



# INSTITUTO UNIVERSITARIO AERONÁUTICO

## Optimización de Motores Stirling

**Informe Técnico:** DMA-013/12

**Revisión:** /

**Proyecto:** Fonarsec INTIHUASI

**Fecha:** 09/10/2012

**Autor:**

Ing. Gustavo Scarpin  
*Director Dpto. Mec. Aer*



## **PROYECTO FONARSEC INTIHUASI**

### **Optimización de Motores Stirling**

Por:

*Ing. Gustavo Scarpin*

#### **RESUMEN**

Dentro del proyecto de desarrollo llevado a cabo por el Consorcio Público Privado Intihuasi, en el que el IUA es parte, y patrocinado por el Fondo de Innovación Tecnológica Sectorial de Energía Solar 2010 que administra el Fondo Argentino Sectorial (FONARSEC), se desarrolló un procedimiento que permite optimizar los diferentes parámetros de diseño de un motor Stirling el cual usa como método de predicción de las performances del motor, el modelo Simple descrito en Ref. [1].

Con esta nueva aproximación es posible usar el modelo Simple, no sólo para predecir el comportamiento del motor Stirling, sino también como herramienta de diseño, optimizando cualquiera de los parámetros ingresados como datos.

Esta nueva herramienta será usada para optimizar los motores Stirling que serán diseñados dentro del Proyecto FONARSEC Intihuasi.

En el Anexo A se presenta el programa de cómputo OptimizaSt.for el cual permite automatizar los cálculos. A modo de ejemplo se optimiza los parámetros geométricos de un motor tipo Alfa.

**Córdoba, 09 de octubre de 2012**



**ÍNDICE**

1. INTRODUCCIÓN .....	3
2. DESARROLLO .....	3
3. RESULTADOS .....	4
4. CONCLUSIONES .....	4
5. REFERENCIAS.....	4
ANEXO A: MANUAL DE USUARIO – PROGRAMA OPTIMIZAST .....	5



## 1. INTRODUCCIÓN

El modelo de análisis Simple fue desarrollado con el propósito de lograr una buena aproximación analítica a las performances de un motor Stirling. Al implementar el método en un programa de cómputos, se observa que el mismo posee un tiempo de corrida muy corto.

Esto, y la forma modular con que fue diseñado el programa, permiten que el mismo sea utilizado como parte de un programa de optimización.

El método de optimización implementado consiste en definir una función costo y una serie de variables de entrada las cuales se modifican hasta obtener un mínimo de la función costo.

En este caso, las variables de entrada son los parámetros geométricos del motor Stirling que se quiere optimizar y la función costo es el rendimiento del motor calculado con el programa descrito en Ref. [1]. Como restricción, se fija la potencia de salida del motor.

## 2. DESARROLLO

El método de optimización consiste en definir una función costo la cual es minimizada utilizando subrutinas estándar obtenidas de librerías Fortran. En este caso se utilizará la Subrutina SMPLX.FOR.

La función costo es definida como:

$$\text{COSTO} = (\text{actWpower} - 140.D0)**2.D0 + 1.D2*(1.D0-\text{acteff})**2.D0$$

Donde

**actWpower**: Potencia de salida del motor Stirling calculado con el modelo Simple de ref. [1].

**acteff**: Eficiencia real del motor calculada con el modelo Simple de ref. [1].

**(actWpower - 140.D0)\*\*2.D0** es la restricción que fija la potencia de salida deseada (132 Watt)

**1.D2\*(1.D0-acteff)\*\*2.D0** es la función a minimizar. La misma posee un peso de 100.

Los parámetros que se pueden optimizar son variados, básicamente cualquier parámetro de entrada.

Por ejemplo, en el caso de los intercambiadores de calor, permite optimizar su geometría ya que es deseable que los mismos posean alta transferencia de calor, baja pérdida de presión y pequeño volumen. Todo ello analizado en un ambiente dinámico e interactuando con otros intercambiadores de calor y componentes, como es el caso de los motores Stirling.

En la metodología aquí presentada se optimizan todos los componentes actuando en conjunto y no sólo la optimización individual de cada uno de ellos.



### 3. RESULTADOS

A modo de ejemplo, en el Anexo A se presentan los datos de entrada y los datos de salida luego de la optimización. En la siguiente tabla se presenta un resumen de los mismos.

<b>Performance</b>	<b>Sin Optimización</b>	<b>Optimizado</b>
<b>Entalpía neta perdida [W]</b>	134.9	56.3
<b>Fuga de calor en la pared [W]</b>	1.18	2.18
<b>Trabajo perdido por pérdida de Presión [W]</b>	3.29	11.3
<b>Calor real transferido al calentador [W]</b>	182.5	320.9
<b>Potencia real [W]</b>	<b>27.0</b>	<b>139.9</b>
<b>Eficiencia térmica real del motor [%]</b>	<b>14.8</b>	<b>43.6</b>

En este ejemplo, se minimiza la función costo presentado en §2. Se observa en la tabla que luego de modificar la geometría, el nuevo motor tiene la potencia deseada de 140 W con una alta eficiencia de 43.6%.

### 4. CONCLUSIONES

En el marco del proyecto Stirling-Solar se ha desarrollado un programa de cómputos que permite optimizar los parámetros geométricos de cada una de las partes de un motor con ciclo Stirling.

El programa optimiza la geometría para lograr maximizar la eficiencia, la cual es obtenida mediante el modelo Simple presentado en Ref. [1].

El programa que aquí se presenta es una importante herramienta que permitirá dimensionar los motores Stirling que se desarrollarán en el Proyecto FONARSEC Intihuasi.

### 5. REFERENCIAS

- [1] Gustavo Scarpin, "Modelo Simple de Motores Stirling", Inf. DMA-012/12, IUA-FI-DMA, 27/08/2012.



---

## ANEXO A: MANUAL DE USUARIO – PROGRAMA OPTIMIZAST

### B-1 INTRODUCCION

El programa OptimizaSt es un programa en Fortran que permite optimizar las performances de un motor Stirling utilizando el modelo Simple para calcular las performances. El modelo matemático y las rutinas de cálculo programadas en Fortran del modelo Simple fueron documentados en Ref. [1]. En este Anexo se presenta el listado fuente del programa junto con un archivo de entrada de ejemplo y su correspondiente archivo de salida.

### B-2 ARCHIVO DE ENTRADA

El archivo de entrada OptimizaSt.dat consta de diez Namelist, con la siguiente información:

#### 1. **&Motor**, datos generales de los pistones

- 1.1. **vswc**, Volumen de barrido de compresión (zona fría) [m<sup>3</sup>]
- 1.2. **vswe**, Volumen de barrido de expansión (zona caliente) [m<sup>3</sup>]
- 1.3. **vclic0**, Volumen libre de compresión (volumen libre de la zona fría) [m<sup>3</sup>]
- 1.4. **vcle0**, Volumen libre de expansión (volumen libre de la zona caliente) [m<sup>3</sup>]
- 1.5. **phase**, Desfasaje de volumen [°]

Nota: los volúmenes libres no incluyen los conductos de unión.

#### 2. **&Inter**, Define los tipos de intercambiadores de calor y matriz del regenerador.

- 2.1. **t\_enfriador**, Tipo de enfriador: t)ubos a)nillo r)anurado m)alla de alambres
- 2.2. **t\_calentador**, Tipo de calentador: t)ubos a)nillo r)anurado m)alla de alambres
- 2.3. **t\_regenerador**, Tipo de regenerador: t)ubular a)nular
- 2.4. **t\_matriz**, Tipo de matriz: m)alla de alambres l)ámina a)lambres apilados s)in matriz

#### 3. **&Geom\_Enf**, Datos geométricos del enfriador. Los datos de entrada dependen del tipo seleccionado en §2.1. A modo de ejemplo se presenta el tipo m).

- 3.1. **din**, Diámetro interno del tubo [m]
- 3.2. **len**, Longitud del intercambiador de calor [m]
- 3.3. **dwire**, Diámetro de los alambres de la malla de alambre [m]
- 3.4. **porosity**, Porosidad de la tela de alambres (ver § A-5 de Ref. [1])



- 
4. **&Geom\_Cal**, Datos geométricos del calentador. Los datos de entrada dependen del tipo seleccionado en §2.2 de Ref. [1]. A modo de ejemplo se presenta el tipo r).
    - 4.1. **w**, Ancho de la ranura [m]
    - 4.2. **h**, Alto de la ranura [m]
    - 4.3. **len**, Longitud del intercambiador de calor [m]
    - 4.4. **num**, Número de ranuras del paquete
  
  5. **&Geom\_Malla**, Datos geométricos de la malla del regenerador. Los datos de entrada dependen del tipo seleccionado en §2.4. A modo de ejemplo se presenta el tipo m).
    - 5.1. **dal**, Diámetro de los alambres de la malla de alambre [m]
    - 5.2. **porosity**, Porosidad de la tela de alambres (ver § A-5 de Ref. [1])
    - 5.3. **domat**, Diámetro interno del cárter del tubo [m]
    - 5.4. **espesor**, de las paredes del regenerador =  $(d_{out}-d_{omat})/2$  [m]
    - 5.5. **lr**, Longitud del regenerador [m]
  
  6. **&Geom\_Conk**, Datos geométricos de la conexión enfriador-regenerador.
    - 6.1. **dck**, Diámetro hidráulico del tubo del enfriador [m]
    - 6.2. **lck**, Longitud efectiva del tubo del enfriador [m]
  
  7. **&Geom\_Conh**, Datos geométricos de la conexión calentador-regenerador.
    - 7.1. **dch**, Diámetro hidráulico del tubo del calentador [m]
    - 7.2. **lch**, Longitud efectiva del tubo del calentador [m]
  
  8. **&Tipo\_Gas**, Tipo de gas de trabajo.
    - 8.1. **t\_gas**, Tipo de gas de trabajo: ai)re, he)lio, hi)drógeno
  
  9. **&Operacion**, Datos geométricos de la conexión calentador-regenerador.
    - 9.1. **pmean**, Presión promedio [Pa]
    - 9.2. **tk**, Temperatura externa en el enfriador [K]
    - 9.3. **th**, Temperatura externa en el calentador [K]
    - 9.4. **freq**, Velocidad de rotación de la manivela [Hz]



Archivo de entrada ejemplo (OptimizaSt.dat):

```
&Motor
  vswc = 98.17D-6
  vswe = 98.17D-6
  vclc0 = 20.11008D-6
  vcle0 = 41.7946D-6
  phase = 90.0D+0
/
&Inter
  t_enfriador = 'm'
  t_calentador = 'm'
  t_regenerador = 't'
  t_matriz = 'm'
/
&Geom_Enf
  din = 80.0D-3
  len = 5.0D-3
  dwire = 0.18D-3
  porosity = 0.63226D0
/
&Geom_Conk
  dck = 10.0D-3
  lck = 125.132D-3
/
&Geom_Malla
  porosity = 0.63226D0
  dal = 0.18D-3
  domat = 80.D-3
  espesor = .5D-3
  lr = 20.D-3
/
&Geom_Conh
  dch = 20.D-3
  lch = 57.0D-3
/
&Geom_Cal
  din = 90.D-2
  len = 5.D-3
  dwire = 0.18D-3
  porosity = 0.63226D0
/
&Tipo_Gas
  t_gas = 'ai'
/
&Operacion
  pmean = 100000.D0
  tk = 300.D0
  th = 923.D0
  freq = 50.D0
/
```





### **B-3 DATOS POR PANTALLA**

Cuando se corre el programa aparece por pantalla la misma información que se describe en el Anexo B de Ref. [1].

Por pantalla también se ingresa la opción si se desea optimizar o no y si se quiere los resultados calculado con el modelo Adiabático o Simple.

### **B-4 ARCHIVOS DE SALIDA**

Se generan 3 archivos de salida con los datos presentados en columnas para ser leídos fácilmente desde Excel<sup>®</sup>. Estos archivos son idénticos a los descritos en el Anexo B de Ref. [1].

Por otro lado se genera un archivo de salida denominado OptimizaSt.out, el cual contiene un namelist &SALIDA y en el que se especifican todos los datos de entrada pero ahora con los nuevos valores obtenidos de la optimización (o los datos originales si no se ha optimizado o en blanco si se simula el modelo adiabático).

Los datos que contiene OptimizaSt.out son:

- **VSWC**, Volumen de barrido de compresión (zona fría) [m<sup>3</sup>]
- **VSWE**, Volumen de barrido de expansión (zona caliente) [m<sup>3</sup>]
- **VCLC0**, Volumen libre de compresión (volumen libre de la zona fría) [m<sup>3</sup>]
- **VCLE0**, Volumen libre de expansión (volumen libre de la zona caliente) [m<sup>3</sup>]
- **PHASE**, Desfasaje de volumen [°]
- **T\_ENFRIADOR**, Tipo de enfriador: t)ubos a)nillo r)anurado m)alla de alambres
- **T\_CALENTADOR**, Tipo de calentador: t)ubos a)nillo r)anurado m)alla de alambres
- **T\_REGENERADOR**, Tipo de regenerador: t)ubular a)nular
- **T\_MATRIZ**, Tipo de matriz: m)alla de alambres l)ámina a)lambres apilados s)in matriz
- **DINK**, Diámetro interno del tubo del enfriador [m]
- **LK**, Longitud del enfriador [m]
- **DWIREK**, Diámetro de los alambres de la malla de alambre del enfriador [m]
- **POROSITYK**, Porosidad de la tela de alambres del enfriador (ver § A-5 de Ref. [1])
- **DOMAT**, Diámetro interno del cárter del tubo del regenerador [m]
- **ESPEJOR**, de las paredes del regenerador = (dout-domat)/2 [m]
- **LR**, Longitud del regenerador [m]
- **DAL**, Diámetro de los alambres de la malla de alambre del regenerador [m]
- **POROSITY**, Porosidad de la tela de alambres del regenerador (ver § A-5 de Ref. [1])
- **DINH**, Diámetro interno del tubo del calentador [m]
- **LH**, Longitud del calentador [m]
- **DWIREH**, Diámetro de los alambres de la malla de alambre del calentador [m]



- **POROSITYH**, Porosidad de la tela de alambres del calentador (ver § A-5 de Ref. [1])
- **T\_GAS**, Tipo de gas de trabajo: ai)re, he)lio, hi)drógeno
- **PMEAN**, Presión promedio [Pa]
- **TK**, Temperatura externa en el enfriador [K]
- **TH**, Temperatura externa en el calentador [K]
- **FREQ**, Velocidad de rotación de la manivela [Hz]

Como ejemplo de salida se presenta primero una salida sin optimización y luego con el motor optimizado.

- Sin Optimizar (los datos coinciden con los de entrada, ver OptimizaSt.dat):

```
&SALIDA
VCLC      = 2.993792429822560E-005,
VCLE      = 5.970167812546300E-005,
VSWC      = 9.817000000000000E-005,
VSWE      = 9.817000000000000E-005,
PHASE     = 90.000000000000000    ,
T_ENFRIADOR   = m,
T_CALENTADOR  = m,
T_REGENERADOR = t,
T_MATRIZ      = m,
DINK       = 8.000000000000000E-002,
LK         = 5.000000000000000E-003,
DWIREK     = 1.800000000000000E-004,
POROSITYK   = 0.632260000000000    ,
DOMAT      = 8.000000000000000E-002,
ESPESOR    = 5.000000000000000E-004,
LR         = 2.000000000000000E-002,
DAL        = 1.800000000000000E-004,
POROSITY    = 0.632260000000000    ,
DINH       = 0.900000000000000    ,
LH         = 5.000000000000000E-003,
DWIREH     = 1.800000000000000E-004,
POROSITYH   = 0.632260000000000    ,
T_GAS      = ai,
PMEAN      = 100000.0000000000    ,
TK0        = 300.0000000000000    ,
TH0        = 923.0000000000000    ,
FREQ       = 50.000000000000000    ,
/
```

- Con optimización (se modifican los parámetros optimizados):

```
&SALIDA
VCLC      = 3.350932691157014E-005,
VCLE      = 5.422371733386637E-005,
VSWC      = 9.817000000000000E-005,
VSWE      = 9.817000000000000E-005,
PHASE     = 90.000000000000000    ,
T_ENFRIADOR   = m,
T_CALENTADOR  = m,
T_REGENERADOR = t,
```



```
T_MATRIZ      = m,
DINK          = 6.425523233622733E-002,
LK            = 5.258491193846906E-003,
DWIREK       = 1.800000000000000E-004,
POROSITYK    = 0.632260000000000 ,
DOMAT        = 9.726777128250297E-002,
ESPESOR      = 5.000000000000000E-004,
LR           = 1.314285968030112E-002,
DAL          = 1.800000000000000E-004,
POROSITY     = 0.632260000000000 ,
DINH        = 0.120241307085822 ,
LH           = 3.796671908899896E-003,
DWIREH       = 1.800000000000000E-004,
POROSITYH    = 0.632260000000000 ,
T_GAS        = ai,
PMEAN        = 100000.000000000 ,
TKO          = 300.000000000000 ,
THO          = 923.000000000000 ,
FREQ         = 50.0000000000000
/
```

### B-5 LISTADO FUENTE

Seguidamente se lista el programa fuente y sus subrutinas. La mayoría de las subrutinas son idénticas a las utilizadas en el programa SitirSimple.for listado en el Anexo B de Ref. [1]. Seguidamente se listan sólo los archivos que fueron modificados.

#### PROGRAMA PRINCIPAL: *OptimizaSt*

```
*****
**          Optimizacion de motores Stirling usando el          **
**          Modelo Simple Análisis                             **
*****
** Tema      : Optimización de las performances de un motor Stirling**
** Descripcion: Modelo adiabático corregido por pérdidas de presión **
**           y transferencia de calor                             **
**           Modelo basado en el desarrollo del Dr. Israel Urieli **
*****
** Estructura del programa:                                     **
**                                                                 **
** 1      StirSimple                                           **
** 1.1    define                                               **
** 1.1.1  engine                                               **
** 1.1.1.1   sindrive                                           **
** 1.1.2    heatex                                             **
** 1.1.2.1   cooler                                             **
** 1.1.2.1.1   pipes                                             **
** 1.1.2.1.2   annulus                                           **
** 1.1.2.1.3   slots                                             **
** 1.1.2.1.4   tela                                             **
** 1.1.2.2    heater                                           **
** 1.1.2.2.1   pipes                                             **
** 1.1.2.2.2   annulus                                           **
** 1.1.2.2.3   slots                                             **
** 1.1.2.2.4   tela                                             **
** 1.1.2.3    regen                                           **
** 1.1.2.3.1   tubular                                           **
** 1.1.2.3.2   annular                                           **
** 1.1.2.3.3   matrix                                           **
** 1.1.2.3.3.1   mesh                                           **
** 1.1.2.3.3.2   foil                                           **
**                                                                 **
*****
```



```
** 1.1.2.3.3.3          alambre          **
** 1.1.2.3.3.4          nomatrix         **
** 1.1.2.3.4           ther_cond        **
** 1.1.3               gas              **
** 1.1.3.1             air              **
** 1.1.3.2             helium           **
** 1.1.3.3             hydrogen         **
** 1.1.4               operat          **
** 1.2                 Schmidt          **
** 1.2.1               plotpv          **
** 1.2.2               plotmass        **
** 1.3                 adiabatic        **
** 1.3.1               adiab           **
** 1.3.1.1             rk4             **
** 1.3.1.1.1          dadiab           **
** 1.3.1.1.1.1       volume           **
** 1.3.1.1.1.1.1    sinevol          **
** 1.3.1.2           filmatrix         **
** 1.3.2             plotadiab         **
** 1.4                 simple          **
** 1.4.1             adiab             **
** 1.4.1.1           rk4              **
** 1.4.1.1.1        dadiab           **
** 1.4.1.1.1.1     volume           **
** 1.4.1.1.1.1.1  sinevol          **
** 1.4.2             SimReg           **
** 1.4.2.1           reynum           **
** 1.4.2.2           frmatrix         **
** 1.4.2.3           frfoil           **
** 1.4.2.4           fralambre        **
** 1.4.3             SimHot           **
** 1.4.3.1           reynum           **
** 1.4.3.2           frpipe           **
** 1.4.4             SimKol           **
** 1.4.4.1           reynum           **
** 1.4.4.2           frpipe           **
** 1.4.5             SimWork          **
** 1.4.5.1           reynum           **
** 1.4.5.2           frmatrix         **
** 1.4.5.3           frfoil           **
** 1.4.5.4           frpipe           **
** 1.4.5.5           fralambre        **
** 1.4.5.6           plotloss         **
** 1.4.6             plotadiab         **
** 1.5                 Optimizacion    **
** 1.5.1             simple           **
** 1.5.2             smplx            **
** 1.5.2.1           costo            **
** 1.5.2.1.1        heatex           **
** 1.5.2.1.2        simple           **
*****
** Programa Principal : OptimizaSt.for **
** Autor               : Gustavo Scarpin **
** Fecha               : 31 de Mayo de 2012 **
** Ultima modificación: 08 de Octubre de 2012 **
*****
```

```
PROGRAM OptimizaSt
INCLUDE 'StirSimple.inc'
C Definir las variables locales
CHARACTER*1 respuesta,tipo_anal

NAMELIST/salida/vclc,vcle,vswc,vswe,phase,t_enfriador,t_calentador
.,t_regenerador,t_matriz,dink,lk,dwirek,porosityk,domat,espesor,
.lrdal,porosity,dinh,lh,dwireh,porosityh,t_gas,pmean,tk0,th0,freq

OPEN(UNIT=3,FILE='OptimizaSt.out',STATUS='UNKNOWN')

CALL Define
CALL Schmidt

C PREGUNTA SI SE QUIERE OPTIMIZAR, s) i ó n) o
WRITE(*,1000)
READ(*,'(A1)') respuesta
```



```
IF (respuesta.EQ.'s'.OR.respuesta.EQ.'S') CALL OPTIMIZACION
C INGRESA EL TIPO DE ANALISIS QUE SE QUIERE REALIZAR
C a)diabatico s)imple
WRITE(*,1010)
READ(*,'(A1)') tipo_anal

IF (tipo_anal.EQ.'a') THEN
CALL Adiabatic
ELSE IF (tipo_anal.EQ.'s') THEN
write(3,salida)
CALL Simple
ELSE
STOP 'FIN DEL PROGRAMA StirSimple'
ENDIF

1000 FORMAT(2X,'Corre optimizacion, s)i o n)o ? ')
1010 FORMAT(2X,'Tipo de simulacion a)diabatICA o s)imple ')

END PROGRAM OptimizaSt
```

### **FUNCION: Costo**

En negrita se muestra las variables que el usuario ha seleccionado para optimizar en este caso.

```
*****
**
**                               C O S T O . F O R
**
*****
**                               **
**   Calcula el costo de la minimización
**                               **
*****
**   PROGRAMA PRINCIPAL: SIM_CABURE.FOR
**                               **
**   AUTORES: Ing. Alejandra Rizzo
**             Ing. Gustavo H. Scarpin           ** CORDOBA, 08-FEB-2008 **
**                               **
*****

DOUBLE PRECISION FUNCTION COSTO(S)

INCLUDE 'StirSimple.inc'

REAL*8 S(*)
REAL*8 Eff, COST

NAMELIST/SalOpt/COST,dink,dinh,domat,dck,dch,lr,lk,lh,alpha

** Restricción

    domat = S(1)
    lr    = S(2)
    lk    = S(3)
    lh    = S(4)
    dck   = S(5)
    dch   = S(6)
    dink  = S(7)
    dinh  = S(8)

CALL Heatex

CALL Simple

COSTO = (actWpower - 140.D0)**2.D0 + 1.D2*(1.D0-acteff)**2.D0

COST = COSTO

WRITE(*,SalOpt)

END FUNCTION COSTO
```

**SUBROUTINAS**

En negrita se muestra las variables que el usuario puede modificar a voluntad para cambiar los parámetros que desee optimizar.

```
*****
**                                                                 **
**              O P T I M I Z A C I O N . F O R                    **
**                                                                 **
*****
** Permite optimizar funciones de múltiples variables             **
**                                                                 **
*****
** PROGRAMA PRINCIPAL: OptimizaSt.for                             **
**                                                                 **
** AUTORES: Ing. Gustavo H. Scarpin          ** CORDOBA, 14-JUL-2012 **
**                                                                 **
*****

SUBROUTINE OPTIMIZACION

  INCLUDE 'StirSimple.inc'

  INTEGER, PARAMETER :: Np = 8 !Número de parámetros a optimizar

  REAL*8 S(Np),DS(Np),Y0,YL,tolerancia,costo
  INTEGER M,iteraciones_maximas,i

  CALL Simple

** DATOS INICIALES DE LA OPTIMIZACION
  tolerancia      = 1.0D-10
  iteraciones_maximas = 100000

  S(1) = domat
  S(2) = lr
  S(3) = lk
  S(4) = lh
  S(5) = dck
  S(6) = dch
  S(7) = dink
  S(8) = dinh

** DATOS INICIALES DE LAS PERTURBACIONES
  DO i=1,Np
    DS(i) = S(i)/10.D0
  ENDDO

  M = iteraciones_maximas

  CALL SMPLX(Np,S,DS,tolerancia,M,Y0,YL)

  IF (tolerancia.GE.0.D0.AND.M.GE.iteraciones_maximas) THEN
    WRITE(*,*)'No pudo lograrse la estabilizacion'
    WRITE(*,*)'Se han superado las iteraciones maximas',M,tolerancia
    STOP
  ENDIF

END SUBROUTINE OPTIMIZACION

*****
**                                                                 **
**              S M P L X . F O R                                **
**                                                                 **
*****
** Este simple algoritmo minimiza COSTO(X), donde X es (Nx1).     **
** DX contiene las perturbaciones iniciales en X.                 **
** SD deberia ser fijada de acuerdo a la tolerancia requerida;    **
** cuando SD<0 el algoritmo llamará a COSTO M veces               **
**                                                                 **
*****
```



```
**
*****
**
** PROGRAMA PRINCIPAL: OptimizaSt.FOR
**
** AUTORES: Ing. Alejandra Rizzo
**           Ing. Gustavo H. Scarpin          ** CORDOBA, 08-FEB-2008
**
*****

SUBROUTINE SMPLX(N,X,DX,SD,M,Y0,YL)
IMPLICIT NONE

REAL*8 X(*),DX(*),XX(32),XC(32),Y(33),V(32,32)
REAL*8 D,Y0,YH,YL,YB,SDA,YR,YE,Y2,YC,COSTO,SD
INTEGER i,j,k,N,NH,NV,NL,M

NV=N+1

DO i=1,N
  DO j=1,NV
    V(i,j) = X(i)
  END DO
  V(i,i+1) = X(i)+DX(i)
END DO

Y0 = COSTO(X)
Y(1) = Y0

DO j=2,NV
  Y(j) = COSTO(V(1,j))
END DO

K = NV

4  YH = Y(1)
   YL = Y(1)
   NH = 1
   NL = 1
   DO j=2,NV
     IF (Y(j).GT.YH) THEN
       YH=Y(j)
       NH=j
     ELSE IF (Y(j).LT.YL) THEN
       YL = Y(j)
       NL = j
     END IF
   END DO
   YB = Y(1)
   DO j=2,NV
     YB = YB+Y(j)
   END DO
   YB = YB/NV

   D = 0.0
   DO j=1,NV
     D = D+(Y(j)-YB)**2
   END DO

   SDA = DSQRT(D/NV)

   IF ((K.GE.M).OR.(SDA.LE.SD)) THEN
     SD=SDA
     M=K
     YL = Y(NL)
     DO i=1,N
       X(i) = V(i,NL)
     END DO

     RETURN
   END IF

   DO i=1,N
     XC(i) = 0.0
     DO j=1,NV
       IF (j.NE.NH) XC(i) = XC(i)+V(i,j)
     END DO
   END DO
```



```
END DO
XC(i) = XC(i)/N
END DO

DO i=1,N
X(i) = XC(i)+XC(i)-V(i,NH)
END DO
K=K+1
YR=COSTO(X)

IF (YR.LT.YL) THEN
DO i=1,N
XX(i) = X(i)+X(i)-XC(i)
END DO
K=K+1
YE = COSTO(XX)
IF (YE.LT.YR) THEN
Y(NH) = YE
DO I=1,N
V(i,NH) = XX(i)
END DO
ELSE
Y(NH) = YR
DO I=1,N
V(i,NH) = X(i)
END DO
END IF
GOTO 4
END IF

Y2 = Y(NL)
DO j=1,NV
IF ((j.NE.NL).AND.(j.NE.NH).AND.(Y(j).GT.Y2)) Y2 = Y(j)
END DO
IF (YR.LT.YH) THEN
Y(NH)=YR
DO i=1,N
V(i,NH) = X(i)
END DO
IF (YR.LT.Y2) GOTO 4
END IF

DO i=1,N
XX(i) = 0.5D0*(V(i,NH)+XC(i))
END DO
K=K+1
YC=COSTO(XX)
IF (YC.LT.YH) THEN
Y(NH) = YC
DO i=1,N
V(i,NH) = XX(i)
END DO
ELSE
DO j=1,NV
DO i=1,N
V(i,j) = 0.5D0*(V(i,j)+V(i,NL))
END DO
IF (j.NE.NL) Y(j) = COSTO(V(1,j))
END DO
K=K+N
END IF

GOTO 4

END SUBROUTINE SEMPLX
```

---

```
*****
**                               Modelo Simple Análisis de un motor Stirling                               **
*****
** Tema       : Geometría de los intercambiadores de calor                                           **
** Descripcion: Especifica la geometría de tres tipos de intercamb.                                   **
*****
** Subroutine   : Heatex.for                                                                              **
** Programa Principal : StirSimple.for                                                                    **
** Autor       : Gustavo Scarpin                                                                           **
*****
```

---





\*\* Fecha : 2 de Junio de 2012 \*\*  
\*\* Ultima modificación: 2 de Junio de 2012 \*\*  
\*\*\*\*\*

```
SUBROUTINE Heatex

INCLUDE 'StirSimple.inc'

NAMELIST/Inter/t_enfriador,t_calentador,t_regenerador,
.
t_matriz

IF(iban.eq.0) READ(1,Inter)

CALL Cooler
CALL Conk
CALL Regen
CALL Conh
CALL Heater

iban = 1 !Bandera (1) Archivo leído

END SUBROUTINE Heatex
```

\*\*\*\*\*  
\*\* Subroutine: Cooler.for \*\*  
\*\*\*\*\*  
\*\* Tema : Especifica los parámetros geométricos del cooler \*\*  
\*\* Descricion: Definición de la geometría del enfriador \*\*  
\*\*\*\*\*

```
SUBROUTINE Cooler

INCLUDE 'StirSimple.inc'

NAMELIST/Enfriador/t_enfriador,vk,ak,awgk,dk,lk

C INGRESA LA GEOMETRIA SEGUN EL TIPO DE ENFRIADOR
C t)ubos a)nillo r)anurado

IF (t_enfriador.EQ.'t') THEN

CALL Pipes(iban,'k',vk,ak,awgk,dk,lk,numk) !Enfriador formado por tubos

ELSE IF (t_enfriador.EQ.'a') THEN

CALL Annulus(iban,'k',vk,ak,awgk,dk,doutk,dink,lk) !Enfriador tipo anillo

ELSE IF (t_enfriador.EQ.'r') THEN

CALL Slots(iban,'k',vk,ak,awgk,dk,wk,hk,lk,numk) !Enfriador ranurado

ELSE IF (t_enfriador.EQ.'m') THEN

CALL Tela(iban,'k',vk,ak,awgk,dk,dink,lk,dwirek,porosityk) !Enfriador tipo malla de alambres

ELSE

STOP 'TIPO DE ENFRIADOR NO DEFINIDO'

END IF

WRITE(*,Enfriador)

END SUBROUTINE Cooler
```

\*\*\*\*\*  
\*\* Subroutine: Heater.for \*\*  
\*\*\*\*\*  
\*\* Tema : Especifica los parámetros geométricos del heater \*\*  
\*\* Descricion: Definición de la geometría del calentador \*\*  
\*\*\*\*\*

```
SUBROUTINE Heater
```



```
INCLUDE 'StirSimple.inc'

NAMELIST/Calentador/t_calentador,vh,ah,awgh,dh,lh

C INGRESA LA GEOMETRIA SEGUN EL TIPO DE CALENTADOR
C t)ubos a)nillo r)anurado

IF (t_calentador.EQ.'t') THEN

    CALL Pipes(iban,'h',vh,ah,awgh,dh,lh,numh)                !Calentador formado por tubos

ELSE IF (t_calentador.EQ.'a') THEN

    CALL Annulus(iban,'h',vh,ah,awgh,dh,douth,dinh,lh)      !Calentador tipo anillo

ELSE IF (t_calentador.EQ.'r') THEN

    CALL Slots(iban,'h',vh,ah,awgh,dh,wh,hh,lh,numh)         !Calentador ranurado

ELSE IF (t_calentador.EQ.'m') THEN

    CALL Tela(iban,'h',vh,ah,awgh,dh,dinh,lh,dwireh,porosityh) !Calentador tipo malla de
alambres

ELSE

    STOP 'TIPO DE CALENTADOR NO DEFINIDO'

END IF

WRITE(*,Calentador)

END SUBROUTINE Heater

*****
** Subroutine:          Pipes.for                               **
*****
** Tema                : Geometria de un intercambiador formado por caños **
** Descripcion:        Se consideran caños lisos                **
*****

SUBROUTINE Pipes(iban,tipo,v,a,awg,d,len,num)

C Definir las variables locales
CHARACTER*1 tipo      !Tipo de intercambiador de calor
INTEGER iban         !Bandera de lectura (0 lee, 1 no lee)
REAL*8 num           !Número de caños del paquete
REAL*8 v             !Volumen libre del intercambiador de calor [m^3]
REAL*8 a             !Area de flujo libre interna del intercambiador de calor [m^2]
REAL*8 awg           !Area mojada interna del intercambiador de calor [m^2]
REAL*8 d             !Diámetro interno del tubo [m]
REAL*8 len           !Longitud del intercambiador de calor [m]

REAL*8, PARAMETER :: PI = 3.14159265359D0

NAMELIST/Geom_Enf/d,len,num
NAMELIST/Geom_Cal/d,len,num

C Lee por archivo la geometria del intercambiador de calor

IF (tipo.EQ.'k') THEN
    IF(iban.eq.0) READ(1,Geom_Enf)
ELSE IF (tipo.EQ.'h') THEN
    IF(iban.eq.0) READ(1,Geom_Cal)
ELSE
    STOP 'Error en la subrutina Pipes'
END IF

C Cálculo de las propiedades geométricas del intercambiador de calor

a = num*pi*d*d/4.D0
v = a*len
awg = num*pi*d*len

END SUBROUTINE Pipes
```



```
*****
** Subroutine:          Annulus.for          **
*****
** Tema                : Geometria de un intercambiador anular          **
** Descripcion:        Se consideran un sólo anillo                      **
*****

      SUBROUTINE Annulus(iban,tipo,v,a,awg,d,dout,din,len)

C Definir las variables locales
CHARACTER*1 tipo      !Tipo de intercambiador de calor
INTEGER iban         !Bandera de lectura (0 lee, 1 no lee)
REAL*8  dout        !Diámetro externo de la ranura del anillo [m]
REAL*8  din         !Diámetro interno de la ranura del anillo [m]
REAL*8  v           !Volumen libre del intercambiador de calor [m^3]
REAL*8  a           !Area de flujo libre interna del intercambiador de calor [m^2]
REAL*8  awg        !Area mojada interna del intercambiador de calor [m^2]
REAL*8  d           !Diámetro hidráulico [m]
REAL*8  len        !Longitud del intercambiador de calor [m]

      REAL*8, PARAMETER :: PI = 3.14159265359D0

      NAMELIST/Geom_Enf/dout,din,len
      NAMELIST/Geom_Cal/dout,din,len

C Lee por archivo la geometria del intercambiador de calor

      IF (tipo.EQ.'k') THEN
        IF(iban.eq.0) READ(1,Geom_Enf)
      ELSE IF (tipo.EQ.'h') THEN
        IF(iban.eq.0) READ(1,Geom_Cal)
      ELSE
        STOP 'Error en la subrutina Annulus'
      END IF

C Cálculo de las propiedades geométricas del intercambiador de calor

      a  = pi*(dout*dout - din*din)/4.D0
      v  = a*len
      awg = pi*dout*len
      d  = dout - din

      END SUBROUTINE Annulus

*****
** Subroutine:          Slots.for          **
*****
** Tema                : Geometria de un intercambiador ranurado          **
** Descripcion:        Se consideran un sólo ranurado                      **
*****

      SUBROUTINE Slots(iban,tipo,v,a,awg,d,w,h,len,num)

C Definir las variables locales
CHARACTER*1 tipo      !Tipo de intercambiador de calor
INTEGER iban         !Bandera de lectura (0 lee, 1 no lee)
REAL*8  num         !Número de ranuras del paquete
REAL*8  w           !Ancho de la ranura [m]
REAL*8  h           !Alto de la ranura [m]
REAL*8  v           !Volumen libre del intercambiador de calor [m^3]
REAL*8  a           !Area de flujo libre interna del intercambiador de calor [m^2]
REAL*8  awg        !Area mojada interna del intercambiador de calor [m^2]
REAL*8  d           !Diámetro hidráulico [m]
REAL*8  len        !Longitud del intercambiador de calor [m]

      REAL*8, PARAMETER :: PI = 3.14159265359D0

      NAMELIST/Geom_Enf/w,h,len,num
      NAMELIST/Geom_Cal/w,h,len,num

C Lee por archivo la geometria del intercambiador de calor

      IF (tipo.EQ.'k') THEN
        IF(iban.eq.0) READ(1,Geom_Enf)
      ELSE IF (tipo.EQ.'h') THEN
```



```
IF(iban.eq.0) READ(1,Geom_Cal)
ELSE
  STOP 'Error en la subrutina Slots'
END IF
```

C Cálculo de las propiedades geométricas del intercambiador de calor

```
a = num*w*h
v = a*len
awg = num*(w + 2.D0*h)*len
d = 4.D0*v/awg
```

END SUBROUTINE Slots

```
*****
** Subroutine:          Tela.for          **
*****
** Tema      : Geometria de un intercambiador formado por telas **
**          : metllicas                    **
** Descripcion:          **
*****
```

SUBROUTINE Tela(iban,tipo,v,a,awg,d,din,len,dwire,porosity)

C Definir las variables locales

```
CHARACTER*1 tipo !Tipo de intercambiador de calor
INTEGER iban !Bandera de lectura (0 lee, 1 no lee)
REAL*8 v !Volumen libre del intercambiador de calor [m^3]
REAL*8 a !Area de flujo libre interna del intercambiador de calor [m^2]
REAL*8 awg !Area mojada interna del intercambiador de calor [m^2]
REAL*8 d !Diámetro hidráulico[m]
REAL*8 len !Longitud del intercambiador de calor [m]
REAL*8 porosity !Porosidad de la tela de alambres
REAL*8 din !Diámetro interno del tubo [m]
REAL*8 dwire !Diámetro de los alambres de la tela [m]
```

REAL\*8, PARAMETER :: PI = 3.14159265359D0

```
NAMELIST/Geom_Enf/din,len,dwire,porosity
NAMELIST/Geom_Cal/din,len,dwire,porosity
```

C Lee por archivo la geometria del intercambiador de calor

```
IF (tipo.EQ.'k') THEN
  IF(iban.eq.0) READ(1,Geom_Enf)
ELSE IF (tipo.EQ.'h') THEN
  IF(iban.eq.0) READ(1,Geom_Cal)
ELSE
  STOP 'Error en la subrutina Tela'
END IF
```

C Cálculo de las propiedades geométricas del intercambiador de calor

```
awgr0 = pi*din*len

a = pi*din*din/4.D0*porosity
v = a*len
d = dwire*porosity/(1.D0 - porosity)
awg = 4.D0*v/d + awgr0
```

END SUBROUTINE Tela

```
*****
** Modelo Simple Análisis de un motor Stirling **
*****
** Tema      : Geometria y propiedades térmicas del regenerador **
** Descripcion: Se especifica la geometría de dos tipos de **
**          : regeneradores: tubulares y anulares **
*****
** Subroutine      : Regen.for **
** Programa Principal : StirSimple.for **
** Autor          : Gustavo Scarpin **
** Fecha         : 2 de Junio de 2012 **
** Última modificación: 2 de Junio de 2012 **
*****
```



```
SUBROUTINE Regen
INCLUDE 'StirSimple.inc'
NAMELIST/Regenerador/t_regenerador,vr,ar,awgr,dr,lr,awgr0,cqwr,
.kwr,amat,awr,porosity
CALL Matrix !Propiedades de la matriz
CALL Ther_Cond !Cálculo de la conductividad térmica
WRITE(*,Regenerador)
END SUBROUTINE Regen

*****
** Subroutine: Matrix.for **
*****
** Tema : Especifica la geometría de la matriz del regenerador **
** Descripcion: Se define tres tipos de matrices **
*****

SUBROUTINE Matrix
INCLUDE 'StirSimple.inc'
C INGRESA LA GEOMETRIA SEGUN EL TIPO DE MATRIZ DEL REGENERADOR
C m)alla l)lámina s)in matriz
IF (t_matriz.EQ.'m') THEN
CALL Mesh !Matriz tipo malla de alambres
ELSE IF (t_matriz.EQ.'l') THEN
CALL Foil !Matriz tipo lámina metálica plegada
ELSE IF (t_matriz.EQ.'a') THEN
CALL Alambre !Matriz tipo alambres apilados
ELSE IF (t_matriz.EQ.'s') THEN
CALL NoMatrix !Sin matriz
ELSE
STOP 'TIPO DE MATRIZ NO DEFINIDA'
END IF
END SUBROUTINE Matrix

*****
** Subroutine: Mesh.for **
*****
** Tema : Geometria del regenerador con matriz tipo **
** malla de alambres **
** Descripcion: **
*****

SUBROUTINE Mesh
INCLUDE 'StirSimple.inc'
C Definir las variables locales
NAMELIST/Geom_Malla/porosity,dal,domat,espesor,lr
C Lee por archivo la geometria del regenerador
IF (iban.eq.0) READ(1,Geom_Malla)
```



```
C Cálculo de las propiedades geométricas del regenerador
dout = domat+espesor*2.D0
awgr0 = pi*domat*lr
amat = pi*domat*domat/4.D0
awr = pi*(dout*dout - domat*domat)/4.D0

ar = amat*porosity
vr = ar*lr
dr = dal*porosity/(1.D0 - porosity)
awgr = 4.D0*vr/dr + awgr0

END SUBROUTINE Mesh

*****
** Subroutine: Foil.for **
*****
** Tema : Geometria del regenerador con matriz tipo **
** lámina metálica plegada **
** Descripcion: **
*****

SUBROUTINE Foil

INCLUDE 'StirSimple.inc'

C Definir las variables locales
REAL*8 am !Area de la lámina [m^2]

NAMELIST/Geom_Malla/fl,tf

C Lee por archivo la geometria del regenerador

IF (iban.eq.0) READ(1,Geom_Malla)

C Cálculo de las propiedades geométricas del regenerador

am = tf*fl
ar = amat - am
vr = ar*lr
awgr = 2.D0*lr*fl + awgr0
dr = 4.D0*vr/awgr
porosity = ar/amat

END SUBROUTINE Foil

*****
** Subroutine: Alambre.for **
*****
** Tema : Geometria del regenerador con matriz tipo **
** lámina metálica plegada **
** Descripcion: **
*****

SUBROUTINE Alambre

INCLUDE 'StirSimple.inc'

NAMELIST/Geom_Malla/dal,nal,espesor,lr

C Lee por archivo la geometria del regenerador

IF (iban.eq.0) READ(1,Geom_Malla)

C Cálculo de las propiedades geométricas del regenerador
amat = DSQRT(3.D0)*dal*dal/2.D0*nal
domat = DSQRT(4.D0*amat/pi)
dout = domat+espesor*2.D0
awr = pi*(dout*dout - domat*domat)/4.D0
awgr0 = pi*domat*lr

ar = (DSQRT(3.D0)-PI/2.D0)*dal*dal/2.D0*nal
vr = ar*lr
awgr = PI*dal*lr*nal + awgr0
dr = DSQRT((DSQRT(3.D0)-PI/2)*2.D0/PI)*dal
porosity = 1.D0-PI/(2.D0*DSQRT(3.D0))
```



```
END SUBROUTINE Alambre
```

```
*****
** Subroutine:           NoMatrix.for           **
*****
** Tema                : Geometria del regenerador sin matriz **
** Descripcion:         **
*****
```

```
SUBROUTINE NoMatrix
```

```
INCLUDE 'StirSimple.inc'
```

```
C Cálculo de las propiedades geométricas del regenerador
```

```
ar   = amat
vr   = ar*lr
awgr = awgr0
dr   = 4.D0*vr/awgr
porosity = 1.D0
```

```
END SUBROUTINE NoMatrix
```

```
*****
** Subroutine:           Ther_Cond.for          **
*****
** Tema                : Conductividad térmica del carter del regenerador **
** Descripcion:         Se asume un carter fabricado de acero inoxidable. **
**                    A pesar de que depende de la temperatura, se asume **
**                    un valor constante de conductividad termica: **
**                    25 [W/m/K] para condiciones normales **
*****
```

```
SUBROUTINE Ther_Cond
```

```
INCLUDE 'StirSimple.inc'
```

```
C Cálculo de las propiedades geométricas del regenerador
```

```
c   kwr = 25.D0           !Conductividad térmica del Acero [W/m/K]
c   kwr = 0.3D0          !Conductividad térmica del Caucho Vulcanizado [W/m/K]
c   kwr = 0.5D0          !Conductividad térmica del Caucho Vulcanizado [W/m/K] Sugerido por Omar
c   kwr = 0.03D0         !Conductividad térmica del Aerogel de sílice [W/m/K]
c   cqwr = kwr*awr/lr    !Conductividad térmica de la pared del carter del regenerador [W/K]
```

```
END SUBROUTINE Ther_Cond
```

### **INCLUDE: StirSimple.inc**

```
*****
**                    Modelo Simple Análisis de un motor Stirling **
*****
** Tema              : Definicion de variables **
** Descripcion:      Definicion de variables globales **
*****
** Incluir           : StirSimple.inc **
** Programa Principal : OptimizaSt.for **
** Autor            : Gustavo Scarpin **
** Fecha            : 31 de Mayo de 2012 **
** Ultima modificación: 08 de Octubre de 2012 **
*****
```

```
IMPLICIT NONE
```

```
C DEFINICION DE CONSTANTES GLOBALES
```

```
REAL*8, PARAMETER :: PI = 3.14159265359D0
REAL*8, PARAMETER :: grad_a_rad = 0.01745329252D0
REAL*8, PARAMETER :: rad_a_grad = 57.2957795131D0
```

```
C INDICE DE LAS FILAS DE LOS ARREGLOS var Y dvar.
```

```
INTEGER, PARAMETER :: TC = 1 !Temperatura de la zona de compresión [K]
INTEGER, PARAMETER :: TE = 2 !Temperatura de la zona de expansión [K]
INTEGER, PARAMETER :: QK = 3 !Calor transferido al enfriador [J]
```



```
INTEGER, PARAMETER :: QR = 4 !Calor transferido al regenerador [J]
INTEGER, PARAMETER :: QH = 5 !Calor transferido al calentador [J]
INTEGER, PARAMETER :: WC = 6 !Trabajo realizado por la zona de compresión [J]
INTEGER, PARAMETER :: WE = 7 !Trabajo realizado por la zona de expansión [J]
INTEGER, PARAMETER :: W = 8 !Trabajo total realizado (WC + WE) [J]
INTEGER, PARAMETER :: P = 9 !Presión [J]
INTEGER, PARAMETER :: VC = 10 !Volumen de la zona de compresión [m^3]
INTEGER, PARAMETER :: VE = 11 !Volumen de la zona de expansión [m^3]
INTEGER, PARAMETER :: MC = 12 !Masa de gas en la zona de compresión [kg]
INTEGER, PARAMETER :: MK = 13 !Masa de gas en el enfriador [kg]
INTEGER, PARAMETER :: MR = 14 !Masa de gas en el regenerador [kg]
INTEGER, PARAMETER :: MH = 15 !Masa de gas en el calentador [kg]
INTEGER, PARAMETER :: ME = 16 !Masa de gas en la zona de expansión [kg]
INTEGER, PARAMETER :: TCK = 17 !Temperatura condicional zona de compresión / enfriador [K]
INTEGER, PARAMETER :: THE = 18 !Temperatura condicional calentador / zona de expansión [K]
INTEGER, PARAMETER :: GACK = 19 !Flujo de masa condicional zona de compresión / enfriador
[kg/rad]
INTEGER, PARAMETER :: GAKR = 20 !Flujo de masa condicional enfriador / regenerador [kg/rad]
INTEGER, PARAMETER :: GARH = 21 !Flujo de masa condicional regenerador / calentador [kg/rad]
INTEGER, PARAMETER :: GAHE = 22 !Flujo de masa condicional calentador / zona de expansión
[kg/rad]

C DIMENSIONAMIENTO DE LOS ARREGLOS var Y dvar
INTEGER, PARAMETER :: ROWV = 22 !Número de filas de la matriz var
INTEGER, PARAMETER :: ROWD = 16 !Número de filas de la matriz dvar
INTEGER, PARAMETER :: COL = 37 !Número de columnas en las matrices (cada 10 grados)

REAL*8 var(ROWV,COL)
REAL*8 dvar(ROWD,COL)

COMMON/estado/var,dvar

C PARAMETROS GEOMETRICOS DEL MOTOR
REAL*8 vclc, vcle ! Volumenes libres de compresión y expansion [m^3]
REAL*8 vclc0, vcle0 ! Volumenes libres de las zonas compresión y expansion unicamente [m^3]
REAL*8 vswc, vswc ! Volumenes de barrido de compresión y expansion [m^3]
REAL*8 alpha ! Angulo de fase de avance de la zona de expansión [radianes]
REAL*8 phase ! Angulo de fase de avance de la zona de expansión [grados]

C PARAMETROS GEOMETRICOS DEL ENFRIADOR
CHARACTER*1 t_enfriador !(t)ubos a)nillo r)anurado m)matriz
REAL*8 numk ! Número de caños o ranuras del paquete del enfriador
REAL*8 vk ! Volumen libre del enfriador [m^3]
REAL*8 ak ! Area de flujo libre interna del enfriador [m^2]
REAL*8 awgk ! Area mojada interna del enfriador [m^2]
REAL*8 dk ! Diámetro hidráulico del enfriador [m]
REAL*8 lk ! Longitud efectiva del enfriador [m]
REAL*8 doutk ! Diámetro externo de la ranura del enfriador tipo anillo [m]
REAL*8 dink ! Diámetro interno [m]
REAL*8 wk ! Ancho de la ranura del enfriador tipo Slot [m]
REAL*8 hk ! Alto de la ranura del enfriador tipo Slot [m]
REAL*8 porosityk ! Porosidad de la tela de alambres
REAL*8 dwirek ! Diámetro de los alambres de la tela [m]

C PARAMETROS GEOMETRICOS DEL CALENTADOR
CHARACTER*1 t_calentador !(t)ubos a)nillo r)anurado m)matriz
REAL*8 numh ! Número de caños o ranuras del paquete del calentador
REAL*8 vh ! Volumen libre del calentador [m^3]
REAL*8 ah ! Area de flujo libre interna del calentador [m^2]
REAL*8 awgh ! Area mojada interna del calentador [m^2]
REAL*8 dh ! Diámetro hidráulico del calentador [m]
REAL*8 lh ! Longitud efectiva del calentador [m]
REAL*8 douth ! Diámetro externo de la ranura del calentador tipo anillo [m]
REAL*8 dinh ! Diámetro interno [m]
REAL*8 wh ! Ancho de la ranura del calentador tipo Slot [m]
REAL*8 hh ! Alto de la ranura del calentador tipo Slot [m]
REAL*8 porosityh ! Porosidad de la tela de alambres
REAL*8 dwireh ! Diámetro de los alambres de la tela [m]

C PARAMETROS GEOMETRICOS DEL TUBO DE CONEXION DEL ENFRIADOR
REAL*8 ack ! Area de flujo libre interna del tubo del enfriador [m^2]
REAL*8 awgck ! Area mojada interna del tubo del enfriador [m^2]
REAL*8 dck ! Diámetro hidráulico del tubo del enfriador [m]
REAL*8 lck ! Longitud efectiva del tubo del enfriador [m]
```





```
C PARAMETROS GEOMETRICOS DEL TUBO DE CONEXION DEL CALENTADOR
REAL*8 ach      ! Area de flujo libre interna del tubo del calentador [m^2]
REAL*8 awgch   ! Area mojada interna del tubo del calentador [m^2]
REAL*8 dch     ! Diámetro hidráulico del tubo del calentador [m]
REAL*8 lch     ! Longitud efectiva del tubo del calentador [m]

C PARAMETROS GEOMETRICOS DEL REGENERADOR
CHARACTER*1 t_regenerador  !t)ubular a)nular
CHARACTER*1 t_matriz      !m)alla l)ámina s)in matriz
REAL*8 vr      ! Volumen libre del regenerador [m^3]
REAL*8 ar      ! Area de flujo libre interna del regenerador [m^2]
REAL*8 awgr    ! Area mojada interna del regenerador [m^2]
REAL*8 dr      ! Diámetro hidráulico del regenerador [m]
REAL*8 lr      ! Longitud efectiva del regenerador [m]
REAL*8 awgr0   ! Area mojada del regenerador sin matriz [m^2]
REAL*8 cqwr    ! Conductividad térmica del carter del regenerador [m^2]
REAL*8 kwr     ! Conductividad térmica del regenerador [W/m/K]
REAL*8 amat    ! Area de la matriz del regenerador [m^2]
REAL*8 awr     ! Area de pared del carter del regenerador [m^2]
REAL*8 porosity ! Porosidad de la matriz del regenerador [-]
REAL*8 fl      ! Longitud de la lámina metálica sin plegar [m]
REAL*8 tf      ! Espesor de la lámina metálica [m]
REAL*8 dout    ! Diámetro externo del carter del tubo [m]
REAL*8 domat   ! Diámetro interno del carter del tubo [m]
REAL*8 dimat   ! Diámetro interno de la matriz [m]
REAL*8 dal     ! Diámetro del alambre [m]
REAL*8 nal     ! Real indicando el número de alambre aproximados
REAL*8 espesor ! Espesor de la pared del regenerador

COMMON/geometria/vclc,vcle,vswc,vswe,alpha,phase,vk,ak,awgk,dk,lk,
. numk,doutk,dink,wk,hk,porosityk,dwirek,
. numh,douth,dinh,wh,hh,porosityh,dwireh,espesor,
. amat,awr,porosity,fl,tf,dout,domat,dimat,dal,nal,
. vh,ah,awgh,dh,lh,vr,ar,awgr,dr,lr,awgr0,cqwr,kwr,
. ack,awgck,dck,lck,ach,awgch,dch,lch,vclc0,vcle0,
. t_enfriador,t_calentador,t_regenerador,t_matriz

C PARAMETROS DEL GAS DE TRABAJO
CHARACTER*2 t_gas  ! ai)re he)llo hi)drógeno
REAL*8 rgas       ! Constante del gas [J/kg/K]
REAL*8 cp         ! Calor específico a presión constante [J/kg/K]
REAL*8 cv         ! Calor específico a volumen constante [J/kg/K]
REAL*8 gama       ! Constante adiabática, relación cp/cv [-]
REAL*8 mu0        ! Viscosidad dinámica a la temperatura de referencia t0 [kg*m/s]
REAL*8 t0, t_suth ! Temperatura de referencia [K], Constante de Sutherland [K]
REAL*8 prandtl    ! Número de Prandtl

COMMON/dato_gas/rgas,cp,cv,gama,mu0,t0,t_suth,prandtl,t_gas

C PARAMETROS DE OPERACION
REAL*8 pmean      ! Presión (de carga) media [Pa]
REAL*8 tk, tr, th ! Temperaturas del enfriador, regenerador y calentador [K]
REAL*8 tk0, th0  ! Temperaturas exterior del enfriador y calentador [K]
REAL*8 freq, omega ! Frecuencia del ciclo [Herz], [rad/seg]
REAL*8 mgas      ! Masa total de gas en el motor [kg]
REAL*8 mvol      ! Volumen promedio del motor [m^3]
REAL*8 beta      ! Angulo de fase de la presión [rad]

COMMON/operation/pmean,tk,tr,th,freq,omega,mgas,mvol,beta,tk0,th0

C PARAMETROS DE OPTIMIZACION
INTEGER iban      !Bandera de lectura (0 lee, 1 no lee)
REAL*8 actWpower  !Potencia real de salida del motor [W]
REAL*8 actQhpower !Calor real transf. al calentador [W]
REAL*8 acteff     !Eficiencia termica real del motor[-]

COMMON/optimiza/iban,actWpower,actQhpower,acteff
```