

```

TEMPO=QC/(RIE-RIA)
IF (ABS((TEMPO-RMF)/RMF).GT.0.001) THEN
  RMF=TEMPO
  GO TO 1
ENDIF
QCO=QC

```

```

TCOMPTH=RMF*(RIE-RIC)
RENDEMENT=QC/TCOMPTH

```

```

return
end

```

```

-----
c      subroutine simul2
-----

```

```

c
c Routine de simulation dans le cas du regime :
c "vitesse du compresseur imposee"
c
COMMON/INOUT/QC, TC, TSEEV, TEEEV, RMEEV, TSECO, TEECO, RMECO
COMMON/CONSTHER/RKCO, RKEV, CPEAU, CPFREON
COMMON/COMPRESSEUR/RVP, RENDCOMP
COMMON/REGIME/IQCOND, IREGIME
COMMON/CYCLE/TPA, TPB, TPC, TPD, TPE, TPG,
*      RIA, RIB, RIC, RID, RIE, RIG,
*      SA, SB, SC, SD, SE, SG,
*      PC, PE
COMMON/PUISS/QEV, QCO, TCOMPTH, RENDEMENT, RMF, RNCOMP

```

```

c
c Principe de resolution :
c Sont connus les debits et les temperatures d'entree
c de l'eau dans les deux echangeurs et la vitesse du compresseur.
c La temperature de sortie de l'eau au condenseur est inconnue et
c donc egalement la puissance echangee au condenseur
c En eliminant la temperature de sortie de l'eau a l'evaporateur
c dans les equations, on obtient un systeme de 5 equations a 5 inconnues
c (les temperatures d'evaporation et de condensation, le debit de freon
c et les puissances echangees a l'evaporateur et au condenseur).
c Ce systeme d'equations est resolu par iterations.
c

```

```

c
1000 RHEV=RMEEV*CPEAU*(-1+EXP(-RKEV/RMEEV/CPEAU))
RHCO=RMECO*CPEAU*(+1-EXP(-RKCO/RMECO/CPEAU))
TE=-10.
TC=+40.
RMF=1.
FTE=-TE+RMF/RHEV*(RIVAPSAT(TE)-RILIQSAT(TC))+TEEEV

```

```

c      debut des iterations
115 IF (ABS(FTE).LE.0.0001) GOTO 120

```

```

TE1=TE+1.
RMF=2*RNCOMP*(1-.05*PSAT(TC)/PSAT(TE1))*RVP/RVAPSAT(TE1)
SD=SVAPSAT(TE)
SG=SVAPSAT(TC)
TPD=(TC+273.)*EXP((SD-SG)/CPFREON)-273.
RIG=RIVAPSAT(TC)
RID=RIG+CPFREON*(TPD-TC)
RIC=RIVAPSAT(TE)
RIE=RIC+(RID-RIC)/RENDCOMP
ENTE=RIE-RILIQSAT(TC)
TC=RMF/RHCO*ENTE+TECO
FTE1=-TE1+RMF/RHEV*(RIVAPSAT(TE1)-RILIQSAT(TC))+TEEEV

TE=-FTE/(FTE1-FTE)+TE

```

```

RMF=2*RNCOMP*(1-.05*PSAT(TC)/PSAT(TE))*RVP/RVAPSAT(TE)
SD=SVAPSAT(TE)
SG=SVAPSAT(TC)
TPD=(TC+273.)*EXP((SD-SG)/CPFREON)-273.
RIG=RIVAPSAT(TC)
RID=RIG+CPFREON*(TPD-TC)
RIC=RIVAPSAT(TE)
RIE=RIC+(RID-RIC)/RENDCOMP

```

```

ENTEA=RIE-RILIQSAT(TC)
TC=RMF/RHCO*ENTEA+TEECO
FTE=-TE+RMF/RHEV*(RIVAPSAT(TE)-RILIQSAT(TC))+TEEEV

```

```

GOTO 115
      fin des iterations

```

```

c
120 continue
QC=RMF*ENTEA
1 XE=(RILIQSAT(TC)-RILIQSAT(TE))/(RIVAPSAT(TE)-RILIQSAT(TE))
RLVE=RIVAPSAT(TE)-RILIQSAT(TE)
QE=RMF*(1-XE)*RLVE
TSEEV=TEEEV-QE/(RMEEV*CPEAU)

```

```

c
c Les temperatures de condensation et d'evaporation, ainsi que
c le debit du freon etant connus, on peut rechercher les differents
c points du cycle dans le diagramme du freon.

```

```

PE=PSAT(TE)
RVPC=RVAPSAT(TE)
PC=PSAT(TC)

```

```

SA=SLIQSAT(TC)
SC=SVAPSAT(TE)
SF=SLIQSAT(TE)
SB=SF+XE*(SC-SF)
SD=SC
SG=SVAPSAT(TC)

```

```

TPD=(TC+273)*EXP((SD-SG)/CPFREON)-273
TPA=TC
TPB=TE
TPC=TE
TPG=TC

```

```

RIA=RILIQSAT(TPA)
RIF=RILIQSAT(TE)
RIC=RIVAPSAT(TPC)
RIB=RIF+XE*(RIC-RIF)
RIG=RIVAPSAT(TPG)
RID=RIG+CPFREON*(TPD-TPG)

```

```

RIE=RIC+(RID-RIC)/RENDCOMP
TPE=(RIE-RIG)/CPFREON+TPG
TS=TPE
SE=SD+CPFREON*ALOG((TPE+273)/(TPD+273))
QEV=RMF*(RIC-RIB)
QCO=QC

```

```

TCOMPTH=RMF*(RIE-RIC)
RENDEMENT=QC/TCOMPTH

```

```

TSECO=QC/RMECO/CPEAU+TEECO
TSEEV=-QE/RMEEV/CPEAU+TEESE

```

```

return
end

```

```

-----
SUBROUTINE SIMUL3(REP)
-----

```

```

c
c Routine de simulation dans le cas d'une variation de parametres

```

```

c
c implicit integer*2 (v)
COMMON/INOUT/QC,TC,TSEEV,TEEEV,RMEEV,TSECO,TEECO,RMECO
COMMON/INOUT2/TEEEV2,RMEEV2,TEECO2,RMECO2
COMMON/SIMUL/COP(100),QCOND(100),QEVAP(100)
COMMON/CONSTHER/RKCO,RKEV,CPEAU,CPFREON
COMMON/COMPRESSEUR/RVP,RENDCOMP
COMMON/REGIME/IQCOND,IREGIME
COMMON/CYCLE/TPA,TPB,TPC,TPD,TPE,TPG,
* RIA,RIB,RIC,RID,RIE,RIG,
* SA,SB,SC,SD,SE,SG,
* PC,PE
COMMON/PUISS/QEV,QCO,TCOMPTH,RENDEMENT,RMF,RNCOMP
common /graph/ vxmax,vymax,
* devhandle

```

```

character*80 texte,rep,rep1

```

```

TEEEV1=TEEEV
TEECO1=TEECO
RMEEV1=RMEEV
RMECO1=RMECO

```

```

c
c On realise NINT simulations simples (SIMUL2) et ont note chaque fois
c les valeurs desirees

```

```

NINT=19

```

```

DO 10 I=0,NINT

```

```

texte=' '
ncount=8
VCOUL=1
CALL PRNT(VCOUL,NCOUNT,12.,65.,TEXTE)
texte='PATIENCE'
CALL PRNT(VCOUL,NCOUNT,12.,65.,TEXTE)

```

```

TEECO=TEECO1+(TEECO2-TEECO1)/NINT*I
TEEEV=TEEEV1+(TEEEV2-TEEEV1)/NINT*I
RMECO=RMECO1+(RMECO2-RMECO1)/NINT*I
RMEEV=RMEEV1+(RMEEV2-RMEEV1)/NINT*I

```

```

CALL SIMUL2

```

```

c
c Sauvetage des abscisses

```

```

c
IF (REP(1:1).EQ.'1') THEN
QCOND(2*I+1)=TEECO
COP(2*I+1)=TEECO
QEVAP(2*I+1)=TEECO
ENDIF

```

```

IF (REP(1:1).EQ.'2') THEN
QCOND(2*I+1)=RMECO
COP(2*I+1)=RMECO

```



```

QEVAP (2*I+1)=RMECO
ENDIF

```

```

IF (REP(1:1).EQ.'3') THEN
QCOND (2*I+1)=TEEEV
COP (2*I+1)=TEEEV
QEVAP (2*I+1)=TEEEV
ENDIF

```

```

IF (REP(1:1).EQ.'4') THEN
QCOND (2*I+1)=RMEEV
COP (2*I+1)=RMEEV
QEVAP (2*I+1)=RMEEV
ENDIF

```

```

c
c Sauvetage des ordonnees
c
QCOND (2*(I+1))=QCO
QEVAP (2*(I+1))=QEV
COP (2*(I+1))=(QCO)/(QCO-QEV)

10 CONTINUE

RETURN
END

```

MENUS . FOR

```

-----
c
c      subroutine menuc (rep)
-----
c
c Menu de choix, apres une simulation, du dessin de la PAC,
c d'une nouvelle simulation ou du retour au menu principal
c
implicit integer*2 (v)
common /graph/ vxmax, vymax,
*              devhandle

character*80 texte, rep

VCOUL=1
TEXTE='1.CALCUL D''UN NOUVEAU CYCLE    2.SHEMA PAC    3.QUITTER'
NCOUNT=58
CALL INPUT (VCOUL, NCOUNT, 0., 10., TEXTE, REP)

RETURN
END

```

```

-----
c
c      subroutine menud (rep)
-----
c
c Menu de choix, apres une simulation, d'affichage du diagramme TS,
c de nouvelle simulation ou de retour au menu principal
c
implicit integer*2 (v)
common /graph/ vxmax, vymax,
*              devhandle

character*80 texte, rep

VCOUL=1
TEXTE='1.CALCUL D''UN NOUVEAU CYCLE    2.DIAGRAMME T-S '
TEXTE(48:74)='3.RETOUR AU MENU PRINCIPAL'
NCOUNT=74
CALL INPUT (VCOUL, NCOUNT, 0., 3., TEXTE, REP)

RETURN
END

```

```

-----
c
c      subroutine menu3 (rep)
-----
c
c Menu principal de choix d'un type de simulation (fixe ou
c a parametre variable) ou de sortie du programme
c
implicit integer*2 (v)
common /graph/ vxmax, vymax,

```

```

*          devhandle

character*80 texte,rep

VCOUL=1
texte='
TEXTE(60:80)='
ncount=80
CALL PRNT(VCOUL,NCOUNT,0.,0.,TEXTE)
CALL PRNT(VCOUL,NCOUNT,1.,0.,TEXTE)

TEXTE='1.SIMULATION 2.EVOLUTION DES PARAMETRES 3.QUITTER
NCOUNT=58
CALL INPUT(VCOUL,NCOUNT,0.,10.,TEXTE,REP)

RETURN
END

```

```

-----
subroutine menu4 (rep)
-----

```

```

c
c Menu de choix d'un parametre a faire varier dans la simulation
c
implicit integer*2 (v)
common /graph/ vxmax,vymax,
*          devhandle

character*80 texte,rep

VCOUL=1
TEXTE='QUEL PARAMETRE VOULEZ-VOUS COMME VARIABLE
NCOUNT=58
CALL PRNT(VCOUL,NCOUNT,0.,1.,TEXTE)

VCOUL=1
TEXTE='1.T ent eau cond 2.Debit eau cond
TEXTE(37:88)='3.T ent eau evap 4.Debit eau evap
NCOUNT=88
CALL INPUT(VCOUL,NCOUNT,1.,1.,TEXTE,REP)

RETURN
END

```

```

-----
subroutine menu5 (rep)
-----

```

```

c
c Menu de choix d'une variable dont on veut dessiner la variation
c en fonction du parametre dans le cas d'une simulation
c a parametre variable
c
implicit integer*2 (v)
common /graph/ vxmax,vymax,
*          devhandle

character*80 texte,rep

VCOUL=1
texte='
TEXTE(60:80)='
ncount=80
CALL PRNT(VCOUL,NCOUNT,1.,1.,TEXTE)

```

```

CALL PRNT(VCOUL,NCOUNT,0.,1.,TEXTE)

TEXTE='1.COP 2.Puiss. cond 3.Puiss. evap 4.RETOUR AU MENU PRIN
&CIPAL '
NCOUNT=65
CALL INPUT(VCOUL,NCOUNT,0.,9.,TEXTE,REP)

RETURN
END

```

```

-----
subroutine entrel
-----

```

```

c
c Entree des parametres (temperatures, debits,...) dans le cas
c du regime "Vitesse du compresseur libre"
c

```

```

implicit integer*2 (v)
COMMON/INOUT/QC,TC,TSEEV,TEEEV,RMEEV,TSECO,TEECO,RMECO
COMMON/CONSTHER/RKCO,RKEV,CPEAU,CPFREON
COMMON/COMPRESSEUR/RVE,RENDCOMP
COMMON/REGIME/IQCOND,IREGIME
COMMON/CYCLE/TPA,TPB,TPC,TPD,TPE,TPG,
*          RIA,RIB,RIC,RID,RIE,RIG,
*          SA,SB,SC,SD,SE,SG,
*          PC,PE
COMMON/PUISS/QEV,QCO,TCOMP,REND,REMF,RNCOMP
common /graph/ vxmax,vymax,
*          devhandle

```

```

character*80 texte,rep

```

```

QC=0.
TSEEV=0.
TEEEV=0.
RMEEV=0.
TSECO=0.
TEECO=0.
RMECO=0.
RNCOMP=0.

```

```

IQCOND=0

```

```

vcoul=2
TEXTE='T(dg C)='
CALL RINPUT(VCOUL,8,56.,81.,TEXTE,REP)
TEECO=TTN(REP)

```

```

TEXTE='Q(kg/s)='
call rinput(vcoul,8,56.,86.,texte,rep)
RMECO=TTN(REP)

```

```

TEXTE='T(dg C)='
CALL RINPUT(VCOUL,8,11.,81.,TEXTE,REP)
TSECO=TTN(REP)

```

```

VCOUL=7
TEXTE='T(dg C)='
CALL rinput(VCOUL,8,11.,5.0,TEXTE,rep)
TEEEV=TTN(REP)

```

```

TEXTE='Q(kg/s)='
CALL rinput(VCOUL,8,11.,2.0,TEXTE,rep)

```

```

RMEEV=TTN(REP)

VCOUL=1
TEXTE='Vb (cm3)='
call rinput(vcoul,8,0.,47.5,texte,rep)
RVP=1.0E-06*TTN(REP)

RETURN
END

```

```

-----
c      subroutine entre2
-----

```

```

c
c Entree des parametres (temperatures, debits,...) dans le cas
c du regime "Vitesse du compresseur imposee"
c

```

```

implicit integer*2 (v)
COMMON/INOUT/QC,TC,TSEEV,TEEEV,RMEEV,TSECO,TEECO,RMECO
COMMON/CONSTHER/RKCO,RKEV,CPEAU,CPFREON
COMMON/COMPRESSEUR/RVP,RENDCOMP
COMMON/REGIME/IQCOND,IREGIME
COMMON/CYCLE/TPA,TPB,TPC,TPD,TPE,TPG,
*      RIA,RIB,RIC,RID,RIE,RIG,
*      SA,SB,SC,SD,SE,SG,
*      PC,PE
COMMON/PUISS/QEV,QCO,TCOMP,RENDEMENT,RMF,RNCOMP
common /graph/ vxmax,vymax,
*      devhandle

```

```
character*80 texte,rep
```

```

QC=0.
TSEEV=0.
TEEEV=0.
RMEEV=0.
TSECO=0.
TEECO=0.
RMECO=0.
RNCOMP=0.

```

```
IQCOND=0
```

```

vcoul=2
TEXTE='T (dg C)='
CALL RINPUT(VCOUL,8,56.,81.,TEXTE,REP)
TEECO=TTN(REP)

```

```

TEXTE='Q (kg/s)='
call rinput(vcoul,8,56.,86.,texte,rep)
RMECO=TTN(REP)

```

```

VCOUL=7
TEXTE='T (dg C)='
CALL rinput(VCOUL,8,11.,5.0,TEXTE,rep)
TEEEV=TTN(REP)

```

```

TEXTE='Q (kg/s)='
CALL rinput(VCOUL,8,11.,2.0,TEXTE,rep)
RMEEV=TTN(REP)

```

```

VCOUL=1
TEXTE='Vb (cm3)='

```

```

call rinput(vcoul,8,0.,47.5,texte,rep)
RVP=1.0E-6*TTN(REP)

```

```

TEXTE='N(t/s)='
call rinput(vcoul,7,0.,42.5,texte,rep)
RNCOMP=TTN(REP)

```

```

RETURN
END

```

```

-----
c      subroutine entre3 (repl)
-----

```

```

c
c Entree des parametres (temperatures, debits,...) fixes et des
c bornes de variation des parametres variables dans le cas d'une
c simulation a parametre variable
c

```

```

implicit integer*2 (v)
COMMON/INOUT/QC,TC,TSEEV,TEEEV,RMEEV,TSECO,TEECO,RMECO
COMMON/INOUT2/TEEEV2,RMEEV2,TEECO2,RMECO2
COMMON/CONSTHER/RKCO,RKEV,CPEAU,CPFREON
COMMON/COMPRESSEUR/RVP,RENDCOMP
COMMON/REGIME/IQCOND,IREGIME
COMMON/CYCLE/TPA,TPB,TPC,TPD,TPE,TPG,
*      RIA,RIB,RIC,RID,RIE,RIG,
*      SA,SB,SC,SD,SE,SG,
*      PC,PE
COMMON/PUISS/QEV,QCO,TCOMP,RENDEMENT,RMF,RNCOMP
common /graph/ vxmax,vymax,
*      devhandle

```

```
character*80 texte,rep,rep1
```

```

QC=0.
TSEEV=0.
TEEEV=0.
RMEEV=0.
TSECO=0.
TEECO=0.
RMECO=0.
RNCOMP=0.

```

```
IQCOND=0
```

```
c
```

```
vcoul=2
```

```

if (repl(1:1).ne.'1') then
TEXTE='T (dg C)='
CALL RINPUT(VCOUL,8,56.,81.,TEXTE,REP)
TEECO=TTN(REP)
TEECO2=TEECO

```

```

ELSE
TEXTE='T1 (dg C)='
CALL RINPUT(VCOUL,9,40.,81.,TEXTE,REP)
TEECO=TTN(REP)
TEXTE='T2='
CALL RINPUT(VCOUL,3,65.,81.,TEXTE,REP)
TEECO2=TTN(REP)
endif

```

```

C -----
if (repl(1:1).ne.'2') then
  TEXTE= 'Q(kg/s)='
  CALL RINPUT(VCOUL,8,56.,86.,TEXTE,REP)
  RMECO=TTN(REP)
  RMECO2=RMECO

  ELSE
  TEXTE='Q1(kg/s)='
  CALL RINPUT(VCOUL,9,40.,86.,TEXTE,REP)
  RMECO=TTN(REP)
  TEXTE='Q2='
  CALL RINPUT(VCOUL,3,65.,86.,TEXTE,REP)
  RMECO2=TTN(REP)
endif

C -----
VCOUL=7

if (repl(1:1).ne.'3') then
  TEXTE='T(dg C)='
  CALL RINPUT(VCOUL,8,11.,5.0,TEXTE,REP)
  TEEEV=TTN(REP)
  TEEEV2=TEEEV

  ELSE
  TEXTE='T1(dg C)='
  CALL RINPUT(VCOUL,9,0.,5.0,TEXTE,REP)
  TEEEV=TTN(REP)
  TEXTE='T2='
  CALL RINPUT(VCOUL,3,25.,5.0,TEXTE,REP)
  TEEEV2=TTN(REP)
endif

C -----
if (repl(1:1).ne.'4') then
  TEXTE='Q(kg/s)='
  CALL RINPUT(VCOUL,8,11.,2.0,TEXTE,REP)
  RMEEV=TTN(REP)
  RMEEV2=RMEEV

  ELSE
  TEXTE= 'Q1(kg/s)='
  CALL RINPUT(VCOUL,9,0.,2.0,TEXTE,REP)
  RMEEV=TTN(REP)
  TEXTE='Q2='
  CALL RINPUT(VCOUL,3,25.,2.0,TEXTE,REP)
  RMEEV2=TTN(REP)
endif

C -----
VCOUL=1
TEXTE='Vb(cm3)='
call rinput(vcoul,8,0.,47.5,texte,rep)
RVP=1.0E-6*TTN(REP)

TEXTE='N(t/s)='
call rinput(vcoul,7,0.,42.5,texte,rep)
RNCOMP=TTN(REP)
RETURN
END

```

```

C -----
subroutine qregime(rep)
C -----
C
C Menu de choix du regime : vitesse du compresseur libre ou imposee
C
  implicit integer*2 (v)
  common /graph/ vxmax,vymax,
  * devhandle

  character*80 texte,rep

  VCOUL=1
  STATUS=VSTCOL(DEVHANDLE,VCOUL)

  TEXTE='IL EXISTE DEUX REGIMES:1. VITESSE DU COMPRESSEUR LIBRE '
  NCOUNT=56
  CALL PRNT (VCOUL,NCOUNT,0.,0.,TEXTE)

  TEXTE=' :2. VITESSE DU COMPRESSEUR IMPOSEE'
  CALL PRNT (VCOUL,NCOUNT,1.,0.,TEXTE)

  TEXTE='FAITES VOTRE CHOIX:'
  ncount=19
  CALL INPUT(VCOUL,NCOUNT,2.,0.,TEXTE,REP)

  RETURN
  END

```

FICHIERS.FOR

```

-----
c      subroutine openfic
-----
c      Ouverture des fichiers
c
      OPEN(11,FILE='TABLE.DAT')
      OPEN(12,FILE='PACDONSI.DAT')
      OPEN(14,FILE='PACSORSI.DAT')
      OPEN(10,FILE='CYCLE.DAT')
      RETURN
      END
c
-----
c      subroutine closfic
-----
c      Fermeture des fichiers
c
      CLOSE(11)
      CLOSE(12)
      CLOSE(14)
      CLOSE(10)
      RETURN
      END
c
-----
c      subroutine sortie
-----
c      Ecriture des resultats dans PACSORSI
c
      COMMON/INOUT/QC, TC, TSEEV, TEEEV, RMEEV, TSECO, TEECO, RMECO
      COMMON/CONSTHER/RKCO, RKEV, CPEAU, CPFREON
      COMMON/COMPRESSEUR/RVP, RENDCOMP
      COMMON/REGIME/IQCOND, IREGIME
      COMMON/CYCLE/TPA, TPB, TPC, TPD, TPE, TPG,
      *          RIA, RIB, RIC, RID, RIE, RIG,
      *          SA, SB, SC, SD, SE, SG,
      *          PC, PE
      COMMON/PUISS/QEV, QCO, TCOMP, REND, RMF, RNCOMP

      WRITE(14,8000) TEECO, TSECO, RMECO, TPA
      WRITE(14,8010) TEEEV, TSEEV, RMEEV, TPB
      WRITE(14,8020) RKCO
      WRITE(14,8030) RKEV
8000 FORMAT(' AU CONDENSEUR  TEMPERATURE D ENTREE DE L EAU :',
      *F6.1, 'Dg Cel',/,
      *          ' TEMPERATURE DE SORTIE DE L EAU:',
      *F6.1, 'Dg Cel',/,
      *          ' DEBIT DE L EAU :',F6.1, 'Kg/Sec',/,
      *          ' TEMPERATURE DE CONDENSATION :',
      *F6.1, 'Dg Cel',/)

```

```

8010 FORMAT(' A L EVAPORATEUR  TEMPERATURE D ENTREE DE L EAU:',
      *F6.1, 'Dg Cel',/,
      *          ' TEMPERATURE DE SORTIE DE L EAU:',
      *F6.1, 'Dg Cel',/,
      *          ' DEBIT DE L EAU :',F6.1, 'Kg/Sec',/,
      *          ' TEMPERATURE D EVAPORATION:',
      *F6.1, 'Dg Cel',/)
8020 FORMAT(' COEFFICIENT DE TRANSMISSION THERMIQUE AU',
      *          ' CONDENSEUR:',G10.3, 'kJ/Dg/Sec')
8030 FORMAT(' COEFFICIENT DE TRANSMISSION THERMIQUE A',
      *          ' L EVAPORATEUR:',G10.3, 'kJ/Dg/Sec',/)
      WRITE(14,8990)
      WRITE(14,9000) TPA, RIA, SA
      WRITE(14,9010) TPB, RIB, SB
      WRITE(14,9020) TPC, RIC, SC
      WRITE(14,9025) TPE, RIE, SE
      WRITE(14,9030) TPD, RID, SD
      WRITE(14,9040) TPG, RIG, SG
      WRITE(14,9045) PC, PE
9045 FORMAT(/, ' PRESSION AU CONDENSEUR=',G10.3, ' MPA',/,
      *          ' PRESSION A L EVAPORATEUR=',G10.3, ' MPA',/)
8990 FORMAT('          Dg Cel      kJ/kg      kJ/kg/Dg ')
9000 FORMAT(' POINT A      T=',F5.1, ' I=',F6.2, ' S=',F7.4)
9010 FORMAT(' POINT B      T=',F5.1, ' I=',F6.2, ' S=',F7.4)
9020 FORMAT(' POINT C      T=',F5.1, ' I=',F6.2, ' S=',F7.4)
9025 FORMAT(' POINT E      T=',F5.1, ' I=',F6.2, ' S=',F7.4)
9030 FORMAT(' POINT D      T=',F5.1, ' I=',F6.2, ' S=',F7.4)
9040 FORMAT(' POINT G      T=',F5.1, ' I=',F6.2, ' S=',F7.4)
9050 FORMAT(' CHALEUR FOURNIE PAR LE CONDENSEUR :',
      *          G10.4, 'kJ/Sec')
9060 FORMAT(' CHALEUR ABSORBEE A L EVAPORATEUR :',
      *          G10.4, 'kJ/Sec')
9070 FORMAT(' PUISSANCE CONSOMMEE PAR LE COMPRESSEUR :',
      *          G10.4, 'kJ/Sec')
9080 FORMAT(' RENDEMENT THEORIQUE :',G10.4)
9090 FORMAT(' DEBIT DE FREON :',G10.4, ' kg/Sec')
9100 FORMAT(' VITESSE DU COMPRESSEUR :',G10.4, ' Tr/Sec')
9105 FORMAT(' VOLUME DE BALAYAGE DU COMPRESSEUR:',G10.4, 'm**3')
9106 FORMAT(' RENDEMENT THERMIQUE DU COMPRESSEUR:',G10.3,/,
      *          ' ',/)
      WRITE(14,9050) QCO
      WRITE(14,9060) QEV
      WRITE(14,9070) TCOMP
      WRITE(14,9080) REND
      WRITE(14,9090) RMF
      WRITE(14,9100) RNCOMP
      write(14,9105) RVP
      WRITE(14,9106) RENDCOMP

      RETURN
      END
c
-----
c      subroutine lecpadon
-----
c      Lecture dans le fichier PACDONSI.DAT de
c      RKCO      : Coefficient d'echange thermique au condenseur
c      RKEV      : Coefficient d'echange thermique a l'evaporateur
c      CPEAU     : Chaleur specifique du fluide caloporteur
c      CPFREON   : Chaleur specifique du freon
c      RENDCOMP  : Rendement du compresseur
c

```

```

COMMON/INOUT/QC, TC, TSEEV, TEEV, RMEEV, TSECO, TEECO, RMECO
COMMON/CONSTHER/RKCO, RKEV, CPEAU, CPFREON
COMMON/COMPRESSEUR/RVP, RENDCOMP
COMMON/REGIME/IQCOND, IREGIME
COMMON/CYCLE/TPA, TPB, TPC, TPD, TPE, TPG,
*      RIA, RIB, RIC, RID, RIE, RIG,
*      SA, SB, SC, SD, SE, SG,
*      PC, PE
COMMON/PUISS/QEV, QCO, TCOMPTh, RENDEMENT, RMF, RNCOMP

CHARACTER CAR

READ (12, 9000) CAR
READ (12, *) RKCO, RKEV, CPEAU, CPFREON, RENDCOMP
9000 FORMAT (A1)

RETURN
END

C
C-----
C      subroutine cretable
C-----
C
C Ecriture dans TABLE.DAT des valeurs d'enthalpie, d'entropie et de pression
C du freon (etats liquide et vapeur) pour les temperatures -100 / 110 deg.C
C
WRITE (11, 9010)
9010 FORMAT (' T      P      VS      IP      IS      SP      SS', /,
*      'Dg C      MPA      M**3/Kg      kJ/Kg      kJ/Kg      kJ/Kg/K      kJ/Kg/K')
DO 10 I=-100,110,5
T=FLOAT(I)
P=PSAT(T)
VS=RVAPSAT(T)
RIP=RILIQSAT(T)
RIS=RIVAPSAT(T)
SP=SLIQSAT(T)
SS=SVAPSAT(T)
WRITE (11, 9000) T, P, VS, RIP, RIS, SP, SS
10 CONTINUE

9000 FORMAT (F4.0, 2X, F7.5, 2X, F6.4, 2X, F6.2, 2X, F6.2, 2X, F7.5, 2X, F7.5)
END

C
C-----
C      subroutine savcycl
C-----
C
C Sauvetage dans CYCLE.DAT des temperatures et des entropies aux
C differents endroits du cycle pour la simulation envisagee
C
common /cycle/tpa, tpb, tpc, tpd, tpe, tpg,
*      ria, rib, ric, rid, rie, rig,
*      sa, sb, sc, sd, se, sg,
*      pc, pe

write (10, 9000) tpa, tpb, tpc, tpd, tpe, tpg
write (10, 9000) sa, sb, sc, sd, se, sg
9000 format (6g10.4)

return
end

```

DESSIN.FOR

```

C-----
C      subroutine INIT_GRAPH
C-----
C
C      Initialisation du driver graphique
C
C-----
C
implicit integer*2 (v)
common /graph/ vxmax, vymax, devhandle
dimension vsavary (66), vechoxy (2), vnominate (19)
data vnominate /0,1,1,3,1,1,1,0,0,1,1,68,73,83,80,76,65,89,32/
data vechoxy /0,0/

status=vopnwk (vnominate, devhandle, vsavary)
vxmax=vsavary (52)
vymax=vsavary (53)
return
end

C
C-----
C      subroutine logo
C-----
C
C Dessin de la PAC , logo FUL , ...
C
C-----
C
implicit integer*2 (v)
common /graph/vxmax, vymax, devhandle
character*80 texte, rep
vstat=vclrwk (devhandle)
call dessin
texte (1:3)='PAC'
ncount=3
vcoul=2
vreq=10000
vstat=vsthgt (devhandle, vreq, vwid, vcwid, vhgt)
call prnt (vcoul, ncount, 3., 50., texte)
vreq=2000
vcoul=5
texte (1:29)='SIMULATION DE POMPE A CHALEUR'
ncount=29
vstat=vsthgt (devhandle, vreq, vwid, vcwid, vhgt)
call prnt (vcoul, ncount, 10., 0., texte)
vcoul=3
texte (1:27)=' F.U.L. - ARLON - BELGIQUE'
ncount=27
call prnt (vcoul, ncount, 15., 0., texte)
vreq=1000
vcoul=7

```

```

texte(1:30)='PRESSEZ "ENTER" POUR CONTINUER'
NCOUNT=30
vstat=vsthgt(devhandle,vreq,vwid,vswid,vhgt)
call prnt(vcoul,ncount,20.,50.,texte)
ncount=0
call rinput(vcoul,ncount,0.,0.,texte,rep)
vstat=vclrwk(devhandle)
return
end

```

```

-----
subroutine diagts (nt,ns,nsimul,XP1,XP2,YP1,YP2)
-----
-
c Dessin du diagramme TS du freon considere
c et du cycle pour la simulation envisagee
c
implicit integer*2 (v)
common /graph/ vxmax,vymax,devhandle
dimension xy(100),vxy(100),vaxe(4)
character*80 texte,texte2

if ((xp1.lt.15.).and.(xp2.gt.85.).and.
* (yp1.lt.15.).and.(yp2.gt.85.)) then
status=vclrwk(devhandle)
endif

VCOUL=2

T=-100.0
DO 10 I=1,43,2
XY(I)=SLIQSAT(T)
XY(I+1)=T
T=T+10.0
10 CONTINUE

T= 110.0
DO 20 I=45,87,2
XY(I)=SVAPSAT(T)
XY(I+1)=T
T=T-10.0
20 CONTINUE

texte='s(J/kg dgC) '
ncs=11
texte2='T(dgC) '
nct=6
vcoul=7
CALL tracf(devhandle,vcoul,vxmax,vymax,
* ns,ncs,texte,nt,nct,texte2,
* XP1,XP2,YP1,YP2,xy,44)

C ----- AFFICHAGE DES CYCLES-----

vcoul=0

REWIND (10)

do 1000 i=1,nsimul
30 READ (10,9000) TPA,TPB,TPC,TPD,TPE,TPG
READ (10,9000) SA,SB,SC,SD,SE,SG
9000 format(6g10.4)

xy(1)=sa
xy(2)=tpa
xy(3)=sb
xy(4)=tpb
xy(5)=sc
xy(6)=tpc
xy(7)=se
xy(8)=tpe
xy(9)=sd

```

```

xy(10)=tpd
xy(11)=sg
xy(12)=tpg
xy(13)=xy(1)
xy(14)=xy(2)

VCOUL=VCOUL+1
status=vslcol(devhandle,vcoul)
call transf(xy,7,vxy)
vn=7
status=vpline(devhandle,vn,vxy)

1000 continue

RETURN
END
c
-----
c      subroutine dessin
-----
c
c
c Dessin de la Pompe a Chaleur
c
c      implicit integer*2 (v)
c      common /graph/ vxmax,vymax,devhandle
c      dimension xy(100),vxy(100)
c      character*80 texte,rep

VCOUL=2

status=vclrwk(devhandle)

CALL WINDOW(VXMAX,VYMAX,0.,70.,0.0,100.,
*          0.,100.,0.,100.)

1  XY(1)=25.
   XY(2)=10.
   XY(3)=25.
   XY(4)=20.
   XY(5)=38.75
   XY(6)=20.
   XY(7)=40.
   XY(8)=22.5
   XY(9)=42.5
   XY(10)=17.5
   XY(11)=45.
   XY(12)=22.5
   XY(13)=47.5
   XY(14)=17.5
   XY(15)=50.0
   XY(16)=22.5
   XY(17)=52.5
   XY(18)=17.5
   XY(19)=55.0
   XY(20)=22.5
   XY(21)=56.25
   XY(22)=20.0
   XY(23)=70.0
   XY(24)=20.0
   XY(25)=70.0
   XY(26)=10.0

```

```

VCOUL=7
STATUS=VSLCOL (DEVHANDLE, VCOUL)
N=13
VN=N
CALL TRANSF (XY, N, VXY)
STATUS=VPLINE (DEVHANDLE, VN, VXY)
      fleche
C
XY(1)=30.
XY(2)=20.
XY(3)=27.5
XY(4)=21.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)
XY(3)=27.5
XY(4)=18.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)

STATUS=VPLINE (DEVHANDLE, VN, VXY)
XY(1)=25.
XY(2)=40.
XY(3)=25.
XY(4)=25.
XY(5)=38.75
XY(6)=25.
XY(7)=40.
XY(8)=27.5
XY(9)=42.5
XY(10)=22.5
XY(11)=45.
XY(12)=27.5
XY(13)=47.5
XY(14)=22.5
XY(15)=50.0
XY(16)=27.5
XY(17)=52.5
XY(18)=22.5
XY(19)=55.0
XY(20)=27.5
XY(21)=56.25
XY(22)=25.0
XY(23)=70.0
XY(24)=25.0
XY(25)=70.0
XY(26)=40.0
VCOUL=3
STATUS=VSLCOL (DEVHANDLE, VCOUL)
N=13
VN=N
CALL TRANSF (XY, N, VXY)
STATUS=VPLINE (DEVHANDLE, VN, VXY)
      fleche
C
XY(1)=65.
XY(2)=25.
XY(3)=67.5
XY(4)=26.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)
XY(3)=67.5

```



```

XY(4)=23.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

```

XY(1)=25.
XY(2)=50.
XY(3)=25.
XY(4)=65.
XY(5)=38.75
XY(6)=65.
XY(7)=40.
XY(8)=67.5
XY(9)=42.5
XY(10)=62.5
XY(11)=45.
XY(12)=67.5
XY(13)=47.5
XY(14)=62.5
XY(15)=50.0
XY(16)=67.5
XY(17)=52.5
XY(18)=62.5
XY(19)=55.0
XY(20)=67.5
XY(21)=56.25
XY(22)=65.0
XY(23)=70.0
XY(24)=65.0
XY(25)=70.0
XY(26)=50.0
VCOUL=3
STATUS=VSLCOL (DEVHANDLE, VCOUL)
N=13
VN=N
CALL TRANSF (XY, N, VXY)
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

C

```

fleche
XY(1)=30.
XY(2)=65.
XY(3)=27.5
XY(4)=66.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)
XY(3)=27.5
XY(4)=63.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

```

XY(1)=25.
XY(2)=80.
XY(3)=25.
XY(4)=70.
XY(5)=38.75
XY(6)=70.
XY(7)=40.
XY(8)=72.5

```

```

XY(9)=42.5
XY(10)=67.5
XY(11)=45.
XY(12)=72.5
XY(13)=47.5
XY(14)=67.5
XY(15)=50.0
XY(16)=72.5
XY(17)=52.5
XY(18)=67.5
XY(19)=55.0
XY(20)=72.5
XY(21)=56.25
XY(22)=70.0
XY(23)=70.0
XY(24)=70.0
XY(25)=70.0
XY(26)=80.0
VCOUL=2
STATUS=VSLCOL (DEVHANDLE, VCOUL)
N=13
VN=N
CALL TRANSF (XY, N, VXY)
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

C

```

fleche
XY(1)=65.
XY(2)=70.
XY(3)=67.5
XY(4)=71.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)
XY(3)=67.5
XY(4)=68.5
CALL TRANSF (XY, 2, VXY)
VN=2
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

```

X=25.0
Y=45.0
RADIUS=5.
STATUS=VCIRCL (DEVHANDLE, VTRX (X), VTRY (Y), VTRX (RADIUS))

```

```

CALL BOX (35.0, 60.0, 15.0, 30.0, VXY)
VCOUL=1
STATUS=VSLCOL (DEVHANDLE, VCOUL)
VN=5
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

```

X=70.0
Y=45.0
RADIUS=5.
STATUS=VCIRCL (DEVHANDLE, VTRX (X), VTRY (Y), VTRX (RADIUS))

```

```

CALL BOX (35.0, 60.0, 60.0, 75.0, VXY)
VCOUL=1
STATUS=VSLCOL (DEVHANDLE, VCOUL)
VN=5
STATUS=VPLINE (DEVHANDLE, VN, VXY)

```

```

STATUS=VSTCOL (DEVHANDLE, VCOUL)

```

```

vver=0
vhor=0
status=vstaln(devhandle,vhor,vver,vhorr,vverr)
texte='EVAPORATEUR'
vcount=11
status=vgtxts(devhandle,vtrx(40.),vtry(32.),vcount,texte)

vver=2
vhor=0
status=vstaln(devhandle,vhor,vver,vhorr,vverr)
texte='CONDENSEUR'
VCOUNT=10
status=vgtxts(devhandle,vtrx(40.),vtry(58.),vcount,texte)

vver=1
vhor=0
status=vstaln(devhandle,vhor,vver,vhorr,vverr)
texte='COMP'
vcount=4
status=vgtxts(devhandle,vtrx(31.),vtry(45.),vcount,texte)

vver=1
vhor=2
status=vstaln(devhandle,vhor,vver,vhorr,vverr)
texte='DET'
VCOUNT=3
status=vgtxts(devhandle,vtrx(64.),vtry(45.),vcount,texte)

RETURN
END

```

C

C

```
-----
subroutine sortgr
-----
```

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

```

Sortie graphique des resultats de la simulation
(Ecriture des valeurs sur le dessin de la pompe a chaleur)

```

```

implicit integer*2 (v)
COMMON/INOUT/QC,TC,TSEEV,TEEEV,RMEEV,TSECO,TEECO,RMECO
COMMON/CONSTHER/RKCO,RKEV,CPEAU,CPFREON
COMMON/COMPRESSEUR/RVP,RENDCOMP
COMMON/REGIME/IQCOND,IREGIME
COMMON/CYCLE/TPA,TPB,TPC,TPD,TPE,TPG,
*      RIA,RIB,RIC,RID,RIE,RIG,
*      SA,SB,SC,SD,SE,SG,
*      PC,PE
COMMON/PUISS/QEV,QCO,TCOMP,RENDEMENT,RMF,RNCOMP
common /graph/ vxmax,vymax,devhandle

```

```
character*80 texte,rep
```

```

vver=0
vhor=1
status=vstaln(devhandle,vhor,vver,vhorr,vverr)
CONDENSEUR

```

```

vcoul=2
STATUS=VSTCOL(DEVHANDLE,VCoul)
CALL CONVNT(TEECO,REP)
TEXTE='
TEXTE='T='
TEXTE(3:7)=REP(1:5)
TEXTE(8:11)='dg C'

```

```

VCOUNT=11
SATUS=VGTXTS(DEVHANDLE,VTRX(70.),VTRY(80.),VCOUNT,TEXTE)

```

```

CALL CONVNT(RMECO,REP)
TEXTE='
TEXTE='Q='
TEXTE(3:7)=REP(1:5)
TEXTE(8:11)='kg/s'
VCOUNT=11
SATUS=VGTXTS(DEVHANDLE,VTRX(70.),VTRY(85.),VCOUNT,TEXTE)

```

```

CALL CONVNT(TSECO,REP)
TEXTE='
TEXTE='T='
TEXTE(3:7)=REP(1:5)
TEXTE(8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS(DEVHANDLE,VTRX(25.),VTRY(80.),VCOUNT,TEXTE)

```

```

CALL CONVNT(QCO,REP)
TEXTE='
TEXTE='W='
TEXTE(3:7)=REP(1:5)
TEXTE(8:9)='kW'
VCOUNT=9
SATUS=VGTXTS(DEVHANDLE,VTRX(47.5),VTRY(76.),VCOUNT,TEXTE)

```

C

EVAPORATEUR

```

vver=2
vhor=1
status=vstaln(devhandle,vhor,vver,vhorr,vverr)
VCOUL=7
STATUS=VSTCOL(DEVHANDLE,VCOUL)
CALL CONVNT(TEEEV,REP)
TEXTE='
TEXTE='T='
TEXTE(3:7)=REP(1:5)
TEXTE(8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS(DEVHANDLE,VTRX(25.),VTRY(10.0),VCOUNT,TEXTE)

```

```

CALL CONVNT(RMEEV,REP)
TEXTE='
TEXTE='Q='
TEXTE(3:7)=REP(1:5)
TEXTE(8:11)='kg/s'
VCOUNT=11
SATUS=VGTXTS(DEVHANDLE,VTRX(25.),VTRY(5.0),VCOUNT,TEXTE)

```

```

CALL CONVNT(TSEEV,REP)
TEXTE='
TEXTE='T='
TEXTE(3:7)=REP(1:5)
TEXTE(8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS(DEVHANDLE,VTRX(70.),VTRY(10.0),VCOUNT,TEXTE)

```

```

CALL CONVNT(QEV,REP)
TEXTE='
TEXTE='W='
TEXTE(3:7)=REP(1:5)
TEXTE(8:9)='kW'

```

```
VCOUNT=9
SATUS=VGTXTS (DEVHANDLE, VTRX (47.5), VTRY (14.), VCOUNT, TEXTE)
```

```
C
                                COMPRESSEUR
vver=1
vhor=2
status=vstaln (devhandle, vhor, vver, vhor, vver)
VCOUL=1
STATUS=VSTCOL (DEVHANDLE, VCOUL)
tempo=rvp*1.e6
CALL CONVNT (TEMPO, REP)
TEXTE='
TEXTE='Vb='
TEXTE (4:8)=REP (1:5)
TEXTE (8:10)='cm3'
VCOUNT=10
SATUS=VGTXTS (DEVHANDLE, VTRX (18.), VTRY (47.5), VCOUNT, TEXTE)

CALL CONVNT (RNCOMP, REP)
TEXTE='
TEXTE='N='
TEXTE (3:7)=REP (1:5)
TEXTE (7:11)='tr/s'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (20.), VTRY (42.5), VCOUNT, TEXTE)
```

```
C
                                CYCLE FREON
vver=1
vhor=0
status=vstaln (devhandle, vhor, vver, vhor, vver)
VCOUL=3
STATUS=VSTCOL (DEVHANDLE, VCOUL)

CALL CONVNT (TPA, REP)
TEXTE='
TEXTE='T='
TEXTE (3:7)=REP (1:5)
TEXTE (8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (71.), VTRY (60.), VCOUNT, TEXTE)

CALL CONVNT (TPB, REP)
TEXTE='
TEXTE='T='
TEXTE (3:7)=REP (1:5)
TEXTE (8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (71.), VTRY (30.), VCOUNT, TEXTE)

vver=1
vhor=2
status=vstaln (devhandle, vhor, vver, vhor, vver)

CALL CONVNT (TPC, REP)
TEXTE='
TEXTE='T='
TEXTE (3:7)=REP (1:5)
TEXTE (8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (24.), VTRY (30.), VCOUNT, TEXTE)

CALL CONVNT (PE, REP)
TEXTE='
```

```
TEXTE='P='
TEXTE (3:7)=REP (1:5)
TEXTE (8:11)=' MPa'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (24.), VTRY (25.), VCOUNT, TEXTE)
```

```
CALL CONVNT (TPD, REP)
TEXTE='
TEXTE='T='
TEXTE (3:7)=REP (1:5)
TEXTE (8:11)='dg C'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (24.), VTRY (60.), VCOUNT, TEXTE)
```

```
CALL CONVNT (PC, REP)
TEXTE='
TEXTE='P='
TEXTE (3:7)=REP (1:5)
TEXTE (8:11)=' MPa'
VCOUNT=11
SATUS=VGTXTS (DEVHANDLE, VTRX (24.), VTRY (65.), VCOUNT, TEXTE)
```

```
RETURN
END
```

```
C
C-----
                                subroutine VARICURV (choix, rep)
C-----
C
C
C
C
C
C-----
C
implicit integer*2 (v)
character*80 rep, txtx, txty, choix
common /graph/ vxmax, vymax, devhandle
common /simul/cop (100), qcond (100), qevap (100)
if (choix (1:1).eq. '1') txtx='T ent eau cond (dg C)'
if (choix (1:1).eq. '2') txtx='Debit eau cond (kg/s)'
if (choix (1:1).eq. '3') txtx='T ent eau evap (dg C)'
if (choix (1:1).eq. '4') txtx='Debit eau evap (kg/s)'
ncx=21
vcoul=7
if (rep (1:1).eq. '1') then
status=vclrwk (devhandle)
nx=5
ny=5
txty='COP'
ncy=3
call tracf (devhandle, vcoul, vxmax, vymax,
*          nx, ncx, txtx, ny, ncy, txty,
*          10., 90., 10., 90., COP, 20)
return
ENDIF

if (rep (1:1).eq. '2') then
status=vclrwk (devhandle)
nx=5
txty='W cond (kW)'
```

```

ncy=11
ny=5
call tracf(devhandle,vcoul,vxmax,vymax,
*          nx,ncx,txtx,ny,ncy,txty,
*          10.,90.,10.,90.,qcond,20)
return
ENDIF

```

```

if (rep(1:1).eq.'3') then
status=vclrwk(devhandle)
nx=5
ny=5
txty='W evap (kW)'
ncy=11
call tracf(devhandle,vcoul,vxmax,vymax,
*          nx,ncx,txtx,ny,ncy,txty,
*          10.,90.,10.,90.,qevap,20)
return
ENDIF
end

```

```

C
C-----
C          subroutine CLOSE_GRAPH
C-----
C
C          Fermeture du driver graphique
C-----
C
C
implicit integer*2 (v)
common /graph/ vxmax,vymax,devhandle
status=vencur(devhandle)
status=vclswk(devhandle)
return
end

```

FREON12.FOR

```

C
C #####
C #
C #          CARACTERISTIQUES DU FREON R12          #
C #          (Dichlorodifluoromethane)             #
C #
C #          Fonctions ajustees en se basant sur les tableaux #
C #          de valeurs fournis dans :              #
C #          ASHRAE HANDBOOK - FUNDAMENTALS - 1985 - p. 17.5 #
C #
C #####
C-----
C          function psat(t)
C-----
C
C          Tension de vapeur saturante (MPa) en fonction de la temperature
C
P1=0.308857
P2=1.015153
P3=125.565267
P4=0.644616
P5=0.075555
P6=0.061408
P7=0.025888
P8=-0.255452
P9=-0.002657
P10=0.448804
P11=0.011746
P12=-0.353408
P13=-0.004539
P14=0.108214

PSAT=P1+P2/100*T+P3*(T/100)**2+P4*(T/100)**3+P5*(T/100)**4
*      +P6*(T/100)**5+P7*(T/100)**6+P8*(T/100)**7+P9*(T/100)**8
*      +P10*(T/100)**9+P11*(T/100)**10+P12*(T/100)**11
*      +P13*(T/100)**12+P14*(T/100)**13
RETURN
END
C
C-----
C          function rvapsat(t)
C-----
C
C          Volume de la vapeur saturante (m3/kg) en fonction de la temperature
C
RVAPSAT=0.056765-0.171381*T/100.
*      +0.321035*(T/100.)**2-0.593994*(T/100.)**3
*      +0.892463*(T/100.)**4-0.5824000*(T/100.)**5
RETURN
END
C

```

```

-----
function riliqsat(t)
-----
c
c Enthalpie du liquide sature (kJ/kg) en fonction de la temperature
c
P1=200.273002
P2=92.934772
P3=-1064.713298
P4=16.668471
P5=240.496089
P6=-54.372305
P7=-1011.955792
P8=75.866078
P9=1901.165169
P10=-27.765203
P11=-1627.491827
P12=-0.000000
P13=519.713253
P14=-4.536786

RILIQSAT=P1+P2/100*T+P3*(T/1000)**2+P4*(T/100)**3+P5*(T/100)**4
* +P6*(T/100)**5+P7*(T/100)**6+P8*(T/100)**7+P9*(T/100)**8
* +P10*(T/100)**9+P11*(T/100)**10+P12*(T/100)**11
* +P13*(T/100)**12+P14*(T/100)**13
RETURN
END

c
-----
function rivapsat(t)
-----
c
c Enthalpie de la vapeur saturante (kJ/kg) en fonction de la temperature
c
P1=352.193978
P2=45.704183
P3=1206.970975
P4=-30.763004
P5=-214.445019
P6=103.295517
P7=879.026663
P8=-142.370514
P9=-1629.380466
P10=-0.000000
P11=1375.097460
P12=113.023986
P13=-432.919592
P14=-53.700455

RIVAPSAT=P1+P2/100*T+P3*(T/1000)**2+P4*(T/100)**3+P5*(T/100)**4
* +P6*(T/100)**5+P7*(T/100)**6+P8*(T/100)**7+P9*(T/100)**8
* +P10*(T/100)**9+P11*(T/100)**10+P12*(T/100)**11
* +P13*(T/100)**12+P14*(T/100)**13
RETURN
END

c
-----
function sliqsat(t)
-----
c
c Entropie du liquide sature (kJ/kg K) en fonction de la temperature
c
P1=1.000028

```

```

P2=0.350482
P3=-2.738160
P4=-0.259306
P5=-0.080275
P6=2.186783
P7=0.515325
P8=-7.336959
P9=-1.353615
P10=11.900990
P11=1.509591
P12=-9.209584
P13=-0.594452
P14=2.732647
SLIQSAT=P1+P2/100*T+P3*(T/1000)**2+P4*(T/100)**3+P5*(T/100)**4
* +P6*(T/100)**5+P7*(T/100)**6+P8*(T/100)**7+P9*(T/100)**8
* +P10*(T/100)**9+P11*(T/100)**10+P12*(T/100)**11
* +P13*(T/100)**12+P14*(T/100)**13
RETURN
END

c
-----
function svapsat(t)
-----
c
c Entropie de la vapeur saturante (kJ/kg K) en fonction de la temperature
c
P1=1.558090
P2=-0.054401
P3=5.035037
P4=0.219573
P5=0.075782
P6=-2.140359
P7=-0.469958
P8=7.201250
P9=1.247881
P10=-11.731485
P11=-1.396166
P12=9.111004
P13=0.550498
P14=-2.712842
SVAPSAT=P1+P2/100*T+P3*(T/1000)**2+P4*(T/100)**3+P5*(T/100)**4
* +P6*(T/100)**5+P7*(T/100)**6+P8*(T/100)**7+P9*(T/100)**8
* +P10*(T/100)**9+P11*(T/100)**10+P12*(T/100)**11
* +P13*(T/100)**12+P14*(T/100)**13
RETURN
END

```

GLIB.FOR

```

-----
      subroutine tracf(devhandle,vcoul,vxmax,vymax,
*                ngradx,ncx,txtx,ngady,ncy,ttxy,
*                xlp,x2p,ylp,y2p,xy,n)
c
c cette sous-routine trace une fonction representee par n points
c (n<=50) qui sont dans le tableau xy
c xy(1) coordonnee selon l axe x du premier point
c xy(2) " " " " y " " "
c etc
c la fonction est tracee dans la fenetre xlp,x2p selon l axe x
c " " " " ylp,y2p selon l axe y
c xlp,x2p,ylp,y2p sont exprimees en % de l'ecran dont les coordonnees
c absolues maximales sont vxmax et vymax
c vcoul index de la couleur 1-8
-----
      implicit integer*2 (v)
      common/absrel/x1a,x2a,y1a,y2a,x1r,x2r,y1r,y2r
      dimension xy(100),vxy(100),vaxe(4),axe(4)
      character*80 texte,txtx,ttxy

      x1a=xlp
      x2a=x2p
      y1a=ylp
      y2a=y2p
      status=vsicol(devhandle,vcoul)
      status=vstcol(devhandle,vcoul)
c
c -----determination de la fenetre relative-----
      call polymax(xy,n,xmax,ymax)
      call polymin(xy,n,xmin,ymin)
      dx=xmax-xmin
      dy=ymax-ymin
      x1=xmin-dx*0.05
      x2=xmax+dx*0.05
      y1=ymin-dy*0.05
      y2=ymax+dy*0.05
c
c -----
      call window(vxmax,vymax,xlp,x2p,ylp,y2p,x1,x2,y1,y2)
      x1r=x1
      x2r=x2
      y1r=y1
      y2r=y2

      call transf(xy,n,vxy)

      vn=n
      status=vpline(devhandle,vn,vxy)
c
c -----axes-----
      axe(1)=xmin
      axe(2)=(ymin)
      axe(3)=(x2)
      axe(4)=(ymin)

```

```

      call transf(axe,2,vaxe)
      vm=2
      status = vpline(devhandle,vm,vaxe)
      status=vstcol(devhandle,vcoul)
      vver=0
      vhor=2
      status=vstaln(devhandle,vhor,vver,vvhor,vvver)
      vncx=ncx
      vx=vtrx(x2)
      vy=vtry(ymin+(ymax-ymin)/40.)
      status=vgtxts(devhandle,vx,vy,vncx,txtx)

      axe(3)=(xmin)
      axe(4)=(y2)
      call transf(axe,2,vaxe)
      status = vpline(devhandle,vm,vaxe)
      vver=2
      vhor=0
      status=vstaln(devhandle,vhor,vver,vvhor,vvver)
      vncy=ncy
      vx=vtrx(xmin+(xmax-xmin)/40.)
      vy=vtry(y2)
      status=vgtxts(devhandle,vx,vy,vncy,ttxy)

c-----affichage sur l axe X -----
      vheight=(vaxe(4)-vaxe(2))/40
      status=vsthgt(devhandle,vheight,vcharw,vcellw,vcellh)
      vhor=1
      vver=2
      status=vstaln(devhandle,vhor,vver,vvhor,vvver)

      do 60 i=1,ngradx
      x=xmin+(i-1)*(xmax-xmin)/(ngradx-1)
      y=ymin-(ymax-ymin)/40

      call convnt(x,texte)
      vcount=5
      status=vgtxts(devhandle,vtrx(x),vtry(y),vcount,texte)

60    continue

c
c do 90 i=1,ngradx
c xy(1)=xmin+(i-1)*(xmax-xmin)/(ngradx-1)
c xy(2)=ymin
c xy(3)=xy(1)
c xy(4)=ymin-(ymax-ymin)/40.
c call transf(xy,2,vxy)
c vn=2
c status=vpline(devhandle,vn,vxy)
c90  continue

      do 90 i=1,ngradx
      vmode=8
      status=vswrmd(devhandle,vmode)
      vtype=3
      status=vsityp(devhandle,vtype)
      xy(1)=xmin+(i-1)*(xmax-xmin)/(ngradx-1)
      xy(2)=ymin
      xy(3)=xy(1)
      xy(4)=ymax

```

```

call transf(xy,2,vxy)
vn=2
status=vpline(devhandle,vn,vxy)
vmode=4
status=vswrmd(devhandle,vmode)
vtype=1
status=vslytp(devhandle,vtype)
90 continue

c-----affichage sur l axe Y -----
vhor=2
vver=1
status=vstaln(devhandle,vhor,vver,vhorr,vverr)

do 160 i=1,ngrady
y=ymin+(i-1)*(ymax-ymin)/(ngrady-1)
x=xmin-(xmax-xmin)/40

call convnt(y,texte)
vcount=4
status=vgtxts(devhandle,vtrx(x),vtry(y),vcount,texte)

160 continue

c do 190 i=1,ngrady
c xy(1)=xmin
c xy(2)=ymin+(i-1)*(ymax-ymin)/(ngrady-1)
c xy(3)=xmin-(xmax-xmin)/40.
c xy(4)=xy(2)
c call transf(xy,2,vxy)
c vn=2
c status=vpline(devhandle,vn,vxy)
c190 continue

do 190 i=1,ngrady
vmode=8
status=vswrmd(devhandle,vmode)
vtype=3
status=vslytp(devhandle,vtype)
xy(1)=xmin
xy(2)=ymin+(i-1)*(ymax-ymin)/(ngrady-1)
xy(3)=xmax
xy(4)=xy(2)
call transf(xy,2,vxy)
vn=2
status=vpline(devhandle,vn,vxy)
vtype=1
status=vslytp(devhandle,vtype)
vmode=4
status=vswrmd(devhandle,vmode)
190 continue

return
end

```

```

c-----
subroutine polymax (xy,n,xmax,ymax)
c-----
dimension xy(100)
xmax=xy(1)
ymax=xy(2)
nn=2*n
do 10 i=3,nn-1,2
if (xy(i).gt.xmax) xmax=xy(i)
if (xy(i+1).gt.ymax) ymax=xy(i+1)
10 continue
return
end

c-----
subroutine polymin (xy,n,xmin,ymin)
c-----
dimension xy(100)
xmin=xy(1)
ymin=xy(2)
nn=2*n
do 10 i=3,nn-1,2
if (xy(i).lt.xmin) xmin=xy(i)
if (xy(i+1).lt.ymin) ymin=xy(i+1)
10 continue
return
end

c-----
subroutine window (vxmax,vymax,xlp,x2p,ylp,y2p,
* xlr,x2r,y1r,y2r)
c cette sous-routine etablit la correspondance entre les
c coordonnes absolues et relatives
c xlp,x2p,y1p,y2p coordonnes absolues exprimees en %
c xlr,x2r,y1r,y2r coordonnes relatives correspondantes
c vxmax,vymax coordonnes absolues maximales permises par l'ecran
c-----
implicit integer*2 (v)
common /wind/ x1a,x2a,y1a,y2a,x1,x2,y1,y2

x1=x1r
x2=x2r
y1=y1r
y2=y2r

x1a=vxmax/100*xlp
x2a=vxmax/100*x2p
y1a=vymax/100*ylp
y2a=vymax/100*y2p

call wtransf
z= wvtrx(10.)
z= wvtry(10.)
return
end

c-----
subroutine wtransf
c-----
implicit integer*2 (v)
common /wind/ x1a,x2a,y1a,y2a,x1,x2,y1,y2
save /wind/
dimension xy(100),vxy(100)

z=1.
return

```

```

-----
c      entry  transf (xy,n,vxy)
c      cette sous-routine transforme un tableau xy de dimension 2*n
c      representant n points en coordonnees relatives en un tableau vxy
c      representant n points en coordonnees absolues
-----

```

```

      nn=2*n

      do 20 i=1,nn-1,2
      xa=x2a+(xy(i)-x2)*(x1a-x2a)/(x1-x2)
      ya=y2a+(xy(i+1)-y2)*(y1a-y2a)/(y1-y2)
      vxy(i)=xa
      vxy(i+1)=ya
20    continue

      return
      end

```

```

-----
c      function  wvtrx(x)
-----

```

```

      implicit integer*2 (v)
      common /wind/ x1a,x2a,y1a,y2a,x1,x2,y1,y2
      save /wind/

```

```

      wvtrx=x
      return

```

```

-----
c      entry  vtrx(x)

```

```

c      cette fonction transforme la coordonnees relatives selon l axe x
c      en coordonnees absolue
-----

```

```

      xv=x2a+(x-x2)*(x1a-x2a)/(x1-x2)
      vtrx=xv

```

```

      return
      end

```

```

-----
c      function  wvtry(y)
-----

```

```

      implicit integer*2 (v)
      common /wind/ x1a,x2a,y1a,y2a,x1,x2,y1,y2
      save /wind/

```

```

      wvtry=y
      return

```

```

-----
c      entry  vtry(y)

```

```

c      cette fonction transforme la coordonnees relatives selon l axe y
c      en coordonnees absolue
-----

```

```

      yv=y2a+(y-y2)*(y1a-y2a)/(y1-y2)
      vtry=yv

```

```

      return
      end

```

```

-----
c      SUBROUTINE BOX (XMIN, XMAX, YMIN, YMAX, VXY)
-----

```

```

      implicit integer*2 (v)
      common /graph/ vxmax,vymax,
      *          devhandle
      dimension vxy(100)

```

```

      VXY(1)=VTRX(XMIN)
      VXY(2)=VTRY(YMIN)
      VXY(3)=VTRX(XMAX)
      VXY(4)=VXY(2)
      VXY(5)=VXY(3)
      VXY(6)=VTRY(YMAX)
      VXY(7)=VXY(1)
      VXY(8)=VXY(6)
      VXY(9)=VXY(1)
      VXY(10)=VXY(2)

```

```

      RETURN
      END

```

```

-----
c      SUBROUTINE ARROW(VCOUL,X1,Y1,X2,Y2,X3,Y3,X4,Y4)
-----

```

```

      implicit integer*2 (v)
      common /graph/ vxmax,vymax,
      *          devhandle
      dimension vxy(100)

```

```

      STATUS=VSLCOL(DEVHANDLE,VCOUL)

```

```

      VXY(1)=VTRX(X1)
      VXY(2)=VTRY(Y1)
      VXY(3)=VTRX(X2)
      VXY(4)=VTRY(Y2)
      VN=2

```

```

      STATUS=VPLINE(DEVHANDLE,VN,VXY)

```

```

      VXY(1)=VTRX(X3)
      VXY(2)=VTRY(Y3)

```

```

      STATUS=VPLINE(DEVHANDLE,VN,VXY)

```

```

      VXY(1)=VTRX(X4)
      VXY(2)=VTRY(Y4)

```

```

      VN=2

```

```

      STATUS=VPLINE(DEVHANDLE,VN,VXY)

```

```

      RETURN
      END

```

```

-----
c      subroutine input(vcou,ncount,xlig,xcol,texte,rep)
-----

```

```

      implicit integer*2 (v)
      common /graph/ vxmax,vymax,
      *          devhandle
      character*80 texte,rep
      dimension vechoxy(2)

```

```

      status=vstcol(devhandle,vcou)
      vcount=ncount
      count=ncount
      vver=2
      vhor=0

```



```

xmax=vxmax
ymax=vymax
col=xmax+(xcol-80.)*(-xmax)/(-80.)
rlig=ymax+(xlig)*(-ymax)/(24.)
vcol=col
vlig=rlig
status=vstaln(devhandle,vhor,vver,vhor,vver)
status=vgtxts(devhandle,vcol,vlig,vcount,texte)

```

```

x=xcol+count
y=xlig+0.61
col=xmax+(x-80.)*(-xmax)/(-80.)
rlig=ymax+(y)*(-ymax)/(24.)
vechoxy(1)=col
vechoxy(2)=rlig
status=vrqstr(devhandle,80,1,vechoxy,rep)
return
END

```

```

-----
subroutine rinput(vcoul,ncount,x,y,texte,rep)
-----

```

```

implicit integer*2 (v)
common /graph/ vxmax,vymax,
* devhandle
character*80 texte,rep
dimension vechoxy(2)

status=vstcol(devhandle,vcoul)
vcount=ncount
count=ncount
vver=0
vhor=0
status=vstaln(devhandle,vhor,vver,vhor,vver)
status=vgtxts(devhandle,vtrx(x),vtry(y),vcount,texte)

```

```

REP='
vechoxy(1)=vtrx(x)+(vcount)*410
vechoxy(2)=vtry(y)
status=vrqstr(devhandle,80,1,vechoxy,rep)
return
END

```

```

-----
subroutine PRNT(vcoul,ncount,xlig,xcol,texte)
-----

```

```

implicit integer*2 (v)
common /graph/ vxmax,vymax,
* devhandle
character*80 texte
dimension vechoxy(2)

status=vstcol(devhandle,vcoul)
vcount=ncount
count=ncount
vver=2
vhor=0
xmax=vxmax
ymax=vymax
col=xmax+(xcol-80.)*(-xmax)/(-80.)
rlig=ymax+(xlig)*(-ymax)/(24.)
vcol=col
vlig=rlig
status=vstaln(devhandle,vhor,vver,vhor,vver)

```

```

status=vgtxts(devhandle,vcol,vlig,vcount,texte)

```

```

RETURN
END

```

```

-----
FUNCTION TTN(TEXTE)
-----

```

```

CHARACTER*80 TEXTE
CHARACTER*25 TEXTE1

```

```

TEXTE1=TEXTE(1:25)

```

```

CALL RPEN(ICODE,ITYPE,LONG,TEXTE1)

```

```

C
C *****
C *** ajoute de J.Nicolas 5.6.89 ***
C *** cas particulier des entiers ***
C

```

```

IF (ICODE.EQ.4) TEXTE1(ITYPE:ITYPE)='.'

```

```

C ***
C *****
C

```

```

CALL TRARE(ITYPE,TEXTE1,X)

```

```

TTN=X
RETURN
END

```

```

-----
subroutine efface(devhandle,x1,y1,x2,y2)
-----

```

```

C cette sous-routine remplit de noir le rectangle de coordonnes
C (x1,y1) coin inferieur gauche
C (x2,y2) coin superieur droit

```

```

implicit integer*2 (v)
dimension vxy(4)

```

```

vxy(1)=vtrx(x1)
vxy(2)=vtry(y1)
vxy(3)=vtrx(x2)
vxy(4)=vtry(y2)
vcoul=0
vstyle=1

```

```

status=vsfcol(devhandle,vcoul)
status=vsfint(devhandle,vstyle)
status=vbar(devhandle,vxy)

```

```

return
end

```

```

-----
SUBROUTINE RPEN(ICODE,ITYPE,LONG,CALPHA)
-----

```

```

CHARACTER*25 CALPHA
CHARACTER*1 SYMB(15)
SYMB(1)='0'
SYMB(2)='1'
SYMB(3)='2'
SYMB(4)='3'
SYMB(5)='4'
SYMB(6)='5'
SYMB(7)='6'

```

```

SYMB(8)='7'
SYMB(9)='8'
SYMB(10)='9'
SYMB(11)='.'
SYMB(12)='- '
SYMB(13)='+ '
SYMB(14)='E'
SYMB(15)='D'
ICODE=1
ITYPE=0
DO 10 I=1,25,1
IF(CALPHA(I:I).EQ.' ')GOTO 15
DO 5 J=1,15,1
IF(CALPHA(I:I).EQ.SYMB(J))GO TO 7
5 CONTINUE
ICODE=0
7 IF((J.EQ.11).AND.(ITYPE.GE.1))ICODE=0
IF((J.EQ.11).AND.(ICODE.NE.0))ITYPE=I
IF((J.EQ.14).AND.(ICODE.NE.0))ICODE=2
IF((J.EQ.15).AND.(ICODE.NE.0))ICODE=3
IF((J.EQ.12).AND.(ICODE.EQ.1).AND.(I.GT.1))ICODE=0
10 CONTINUE
15 LONG=I-1
C
C *****
C *** Ajoute de J.Nicolas le 5/6/89 ***
C *** Cas particulier des entiers ***
C
IF((ICODE.EQ.1).AND.(ITYPE.EQ.0)) THEN
  ICODE=4
  ITYPE=I
ENDIF
C ***
C *****
C
RETURN
END
C-----
SUBROUTINE TRARE (ITYPE, CALPHA, VAR)
C-----
CHARACTER*25 CALPHA
CHARACTER*1 SYMB(15)
SYMB(1)='1'
SYMB(2)='2'
SYMB(3)='3'
SYMB(4)='4'
SYMB(5)='5'
SYMB(6)='6'
SYMB(7)='7'
SYMB(8)='8'
SYMB(9)='9'
SYMB(10)='0'
SYMB(12)='.'
SYMB(11)='- '
SYMB(13)='+ '
SYMB(14)='E'
SYMB(15)='D'
ISI=0
IF(CALPHA(ITYPE:ITYPE).NE.SYMB(12)) THEN
WRITE(*,1001)CALPHA
1001 FORMAT('ERREUR AU NIVEAU DE LA POS. DU . : ',A25)
RETURN
ENDIF

```

```

LONG=ITYPE-1
VAR=0
IF(LONG.LT.1)GOTO 30
IC=1
DO 20 I=LONG,1,-1
DO 10 J=1,15,1
IF(CALPHA(I:I).EQ.SYMB(J))GO TO 15
10 CONTINUE
15 IF(J.EQ.10)J=0
IF(J.GT.11)THEN
WRITE(*,1000)CALPHA
1000 FORMAT('ERREUR LA CHAINE N''EST PAS CORRECTE ',A25)
RETURN
ENDIF
IF(J.NE.11)THEN
VAR=VAR+IC*J
IC=IC*10
ELSE
IF(I.EQ.1)THEN
VAR=-VAR
ISI=1
ELSE
WRITE(*,1000)CALPHA
ENDIF
ENDIF
20 CONTINUE
30 LONG=ITYPE+1
C=0.1
DO 50 I=LONG,25,1
IF(CALPHA(I:I).EQ.' ')GO TO 100
IF(CALPHA(I:I).EQ.'E')GO TO 70
DO 40 J=1,15,1
IF(CALPHA(I:I).EQ.SYMB(J))GO TO 45
40 CONTINUE
45 IF(J.EQ.10)J=0
IF(J.GE.11)THEN
WRITE(*,1000)CALPHA
RETURN
ENDIF
IF(ISI.EQ.1)THEN
VAR=VAR-J*C
ELSE
VAR=VAR+J*C
ENDIF
C=C/10
50 CONTINUE
70 I=I+1
DO 75 J=1,15,1
IF(CALPHA(I:I).EQ.SYMB(J))GO TO 80
75 CONTINUE
WRITE(*,1000)CALPHA
RETURN
80 CX=10
IF(J.EQ.11)CX=0.1
IF(J.EQ.13)CX=10
IF(J.LT.11)THEN
CX=10
GOTO 90
ENDIF
I=I+1
DO 85 J=1,15,1
IF(CALPHA(I:I).EQ.SYMB(J))GO TO 90
85 CONTINUE

```

```

90 IF (J.EQ.10) J=0
    III=J*10
    I=I+1
    DO 92 J=1,15,1
    IF (CALPHA(I:I).EQ.SYMB(J)) GO TO 94
92 CONTINUE
    IF (CALPHA(I:I).EQ.' ') III=III/10
94 IF (J.EQ.10) J=0
    III=III+J
    IF (III.EQ.0) GOTO 100
    IF (III.EQ.1) GOTO 99
    DO 96 J1=2,III,1
    CX=CX*CX
96 CONTINUE
99 VAR=VAR*CX
100 RETURN
    END

```

 SUBROUTINE CONVNT(X,TEXTE)

```

C CONVNT convertit un nombre reel x en caracteres ASCII dans TEXTE
C
C   character*80 texte
C   integer posit,digit
C   do 2 posit=1,80
C   texte(posit:posit)=' '
C   continue
C   if (x.eq.0.) then
C     texte(1:1)='0'
C     texte(2:2)='.'
C     do 3 posit=3,15
C     texte(posit:posit)='0'
C   continue
C   return
C   endif
C   posit=1
C   if (x.lt.0.) then
C     texte(posit:posit)='- '
C     posit=posit+1
C   endif
C   valeur=abs(x)
C   if (valeur.lt.1.) then
C     n=-1
C   else
C     n=int(alog10(valeur))
C   endif
C   iold=1
C   do 1 i=n,n-15,-1
C   if ((i.lt.0).and.(iold.ge.0)) then
C     texte(posit:posit)='.'
C     posit=posit+1
C   endif
C   digit=int(valeur/10.**i)
C   texte(posit:posit)=char(digit+48)
C   valeur=valeur-digit*10.**i
C   posit=posit+1
C   iold=i
C   continue
C   return
C   end

```

CYCLE.DAT

29.44	-15.39	-15.39	36.08	43.55	29.44
1.098	1.111	1.567	1.567	1.587	1.549

PACSORSI.DAT

AU CONDENSEUR TEMPERATURE D ENTREE DE L EAU : 10.0Dg Cel
 TEMPERATURE DE SORTIE DE L EAU: 22.0Dg Cel
 DEBIT DE L EAU : 1.0Kg/Sec
 TEMPERATURE DE CONDENSATION : 29.4Dg Cel

A L EVAPORATEUR TEMPERATURE D ENTREE DE L EAU: .0Dg Cel
 TEMPERATURE DE SORTIE DE L EAU: -9.5Dg Cel
 DEBIT DE L EAU : 1.0Kg/Sec
 TEMPERATURE D EVAPORATION: -15.4Dg Cel

COEFFICIENT DE TRANSMISSION THERMIQUE AU CONDENSEUR: 4.00 kJ/Dg/Sec
 COEFFICIENT DE TRANSMISSION THERMIQUE A L EVAPORATEUR: 4.00 kJ/Dg/Sec

	Dg Cel	kJ/kg	kJ/kg/Dg
POINT A	T= 29.4	I=228.28	S= 1.0975
POINT B	T=-15.4	I=228.28	S= 1.1114
POINT C	T=-15.4	I=345.44	S= 1.5671
POINT E	T= 43.6	I=376.84	S= 1.5871
POINT D	T= 36.1	I=370.56	S= 1.5671
POINT G	T= 29.4	I=364.99	S= 1.5488

PRESSION AU CONDENSEUR= .734 MPA
 PRESSION A L EVAPORATEUR= .180 MPA

CHALEUR FOURNIE PAR LE CONDENSEUR : 50.06 kJ/Sec
 CHALEUR ABSORBEE A L EVAPORATEUR : 39.62 kJ/Sec
 PUISSANCE CONSOMMEE PAR LE COMPRESSEUR : 10.62 kJ/Sec
 RENDEMENT THEORIQUE : 4.713
 DEBIT DE FREON : .3382 kg/Sec
 VITESSE DU COMPRESSEUR : 100.0 Tr/Sec
 VOLUME DE BALAYAGE DU COMPRESSEUR: .2000E-03m**3
 RENDEMENT THERMIQUE DU COMPRESSEUR: .800

PACDONSI.DAT

RKCO	RKEV	CPEAU	CFREO	REND COMP
4.000	4.00	4.18	0.84	0.8

TABLE.DAT

T Dg C	P MPA	VS M**3/Kg	IP kJ/Kg	IS kJ/Kg	SP kJ/Kg/K	SS kJ/Kg/K
****	.00117	2.6180	112.76	306.45	.60417	1.72374
-95.	.00188	2.1962	116.67	308.74	.62934	1.70392
-90.	.00281	1.8335	121.28	310.80	.65118	1.68855
-85.	.00421	1.5235	125.58	313.08	.67390	1.67332
-80.	.00617	1.2599	129.61	315.49	.69688	1.65876
-75.	.00880	1.0371	133.65	317.87	.71904	1.64583
-70.	.01226	.8499	137.85	320.17	.74003	1.63477
-65.	.01677	.6938	142.21	322.38	.76014	1.62521
-60.	.02257	.5644	146.65	324.60	.77988	1.61660
-55.	.02992	.4579	151.07	326.88	.79959	1.60854
-50.	.03910	.3709	155.40	329.24	.81933	1.60089
-45.	.05041	.3004	159.66	331.69	.83896	1.59378
-40.	.06417	.2435	163.86	334.16	.85824	1.58738
-35.	.08072	.1980	168.10	336.61	.87698	1.58185
-30.	.10044	.1618	172.43	338.98	.89514	1.57720
-25.	.12373	.1330	176.88	341.26	.91281	1.57330
-20.	.15100	.1102	181.47	343.46	.93020	1.56992
-15.	.18269	.0922	186.15	345.61	.94753	1.56683
-10.	.21926	.0778	190.88	347.75	.96494	1.56384
-5.	.26116	.0662	195.60	349.94	.98247	1.56091
0.	.30886	.0568	200.27	352.19	1.00003	1.55809
5.	.36283	.0489	204.90	354.50	1.01745	1.55552
10.	.42358	.0423	209.50	356.83	1.03456	1.55336
15.	.49160	.0367	214.14	359.13	1.05123	1.55169
20.	.56741	.0318	218.88	361.31	1.06747	1.55048
25.	.65154	.0276	223.77	363.34	1.08341	1.54957
30.	.74455	.0240	228.85	365.19	1.09928	1.54873
35.	.84699	.0210	234.11	366.87	1.11532	1.54770
40.	.95943	.0184	239.47	368.45	1.13168	1.54630
45.	1.08247	.0164	244.85	370.00	1.14837	1.54453
50.	1.21669	.0147	250.18	371.59	1.16518	1.54254
55.	1.36273	.0131	255.42	373.25	1.18181	1.54063
60.	1.52124	.0116	260.59	374.93	1.19794	1.53905
65.	1.69294	.0096	265.81	376.50	1.21347	1.53788
70.	1.87859	.0068	271.28	377.75	1.22866	1.53680
75.	2.07906	.0024	277.18	378.49	1.24418	1.53516
80.	2.29526	-.0043	283.61	378.61	1.26082	1.53211
85.	2.52822	-.0143	290.47	378.22	1.27908	1.52718
90.	2.77906	-.0288	297.41	377.65	1.29848	1.52080
95.	3.04906	-.0493	303.99	377.28	1.31737	1.51446
100.	3.33984	-.0775	310.35	376.83	1.33427	1.50922
105.	3.65371	-.1154	318.55	373.84	1.35294	1.50032
110.	3.99448	-.1652	335.42	360.65	1.39485	1.46436

LINK_PAC.BAT

link pac+simul+fichiers+menus+dessin+freon12, , ,fortran+graph.lib

PAC.MAP

Start	Stop	Length	Name	Class
00000H	00299H	0029AH	PAC_TEXT	CODE
0029AH	01659H	013C0H	SIMUL_TEXT	CODE
0165AH	01903H	002AAH	FICHIERS_TEXT	CODE
01904H	02D5FH	0145CH	MENUS_TEXT	CODE
02D60H	0525CH	024FDH	DESSIN_TEXT	CODE
0525DH	05EA1H	00C45H	FREON12_TEXT	CODE
05EA2H	05F26H	00085H	VGTX	CODE
05F28H	05FE4H	000BDH	VCIRC	CODE
05FE6H	06055H	00070H	VCLR	CODE
06056H	060C3H	0006EH	VCLSW	CODE
060C4H	06139H	00076H	VENC	CODE
0613AH	061F3H	000BAH	VOPNW	CODE
061F4H	06266H	00073H	VPLIN	CODE
06268H	062D8H	00071H	VSLCO	CODE
062DAH	0636DH	00094H	VSTAL	CODE
0636EH	063DEH	00071H	VSTCO	CODE
063E0H	0647AH	0009BH	VSTHG	CODE
0647BH	089C4H	0254AH	GLIB_TEXT	CODE
089C5H	08A10H	0004CH	GSSVDI_SEG	CODE
08A12H	08A87H	00076H	VBAR	CODE
08A88H	08B8FH	00108H	VRQST	CODE
08B90H	08C00H	00071H	VSFCO	CODE
08C02H	08C72H	00071H	VSFIN	CODE
08C74H	08CE4H	00071H	VSLTY	CODE
08CE6H	08D56H	00071H	VSWRM	CODE
08D57H	0B707H	029B1H	TEXT	CODE
0B708H	0EA99H	03392H	FR	CODE
0EAA0H	0F231H	00792H	EMULATOR_TEXT	CODE
0F232H	0F2C7H	00096H	3_CHSIZE_TEXT	CODE
0F2C8H	0F2C8H	00000H	C_ETEXT	ENDCODE
0F2D0H	0F31FH	00050H	CYCLE\$A	\$CYCLE
0F320H	0F327H	00008H	REGIME\$A	\$REGIME
0F330H	0F337H	00008H	COMPRES\$A	\$COMPRES
0F340H	0F34FH	00010H	CONSTH\$A	\$CONSTH
0F350H	0F7FFH	004B0H	SIMUL\$A	\$SIMUL

0F800H	0F80FH	00010H	INOUT2\$A	\$INOUT2
0F810H	0F82FH	00020H	INOUT\$A	\$INOUT
0F830H	0F84FH	00020H	ABSREL\$A	\$ABSREL
0F850H	0F857H	00008H	GRAPH\$A	\$GRAPH
0F860H	0F877H	00018H	PUISS\$A	\$PUISS
0F880H	0FE9FH	00620H	FICHIERS5_DATA	FAR_DATA
0FEA0H	0FEA6H	00007H	DESSIN5_DATA	FAR_DATA
0FEB0H	OFF04H	00055H	GLIB5_DATA	FAR_DATA
OFF10H	OFFFFH	000F0H	EMULATOR_DATA	FAR_DATA
10000H	1073AH	0073BH	ERRDIC5_DATA	FAR_DATA
1073CH	1073DH	00002H	FTNVDI	COMMON
1073EH	1073EH	00000H	HIMEM	HIMEM
10740H	1075FH	00020H	WIND\$A	\$WIND
10760H	1079BH	0003CH	NULL	BEGDATA
1079CH	11E65H	016CAH	_DATA	DATA
11E66H	1200DH	001A8H	DATA	DATA
1200EH	1201BH	0000EH	CDATA	DATA
1201CH	1201CH	00000H	XIB	DATA
1201CH	1201FH	00004H	XI	DATA
12020H	12020H	00000H	XIE	DATA
12020H	12020H	00000H	XCB	DATA
12020H	12020H	00000H	XC	DATA
12020H	12020H	00000H	XCE	DATA
12020H	12020H	00000H	XPB	DATA
12020H	12020H	00000H	XP	DATA
12020H	12020H	00000H	XPE	DATA
12020H	12421H	00402H	CONST	CONST
12422H	12422H	00000H	HEAP	MEMORY
12422H	12465H	00044H	COMADS	COMADS
12466H	12466H	00000H	COMMOQ	COMMON
12466H	124AFH	0004AH	_CALLTAB	CALLTAB
124B0H	124B7H	00008H	HDR	MSG
124B8H	126C4H	0020DH	MSG	MSG
126C5H	12728H	00064H	PAD	MSG
12729H	12729H	00001H	EPAD	MSG
1272AH	13D73H	0164AH	_BSS	BSS
13D74H	13D74H	00000H	XOB	BSS
13D74H	13DF3H	00080H	XO	BSS
13DF4H	13DF4H	00000H	XOE	BSS
13E00H	145FFH	00800H	STACK	STACK

Origin Group
1076:0 DGROUP

Program entry point at 08D5:0284

II. 6909

Energie
Pompe à

MINIMISATION DE LA CONSOMMATION D'ENERGIE
D'UNE POMPE A CHALEUR PAR L'EMPLOI D'UNE COMMANDE
A MICRO-PROCESSEUR

Mr. J. EHRHART

Ecole Supérieure d'Ingénieurs en Electrotechnique et Electronique

Abstract

A heat pump heating water and cooling air will be optimized by a micro-processor controlling a variable speed (and power) compressor and a motorised reducing valve.
This command will be performed on two levels :
First : the thermodynamical cycle of the freon will be controlled : complete evaporation in the chiller without overheating of the gas will be checked and a similar test made on the condenser.
The second command will be an adaptation of the heat pump to varying external conditions (inlet temperature of water or air, water flow) in order to maintain at a constant value the water outlet temperature. It will also be possible to perform a time varying evolution of the water outlet temperature in an industrial process.
This second level needs a modelisation of the system. In a first time, this modelisation has been performed through a mathematical simulation of the heat pump. This mathematical model includes a number of empirical formulae and will be accurated by an experimental study of the system.

Contract n° 305-78 REF

Fondation Universitaire
Luxembourgeoise
Centre de Documentation

1 - Objectif du contrat

Il s'agit de minimiser la consommation d'énergie d'une pompe à chaleur en commandant son fonctionnement par un microprocesseur. Cette minimisation revêtira deux aspects :

- Contrôle du cycle de la pompe à chaleur pour des conditions extérieures données.
- Evolution du point de fonctionnement de la machine pour compenser ou créer des variations des paramètres externes.

2 - Méthodes appliquées et mise en oeuvre

2.1 Description du système.

La source froide est de l'air dont la température d'entrée dans la pompe est T_{ea} , la température de sortie T_{sa} , le débit massique m_a .

La source chaude est de l'eau. La température d'entrée dans l'échangeur est T_{ee} , de sortie T_{se} et le m_e . A l'intérieur de la pompe à chaleur le fréon circule avec un débit massique m_f ; il s'évapore à la température T_e et se condense à la température T_c , les pressions étant respectivement P_e et P_c .

Les paramètres de commande de la machine sont N , vitesse de rotation du compresseur et D , diamètre équivalent à l'ouverture de la valve commandable qui sert de détendeur.

Une autre variable d'utilité secondaire sera la température T_s : température de surchauffe du fréon en fin de compression. Q_e et Q_c seront les puissances calorifiques échangées par la pompe à l'évaporateur (source froide) et au condenseur (source chaude), W la puissance mécanique fournie au compresseur.

2.2 Contrôle du cycle du fréon.

Le cycle de la pompe à chaleur comprend :

Une évaporation isobare, une compression isentropique, une conden-

sation isobare, une détente isenthalpique. La compression et la détente ne posent guère de problème. Elles sont modifiables par variation de N et/ou D .

Les transformations isobares qui sont essentiellement des changements d'état doivent faire l'objet d'un contrôle par la commande. Le rendement de la pompe à chaleur sera d'autant meilleur que ces transformations s'effectuent correctement.

Il importe donc de veiller à ce qu'il n'y ait pas d'échauffement isobare du gaz après l'évaporation, ni de refroidissement du liquide après condensation. Ce sera le rôle d'un premier niveau de commande :

- a) Mesure de la différence de température du fréon entre le coeur de l'évaporateur et sa sortie et mise en oeuvre d'une régulation la maintenant à une valeur minimale qui sera fonction des détecteurs. Cette valeur sera non nulle car il faut que l'évaporation soit complète.
- b) Même mesure entre le coeur du condenseur et sa sortie.

2.3 Commande de la pompe en vue de répondre à des contraintes extérieures.

En cas d'évolution des paramètres externes tels que température d'entrée de l'eau ou de l'air, débit de l'eau, les paramètres de fonctionnement du système devront évoluer de façon à optimiser le cycle du fréon et de l'adapter à ces nouvelles conditions extérieures. Il pourra être nécessaire également de faire évoluer dans le temps la température de sortie d'eau afin de répondre aux besoins d'un processus industriel par exemple. Le deuxième niveau de contrôle devra donc fixer les valeurs des variables commandables N (vitesse de rotation et donc puissance du compresseur) et D (ouverture du détendeur) pour que le cycle parcouru par le fréon soit optimal.

2.4 Modélisation de la pompe à chaleur.

Nous avons entrepris en un premier temps, une modélisation mathématique de la pompe à chaleur.

Il s'agit d'écrire les différentes relations entre variables externes, internes et de commande du système et de résoudre le problème suivant.

- Les conditions de fonctionnement externes et la structure physique de la machine étant données, quelles doivent être les valeurs des variables commandables N et D pour que la pompe à chaleur réponde à ces conditions externes en consommant le moins d'énergie possible (cycle optimisé) ?

Cette modélisation dont on trouvera les éléments essentiels en annexe, est une base de travail, car elle ne pourra être qu'une première approximation étant donné le caractère approché des formules utilisées. C'est le comportement réel de la pompe à chaleur et un ensemble de mesures effectuées sur elle qui permettront de préciser le modèle et de l'affiner.

Le rôle de la commande sera, alors, lorsque les conditions extérieures évoluant de mesurer les nouvelles valeurs des paramètres externes et d'en déduire les nouvelles valeurs de N et D qui, par exemple, permettront de maintenir constante la température de sortie d'eau au condenseur, ou bien, pour des conditions extérieures données, de faire évoluer dans le temps N et D afin de faire décrire à la température de sortie d'eau une courbe fixée à l'avance et introduite par un opérateur.

3 - Etat de la recherche et résultats. Prochaines étapes.

Le programme FORTRAN est écrit et sa mise au point très avancée. Nous aurons très prochainement les résultats de cette modélisation.

Il s'agira, à partir de là, par une étude expérimentale, de vérifier le modèle et de l'affiner. Parallèlement il faudra mettre en oeuvre la com-

mande de la pompe au premier niveau (optimisation du cycle) avant de réaliser, à partir du modèle, la commande globale.

4 - Conclusion.

Avec une telle commande, on peut espérer obtenir deux résultats essentiels.

- Un fonctionnement optimisé de la pompe à chaleur pour des conditions extérieures très diverses.

- La possibilité de réaliser des flux de chaleur programmés dans le temps, ce qui peut être très intéressant dans certains processus industriels.

ANNEXE :

Relations entre variables dans la pompe à chaleur. Principe de la modélisation.

1 - Relations entre variables

1.1 Le condenseur.

Il s'agit d'un tube sur lequel vient se condenser le fréon. L'eau circule à l'intérieur du tube.

1.1.1 Chaleur reçue par l'eau.

$$Q_c = m_e C_e (T_{se} - T_{ee}) \quad \text{éq. (1)}$$

Les grandeurs Q_c , T_{se} , T_{ee} , m_e ont été définies plus haut
 C_e : chaleur massique de l'eau.

1.1.2 Echange de chaleur eau-paroi.

$$Q_c = \alpha_e S_{int} (T_{pint} - T_{me}) \quad \text{éq. (2)}$$

α_e : coefficient de transmission thermique par convection de l'eau.

S_{int} : surface interne du tube.

T_{pint} : température de la paroi interne du tube.

$$T_{me} = \frac{T_{se} + T_{ee}}{2}$$

α_e est donné à partir de la formule empirique :

$$\alpha_e = 0,023 \frac{\lambda_e^{0,5} G^{0,8} C_e^{0,5}}{D_{cint}^{0,2} n^{0,3}} \quad \text{éq. (3)}$$

λ_e : conductivité thermique de l'eau.

$$G = \frac{4 m_e}{\pi (D_{cint})^2} \quad n : \text{viscosité dynamique de l'eau.}$$

D_{cint} : diamètre intérieur du tube.

En combinant les équations (2) et (3) on obtient :

$$Q_c = 0,023 \frac{\lambda_e^{0,5} G^{0,8} C_e^{0,5} S_{int}}{D_{cint}^{0,2} n^{0,3}} (T_{pint} - T_{me}) \quad \text{éq. (4)}$$

1.1.3 Chaleur transmise à travers la paroi.

$$Q_c = \frac{\lambda_m}{e} S_{moy} (T_{pext} - T_{pint}) \quad \text{éq. (5)}$$

T_{pext} : température de la paroi externe du tube.

λ_m : conductivité thermique du métal.

e : épaisseur de la paroi.

S_{moy} : surface moyenne logarithmique du tube.

$$S_{moy} = \frac{S_{ext} - S_{int}}{\ln \frac{S_{ext}}{S_{int}}} \quad S_{ext} : \text{surface extérieure du tube.}$$

1.1.4 Chaleur émise par le fréon.

$$Q_c = \alpha_f S_{ext} (T_c - T_{pext}) \quad \text{éq. (6)}$$

α_f : coefficient de transmission thermique du fréon au métal.

$$\alpha_f = 0,725 \left[\frac{g^2 L_{vc} \lambda_f^3}{\mu D_{cext} (T_c - T_{pext})} \right]^{1/4} \quad \text{éq. (7)}$$

g : champ de pesanteur.

L_{vc} : chaleur de vaporisation du fréon.

λ_f : conductivité thermique du fréon liquide.

μ : viscosité dynamique du fréon liquide.

ρ : masse volumique du fréon liquide

D_{cext} : diamètre externe du tube.

1.1.5 Relation entre Q_c et m_f .

$$Q_c = m_f L_{vc} \quad \text{éq. (8)}$$

1.2 L'évaporateur.

Il est constitué de tubes ailetés parallèles sur lesquels l'air est soufflé.

L'air est supposé sec en un premier temps.

1.2.1 Chaleur cédée par l'air.

$$Q_e = m_a C_{pa} (T_{ea} - T_{sa}) \quad \text{éq. (9)}$$

C_{pa} : chaleur massique de l'air à pression constante.

Les autres grandeurs ont été définies dans le texte.

1.2.2 Chaleur transmise par convection au tube.

$$Q_e = \varphi_a \alpha_a S_{ext} (T_{ma} - T_{pext}) \quad \text{éq. (10)}$$

φ_a : rendement d'ailette.

α_a : coefficient de transmission thermique air paroi par convection.

S_{ext} : Surface extérieure du tube avec les ailettes.

$$T_{ma} = \frac{T_{ea} + T_{sa}}{2}$$

T_{pext} : température externe de la paroi.

Le coefficient de transmission thermique α_a est donné par :

$$\alpha_a = 0,174 \left(\frac{v}{\nu_a} \right)^{0,62} D_{ext}^{-0,38} \quad \text{éq. (11)}$$

D_{ext} : diamètre de la soufflerie.

v : vitesse d'écoulement de l'air.

ν_a : viscosité cinématique de l'air.

1.2.3 Chaleur transmise par conduction.

$$Q_e = \frac{\lambda_m}{e} S_{moy} (T_{pext} - T_{pint}) \quad \text{éq. (12)}$$

les symboles sont :

λ_m : conductivité thermique du métal.

e : épaisseur du tube.

S_{moy} : surface moyenne logarithmique du tube.

T_{pint} : température de la paroi interne.

1.2.4 Chaleur reçue par le fréon de la paroi.

$$Q_e = \alpha_f S_{int} (T_{pint} - T_e) \quad \text{éq. (13)}$$

α_f : coefficient de transmission thermique entre le fréon et la paroi.

$$\alpha_f = 0,0145 \left(\frac{Q_u m_f}{D_{int}^{1,8}} \right)^{0,4} \quad \text{éq. (14)}$$

Q_u : puissance frigorifique unitaire.

$Q_u = \frac{Q_e}{S_{int}}$ S_{int} : surface de la paroi interne.

D_{int} : diamètre interne du tube.

1.2.5 Chaleur reçue par le fréon.

$$Q_e = m_f (1 - x_e) L_{ve} \quad \text{éq. (15)}$$

x_e : teneur en vapeur du fréon à l'entrée de l'évaporateur.

L_{ve} : chaleur de vaporisation du fréon.

x_e est déterminé par des tables à partir de la connaissance des pressions d'évaporation et de condensation (P_e et P_c) du fréon.

1.3 Détendeur.

On suppose un diaphragme de diamètre d dans un tube de diamètre D .

$$D_v = 0,6 \times \frac{\pi}{4} d^2 \left[\frac{2(P_c - P_e)}{\rho \left(1 - \frac{d^2}{D^2}\right)} \right]^{1/2} \quad \text{éq. (16)}$$

ρ : masse volumique du liquide.

P_e et P_c : pressions de part et d'autre du diaphragme.

D_v : débit volumique.

1.4 Compresseur.

C'est un compresseur à piston. L'étude donne :

$$N = \frac{m_f V_E}{2(1 - 0,05 \frac{P_c}{P_e}) V_p} \quad \text{éq. (17)}$$

V_E : volume massique du fréon à l'entrée du compresseur.

V_P : volume balayé par un piston.

2 - Principe de la modélisation

Nous utilisons les relations vues précédemment pour modéliser la pompe à chaleur.

Les étapes de la modélisation sont les suivantes :

2.1 Données initiales.

Connaissant les caractéristiques en entrée et en sortie de l'eau côté condenseur à savoir T_{ee} , T_{se} et m_e , nous obtenons par l'équation (1) Q_c la chaleur échangée au condenseur.

L'équation (4) permet de calculer T_{pint} .

L'équation (5) donne T_{pext} .

L'équation (6) et (7) associées à la relation entre la chaleur de vaporisation du fréon L_{vc} et la température T_c , permettent de déterminer T_c et donc L_{vc} .

La résolution de l'équation (8) donne le débit de fréon m_f . La connaissance de T_c donne aussi P_c (pression dans le condenseur).

2.2 L'évaporateur est abordé ensuite.

On fixe d'abord m_a et T_{ea} qui sont des données. On choisit une valeur de Q_e (donc de T_{sa}) qui sera modifiée ensuite par des itérations successives jusqu'à ce qu'elle soit correcte.

La connaissance de Q_e donne par les équations (10) et (11) la valeur de la température externe de la paroi de l'évaporateur. Par (12) on obtient T_{pint} . Par (13) et (14) on trouve T_e et donc T_e L_{vu} .

On peut alors calculer x_e et donc par (15) sa nouvelle valeur de Q_e qui devra être comparée à celle initialement choisie. Si elle est différente une itération aura lieu.

On arrivera au bout de quelques calculs de ce type à la bonne valeur de Q_e .

Toutes les variables sont alors connues pour calculer N dans (17) et d dans (16). Mais on fera encore une itération pour tenir compte de la surchauffe dans la compression.