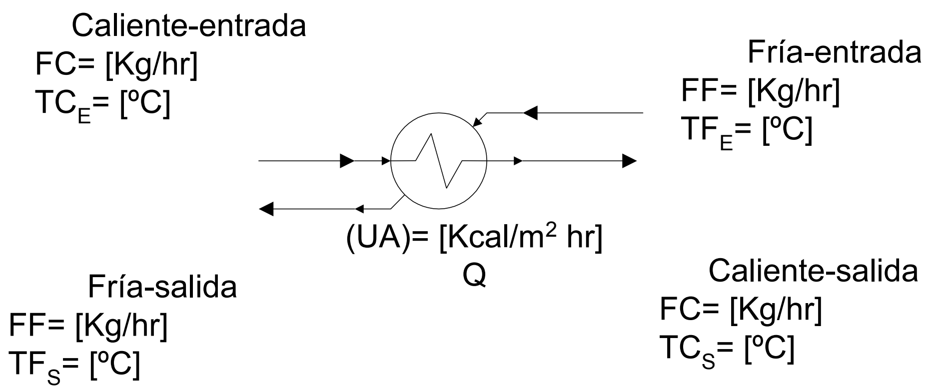
Métodos numéricos. Aplicación de herramientas informáticas.

Caso de aplicación 1: Intercambiador de calor.

Sean dos corrientes que intercambian solamente calor a través de un intercambiador de calor a contracorriente. Los balances de materia global y por componentes resultan ser triviales dado que no hay mezcla ni divisiones de corrientes. Tampoco hay cambio de estado ni reacciones químicas. La figura siguiente representa al mencionado equipo:



Como resultado el sistema conformado se puede formular a través de las siguientes ecuaciones.

Fluido Caliente $$Q = FC \times \overline{\Delta H}_C$$

Fluido Frio $$Q = FF \times \overline{\Delta H}_F$$

Balance Global $$Q = (UA) \times \mathbf{\Delta Tln}$$

Dif de temp media log $$\mathbf{\Delta Tln} = \frac{(TC_E - TF_S) - (TC_S - TF_E)}{\mathbf{ln}\dfrac{(TC_E - TF_S)}{(TC_S - TF_E)}}$$

Siendo además las propiedades fisicoquímicas:

$$\overline{\Delta H}_C = \overline{\Delta H}_C(Tc_E) - \overline{\Delta H}_C(Tc_S)$$
$$\overline{\Delta H}_F = \overline{\Delta H}_F(TF_S) - \overline{\Delta H}_F(TF_E)$$

Una forma de calcular las entalpías es través de las capacidades caloríficas a presión constante (Cp’s). Si asumimos Cp constante:

$$
\begin{array}{ll}
Ec\text{-}1 & Q - FC \times \overline{Cp}_C \times (TC_E - TC_S) = 0 \\
Ec\text{-}2 & Q - FF \times \overline{Cp}_F \times (TF_S - TF_E) = 0 \\
Ec\text{-}3 & Q - (UA) \times \mathbf{\Delta Tln} = \mathbf{0} \\
Ec\text{-}4 & \mathbf{\Delta Tln} - \dfrac{(TC_E - TF_S) - (TC_S - TF_E)}{\mathbf{ln}\dfrac{(TC_E - TF_S)}{(TC_S - TF_E)}} = 0
\end{array}
$$

Se define como grados de libertad de un sistema a:

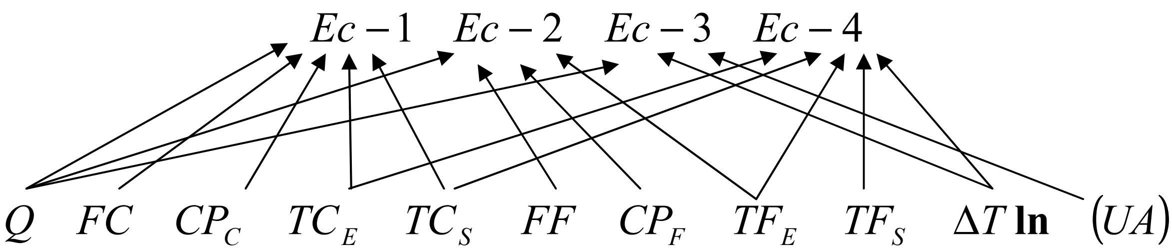
$$GL = NV - NE = 11 - 4 = 7$$

Donde: NV es el número de variables y NE el número de ecuaciones independientes. Para que el sistema tenga una solución única, los grados de libertad deben ser cero por lo que deben fijarse un número GL de variables (ahora llamados parámetros). No obstante, el orden de resolución del sistema resultante depende de cuáles de las variables se asumen como parámetros y cuáles como incógnitas. En ocasiones es factible hacer dicha selección a fin de reducir la complejidad del mismo mientras que otras veces viene impuesto por el mismo problema. A modo ilustrativo definiremos dos casos posibles.

Siendo la funcionalidad de las ecuaciones:

$$
\begin{array}{c}
Ec\text{-}1 = f_1(Q,\ FC,\ CP_C,\ TC_E,\ TC_S) \\
Ec\text{-}2 = f_2(Q,\ FF,\ CP_F,\ TF_E,\ TF_S) \\
Ec\text{-}3 = f_2(Q,\ \mathbf{\Delta Tln},\ (UA)) \\
Ec\text{-}4 = f_4(\mathbf{\Delta Tln},\ TC_E,\ TC_S,\ TF_E,\ TF_S)
\end{array}
$$

O, en forma gráfica:



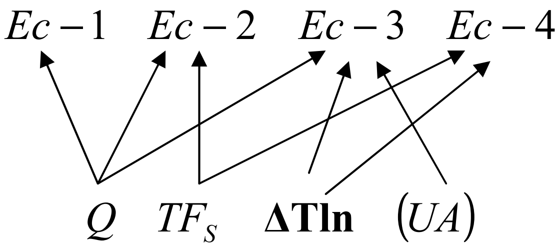
Caso a) entradas totalmente definidas y temperatura de la corriente caliente a la salida, dato.

Serán conocidas pues: FC, CP_C, TC_E, FF, CP_F, TF_E (entradas) y TC_S (salida caliente)= 7 variables/parámetros. Quedando como incógnitas Q, TF_S, ΔTln y (UA) = 4 variables=GL.

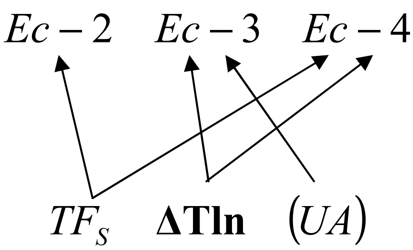
$$GL = NV - NE = 4 - 4 = 0$$

$$Ec\text{-}1 = f_1(Q)$$
$$Ec\text{-}2 = f_2(Q,\ TF_S)$$
$$Ec\text{-}3 = f_2(Q,\ \mathbf{\Delta Tln},\ (UA))$$
$$Ec\text{-}4 = f_4(\mathbf{\Delta Tln},\ \ TF_S)$$

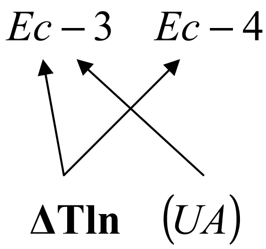
El árbol de variables (eliminando los parámetros) quedan:



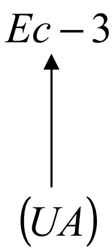
Como se aprecia, la ecuación 1 sólo depende de una variable (Q) la cual se puede explicitar de allí. Conocida Q y eliminada Ec-1 queda:



De la ecuación 2 se puede explicitar TF_S. Se elimina Ec-2:



La ecuación 4 depende sólo de ΔTln de la cual se despeja. Calculada ΔTln se elimina Ec-4. Queda:



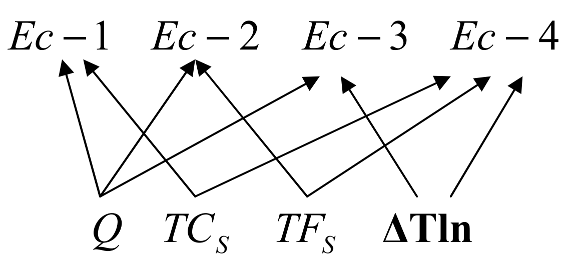
Finalmente, la última incógnita (UA) se calcula de la ecuación 3 con lo que el sistema queda totalmente resuelto.

Este tipo de problemas se denomina acíclico ya que su resolución es lineal. Existen casos en que se genera una estructura cíclica en la que no es posible operar en forma directa por lo que debe recurrirse a métodos iterativos para su resolución.

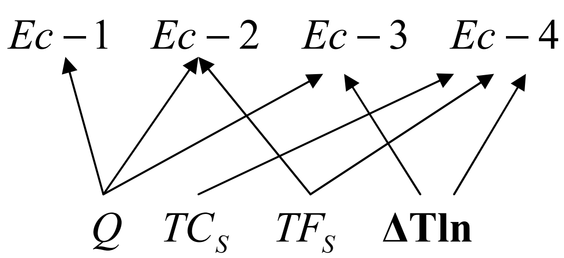
Caso b) entradas totalmente definidas y valor (UA) dato.

Al igual que antes tenemos un sistema de 4 ecuaciones (las mismas) y 4 incógnitas. Estas son Q, TCS, TFS y ΔTln.

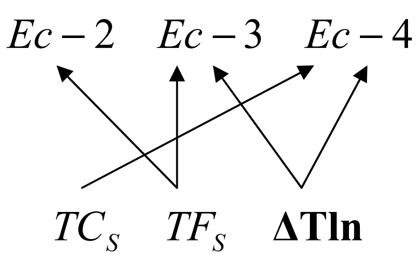
$$Ec\text{-}1 = f_1(Q,\ \ TC_S)$$
$$Ec\text{-}2 = f_2(Q,\ TF_S)$$
$$Ec\text{-}3 = f_2(Q,\ \mathbf{\Delta Tln})$$
$$Ec\text{-}4 = f_4(\mathbf{\Delta Tln},\ TC_S,\ TF_S)$$



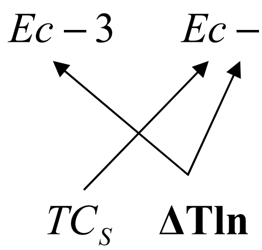
Se aprecia aquí que ninguna ecuación depende de una sola variable por lo que conforman un sistema cíclico. Un método de resolución es estimando una variable que pasaría a ser, momentáneamente, un parámetro. Sea por ejemplo darle un valor supuesto a la variable TC_S. Ahora veremos que el sistema queda:



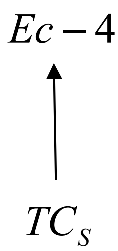
De la ecuación 1 se puede calcular Q. se eliminan variable y ecuación quedando:



La ecuación 2 depende sólo de TFS la cual puede calcularse. Nos queda:



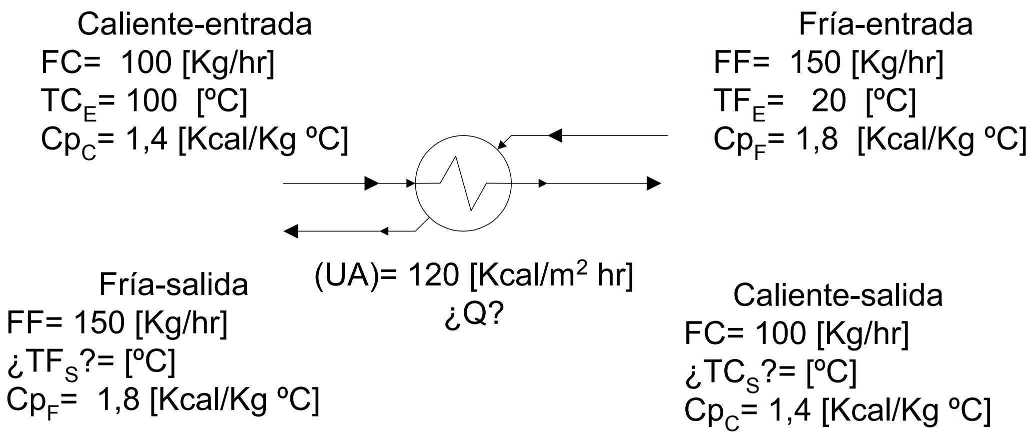
De la ecuación 3 podemos despejar ΔTln y eliminar la ecuación 3. Finalmente queda:



De la ecuación 4 se puede calcular TC_S. Este valor puede ser luego comparado con el valor estimado inicialmente. Si el error está dentro de lo tolerable se obtuvo la solución, de lo contrario se adopta el último valor de TC_S como valor estimado y se repite el ciclo.

No obstante lo expuesto anteriormente debe notarse que la variable TC_S no puede explicitarse de la Ec-4 por lo que también debe calcularse por métodos iterativos. A modo de ejemplo resolveremos el siguiente ejemplo numérico:

Ejemplo 1:



Siguiendo el esquema anterior plantearemos el sistema en el orden de resolución obtenido. El esquema será encuadrado en un modelo denominado caja negra a la cual ingresará un valor de incógnita y saldrá un valor de un función cuya raíz debe resolverse.

Partimos de un valor estimado de TC_S que denominaremos TC_S^*.

$$
TC_S^{*} \rightarrow \boxed{\begin{array}{c} Q = Fc \times \overline{Cp}_C \times \left(TC_E - TC_S^{*}\right) \\ TF_S = \dfrac{Q + FF \times \overline{Cp}_F \times TF_E}{FF \times \overline{Cp}_F} \\ \mathbf{\Delta Tln} = \dfrac{Q}{(UA)} \\ f = \mathbf{\Delta Tln} - \dfrac{\left(TC_E - TF_S\right) - \left(TC_S^{*} - TF_E\right)}{\mathbf{ln}\dfrac{\left(TC_E - TF_S\right)}{\left(TC_S^{*} - TF_E\right)}} \end{array}} \rightarrow f\left(TC_S^{*}\right)
$$

Vamos a resolver el mismo problema con diversas herramientas infomáticas. En primer lugar con EMSO (Environment for Modeling, Simulation, and Optimization).

EMSO Simulator es un simulador de procesos orientado a ecuaciones con una interfaz gráfica para modelar complejos procesos dinámicos o de estado estacionario. Es compatible con el estándar CAPE-OPEN. EMSO significa entorno para modelado, simulación y optimización. El mismo dispone de una librería propia de módulos, métodos de resolución y paquetes fisicoquímicos. No obstante es interesante en la implementación de modelos creados por el usuario basado en sus propias hipótesis y limitaciones. Los módulos incorporados son abiertos pudiéndose copiar y modificar a voluntad.

Sin entrar en mayores detalles que pueden consultarse en la propia documentación del software un archivo de simulación EMSO (extensión mso) contiene varias secciones:

"using" permite vincular ciertas y determinadas librería para que estén disponibles para el caso particular. La librería "types" es de carácter general y casi obligatoria.

FlowSheet: nombre_del_caso, acá va el nombre del caso
PARAMETERS: aquí se listan los parámetros, es decir los nombres de las constantes a emplear
VARIABLES: aquí se indican las variables (valores a calcular)
EQUATIONS aquí se arma el modelo a través de las ecuaciones que lo representan. A fin de que el problema tenga una solución es que debe haber el mismo número de ecuaciones (independientes) que variables definidas en la sección anterior.
SET: aquí se dan valores a todos los parámetros ya indicados en la sección correspondiente.
INITIAL: esta sección se reserva para los valores iniciales de las variables diferenciales cuando el modelo es dinámico.
GUESS: permite asignar valores estimados a las incógnitas. En ocasiones esto facilita la resolución. Se aplica cuando el modelo no es dinámico.
OPTIONS: para opciones generales como elegir el método de resolución, número de iteraciones, precisión y, cuando es dinámico, valores de inicio, final, paso de integración y unidad temporal elegida.

Una ventaja de este simulador es que las ecuaciones no necesitan estar explicitadas dando flexibilidad al modelo. Por ejemplo, en ocasiones es conveniente que una variable que se encuentra en el denominador (que durante la resolución puede adquirir valor nulo), se pase al otro miembro multiplicando. Algunas veces los logaritmos y raíces pueden sustituirse por sus recíprocas evitando evaluar valores negativos (valores que son temporales durante la resolución). En particular, para el caso del intercambiador, y a fin de eliminar el término logaritmo se creó una variable llamada factor. El modelo resultó:

Modelo EMSO:

```
#* Integración III *#

using "types";

FlowSheet Int_de_calor

PARAMETERS
Cpc as Real;
Fc as Real;
Tce as Real;
Cpf as Real;
Ff as Real;
Tfe as Real;
UA as positive;

VARIABLES
Q as Real;
Tcs as Real;
Tfs as Real;
dtln as Real;
factor as Real;

EQUATIONS
Q=Cpc*Fc*(Tce-Tcs);
Q=Cpf*Ff*(Tfs-Tfe);
dtln=((Tce-Tfs)-(Tcs-Tfe))/factor;
Q=UA*dtln;
exp(factor)*(Tcs-Tfe)=(Tce-Tfs);

SET
Cpc= 1.4;
Fc=100;
Tce=100;
Cpf=1.8;
Ff=150;
Tfe=20;
UA=120;

INITIAL

GUESS
Tcs=60;
Tfs=40;
dtln=10;
Q=1000;

OPTIONS
```

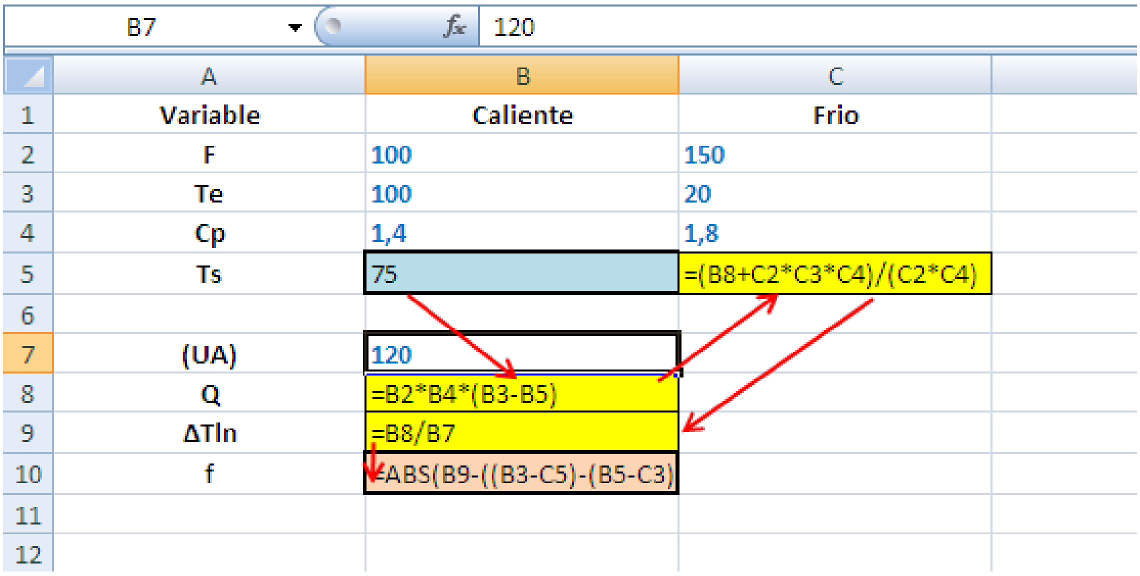
```
TimeStart=10;
TimeEnd=100;
TimeStep=1;
TimeUnit='h';
DAESolver(File="dassl");
Dynamic=false;

end
```

Una vez escrito el código y almacenado en la Pc se lo ejecuta. Los resultados son:

Q = 5765,95
Tcs = 58,8146
Tfs = 41,3554
Dtln = 48,0496
Factor = 0.4126980

Modelo Excel

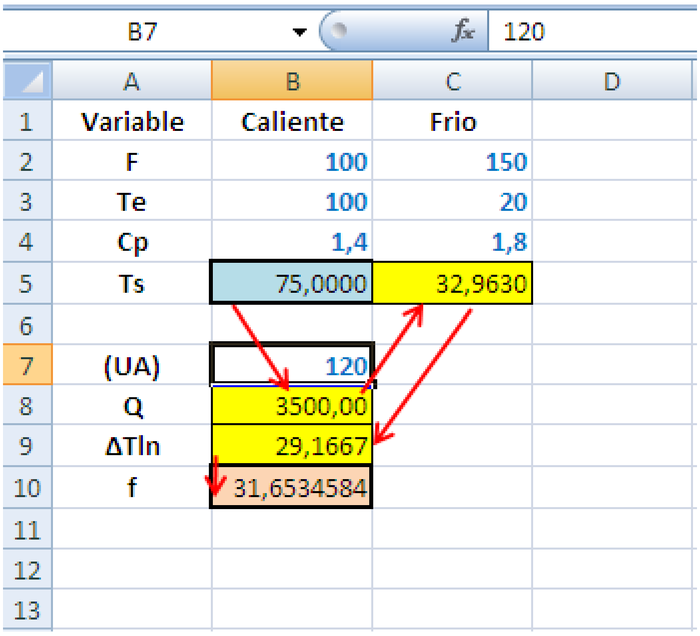
Las planillas Excel son ideales para este tipo de problemas. Para ello armamos una planilla como la siguiente:



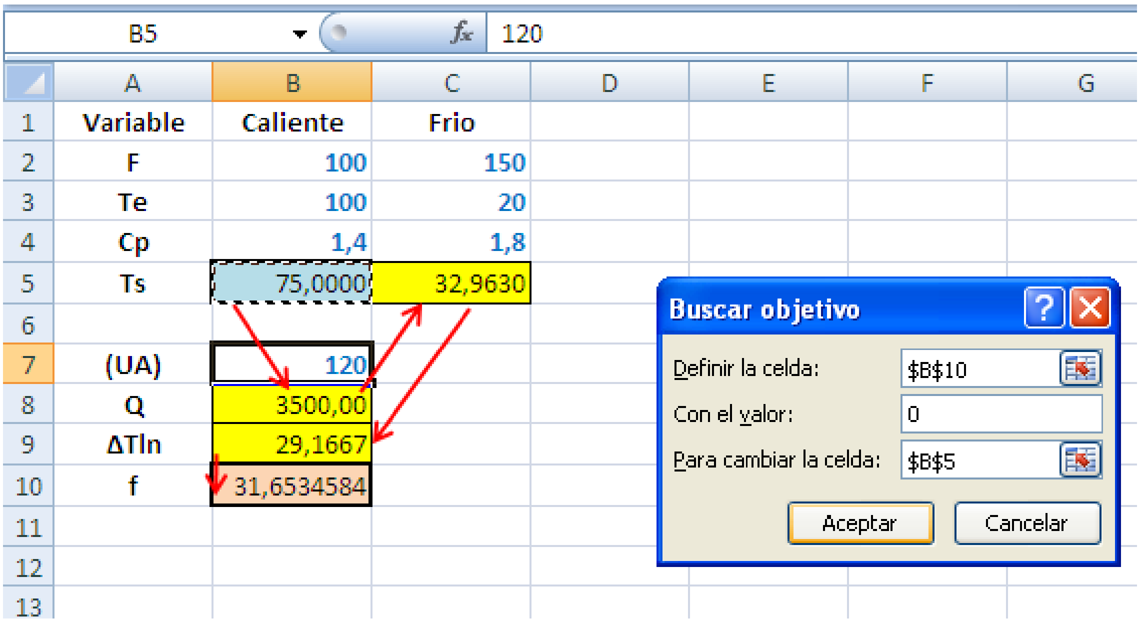
B7 | f_x 120

	A	B	C
1	Variable	Caliente	Frio
2	F	100	150
3	Te	100	20
4	Cp	1,4	1,8
5	Ts	75	=(B8+C2*C3*C4)/(C2*C4)
6			
7	(UA)	120	
8	Q	=B2*B4*(B3-B5)	
9	ΔTln	=B8/B7	
10	f	=ABS(B9-((B3-C5)-(B5-C3))	
11			
12			

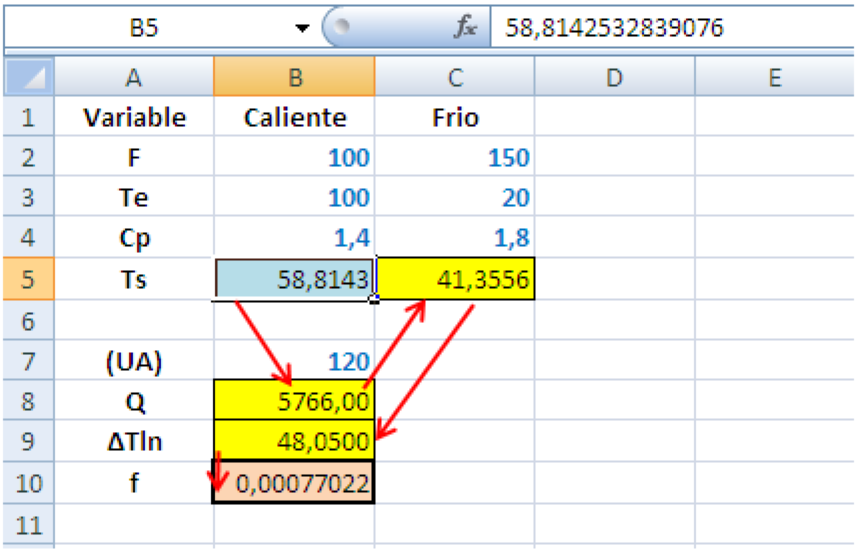
En donde se aprecian que las ecuaciones son las presentadas en la solución algebraica del problema. Así, las flechas indican la secuencia de cálculo que comienza con un valor estimado de Tcs (celda B5=75) y finaliza en la F de la celda B10 cuyo cero se desea encontrar. La planilla resuelta queda:



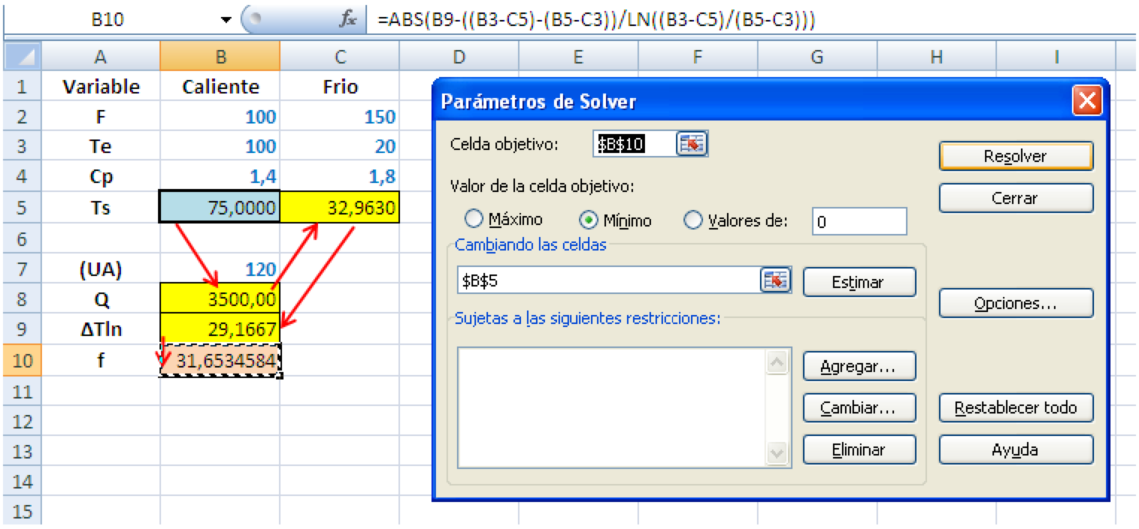
Ya de antemano se ve que 75 ºC no es la solución del problema ya que la f=31,65 ≠ 0. Si bien es posible manipular a mano el valor de la celda B5 hasta hallar la solución, Excel nos ofrece 3 alternativas posibles. La primera es "Busqueda de Objetivos" mediante la cual la herramienta manipula una celda hasta hacer que otra (dependiente de aquella) alcance cierto valor. En nuestro modelo queda:



El cual se resuelve haciendo clic en el botón "Aceptar". Puede ocurrir que no se alcance la solución en un primer intento debido a que el valor actual está lejos de la solución con lo que el método diverge. En ese caso modificamos el valor de la celda B5 por otro valor, por ejemplo 60 ºC. Resolviendo de nuevo nos queda:



El segundo método consiste en utilizar la herramienta Solver que se usa principalmente para la optimización, esto es, hallar una solución mínima o máxima de interés. Alternativamente se puede hallar un valor deseado. La ventaja del SOLVER es que se puede configurar en su precisión además de agregar restricciones.



Y resuelto:

B10 =ABS(B9-((B3-C5)-(B5-C3))/LN((B3-C5)/(B5-C3)))

	A	B	C	D	E	F	G
1	Variable	Caliente	Frio				
2	F	100	150				
3	Te	100	20				
4	Cp	1,4	1,8				
5	Ts	58,8146	41,3554				
6							
7	(UA)	120					
8	Q	5765,95					
9	ΔTln	48,0496					
10	f	4,9166E-05					
11							

Valores más exactos que los anteriores. Pero como se vio en el primer caso, el usuario no puede controlar lo que la herramienta hace por lo que puede implementar sus propios métodos. Así, aprovechamos el modelo hecho copiándolo a otra página del libro de Excel y le haremos algunos cambios. En primer lugar aplicaremos el **Método de sustitución directa** por lo que reemplazaremos la f(x) por otra que llamaremos F(x). La función F(x) reemplaza a la función f(x) cuya raíz se desea encontrar. Para ello se suma TCs a ambos miembros por lo que la igualdad no se altera.

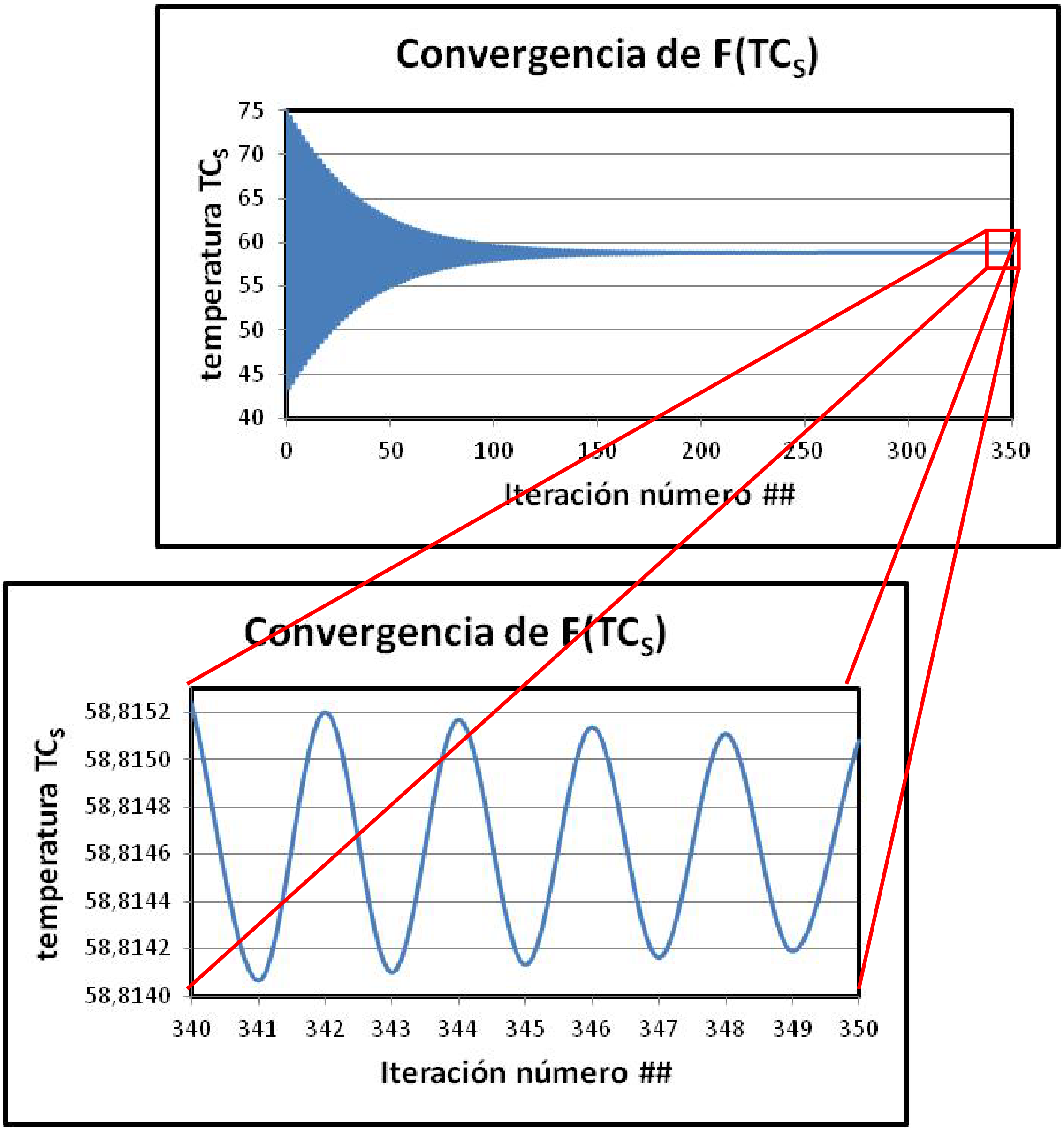
$$TC_S = f(TC_S) + TC_S = F(TC_S)$$

Como es obvio, para la raíz TC_S*, f(TC_S*)=0 por lo que la función anterior se aproxima a TC_S*. Dicho esto veremos cómo queda conformada la planilla de acuerdo a la figura.

G2

	A	B	C	D	E	F	G	H
1	Variable	Caliente	Frio					
2	F	100	150					
3	Te	100	20					
4	Cp	1,4	1,8					
5								
6								
7	(UA)	120						
8								
9								
10								
11								
12	Iteración	TCs	Q	ΔTln	TFS	f	F	
13	0	75,0000	3500,00	29,1667	32,9630	-31,6534584	43,3465	abs(B1-B0)-1e-3
14	1	43,3465	7931,48418	66,0957015	49,3758673	30,85240374	74,1989	**31,6525**

En la celda B13 se inicializa la variable TC_S, en C13 se calcula Q, en D13 ΔTln, en E13 la variable TF_S, en la F13 la f(TC_S) y finalmente en la celda G13 la F(TC_S). Adicionalmente en la H13 se compara el error absoluto con un valor tomado como tolerancia. En este caso 0,001. El valor de la celda B14 se obtiene por igualación en el de F13 y en el resto de las celdas se repiten las fórmulas. Por el algoritmo propuesto (**sustitución directa**) el valor de la columna G debería tender al de la columna B. Cabe aclarar que los parámetros en las ecuaciones deberán ser referenciados en forma absoluta. Copiando la fila 14 hacia abajo se llegará a la solución. En este caso se aprecia una oscilación de la solución convergiendo muy lentamente a la misma. En la figura se aprecia esto en forma gráfica.



La segunda figura muestra la convergencia de las últimas iteraciones siendo necesarias 347 de ellas para alcanzar la solución con la tolerancia escogida.

Las soluciones resultantes son:

Q = 5766,01
Tcs = 48,0501
Tfs = 41,3556
Dtln = 48,0501
It = 347

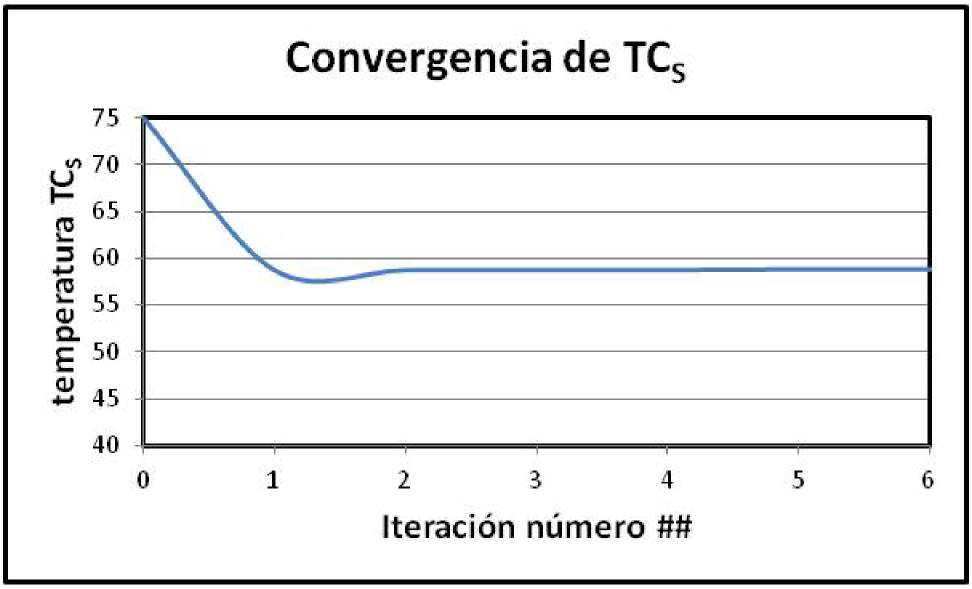
En otra hoja del mismo libro repetiremos el modelo pero utilizando el **Método de Newton-Rahpson**, para lo que haremos uso de la f(TC_S) en lugar de la F(TC_S) y tendremos que calcular la derivada.

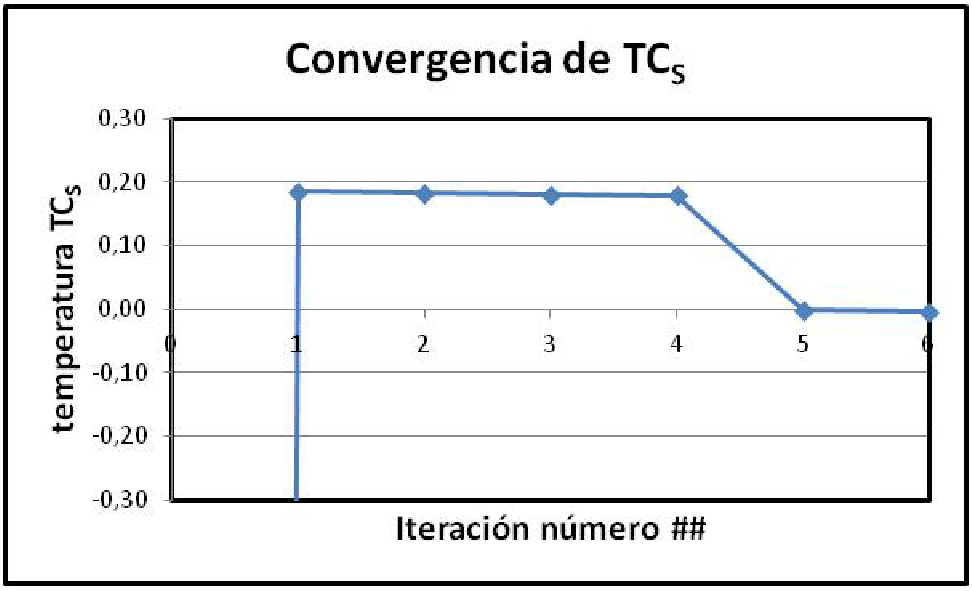
En la figura siguiente se aprecia el modelo y se advierte que a pesar que cada iteración requiere duplicar los cálculos (para la función incrementada y la sin incrementar a fin de calcular la derivada numérica) la solución se alcanza con muy pocas iteraciones.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Variable	Caliente	Frio										
2	F	100	150										
3	Te	100	20										
4	Cp	1,4	1,8										
5													
6													
7	(UA)	120											
8	ΔT	1,00E-03											
9													
10													
11													
12													
13	Iter	TCs	Q	TFS	ΔTln	f	TCs +ΔT	Q	TFS	ΔTln	f	der	
14	0	75,0000	3500,00	32,9630	29,1667	-31,653458	7,50E+01	3499,86	32,9624	29,1655	-31,655403	-1,94E+00	
15	1	58,7200	5779,20	41,40	48,16	282,11	58,72	5779,06	41,40	48,16	0,18	-281921,93	16,2790
16	2	58,7210	5779,05	41,40	48,16	289,84	58,72	5778,91	41,40	48,16	0,18	-289657,65	0,0000
17	3	58,7220	5778,91	41,40	48,16	289,85	58,72	5778,77	41,40	48,16	0,18	-289669,52	0,0000
18	4	58,7230	5778,77	41,40	48,16	0,18	58,72	5778,63	41,40	48,16	0,18	-1,9703735	0,0000
19	5	58,8146	5765,95	41,36	48,05	290,77	58,82	5765,81	41,35	48,05	0,00	-290768,72	0,0906
20	6	58,8156	5765,81	41,35	48,05	290,78	58,82	5765,67	41,35	48,05	0,00	-290783,07	-0,0000

Las soluciones resultantes son:

Q = 5765,81
Tcs = 58,8156
Tfs = 41,3549
Dtln = 48,0484
It = 6





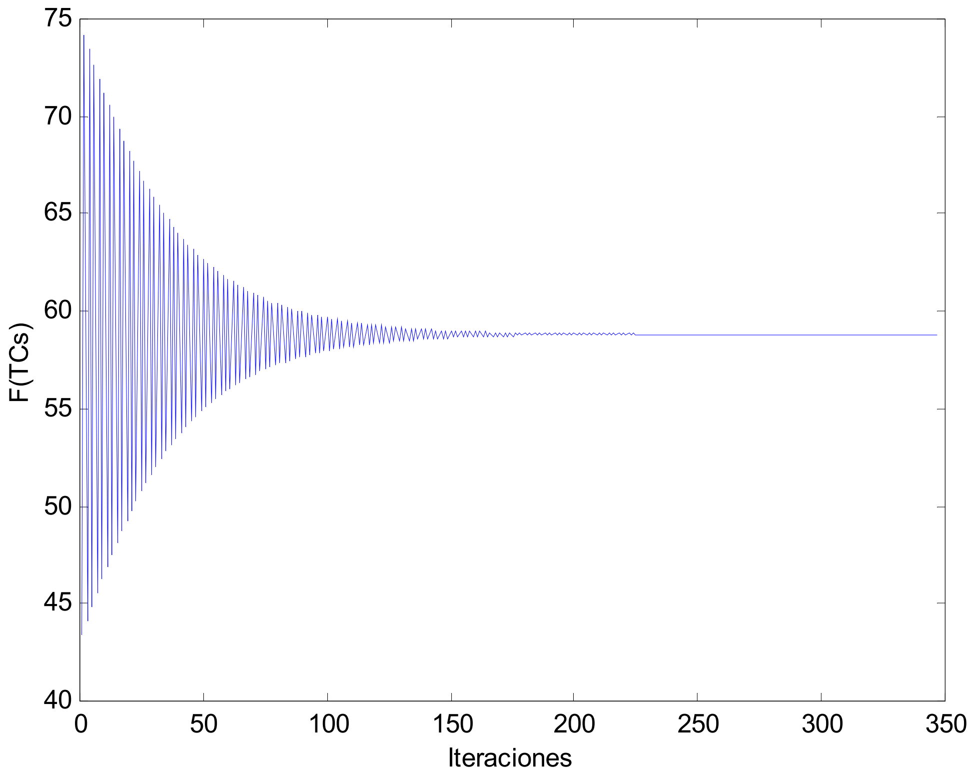
Implementación con MATLAB.

A continuación un ejemplo de código MATLAB para la F(TC_S) y el **método de sustitución directa** que hallará la solución (ResolSD).

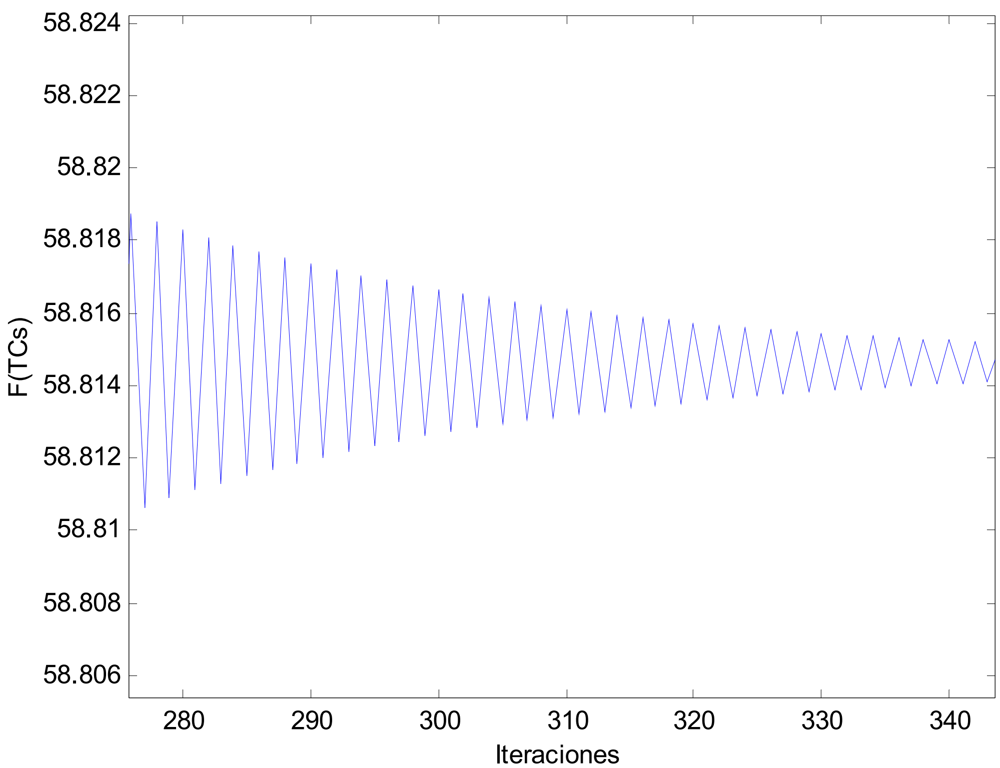
```
function F=F(TCS);
FC=100;
CPC=1.4;
TCE=100;
FF=150;
CPF=1.8;
TFE=20;
UA=120;
Q=FC*CPC*(TCE-TCS)
TFS=(Q+FF*CPF*TFE)/(FF*CPF)
DTLN=Q/UA
F=DTLN-((TCE-TFS)-(TCS-TFE))/log((TCE-TFS)/(TCS-TFE))+TCS
end


function ResolSD=ResolSD()
TCS=75;
it=0;
yerro=1;
while yerro>1e-3;
   T1=F(TCS);
   yerro=abs(T1-TCS);
   TCS=T1;
   it=it+1
end
```

Las soluciones resultantes son:

Q = 5765,9
Tcs = 58,8142
Tfs = 41,3551
Dtln = 48,0490
It = 347



Haciendo zoom en las últimas iteraciones:



Al resolverlo por el **Método de Newton-Rahson** se implementan los siguientes códigos:

```
function f0=f0(TCS);
FC=100;
CPC=1.4;
TCE=100;
FF=150;
CPF=1.8;
TFE=20;
UA=120;
Q=FC*CPC*(TCE-TCS)
TFS=(Q+FF*CPF*TFE)/(FF*CPF)
DTLN=Q/UA
f0=DTLN-((TCE-TFS)-(TCS-TFE))/log((TCE-TFS)/(TCS-TFE))
end
```

```
function ResolNR=ResolNR()
TCS=75;
it=0;
yerro=1;
while yerro>1e-3;
   der=(f0(TCS+1e-3)-f0(TCS))/1e-3
   T1=TCS-f0(TCS)/der;
   yerro=abs(T1-TCS);
   TCS=T1
   it=it+1
end
```

Las soluciones resultantes son:

Q = 5766,0
Tcs = 58,8146
Tfs = 41,3554
Dtln = 48,0496
It = 3

O a través del panel de comandos:

[TFS,Q,TCS,DTLN]=solve ('Q=100*1.4*(100-TCS)', 'Q=150*1.8*(TFS-20)','Q=DTLN*120','DTLN=((100-TFS)-(TCS-20))/log((100-TFS)/(TCS-20))')

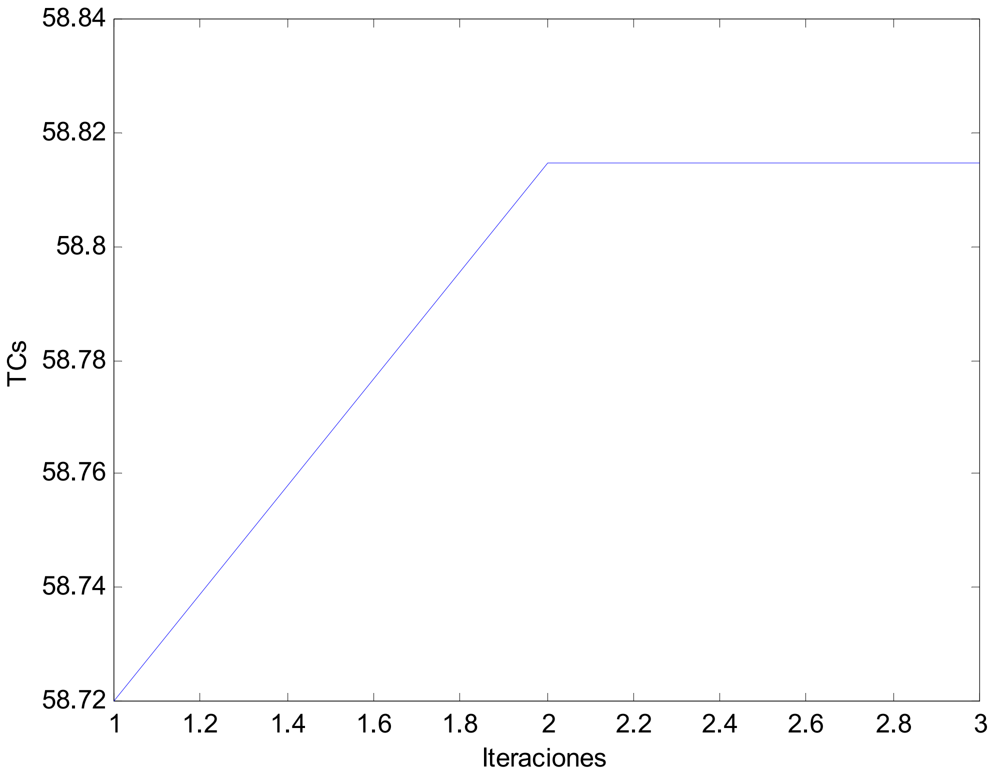
Lo que da:

TFS = 48.049581738522020015549713539263

Q = 5765.9498086226424018659656247116

TCS =58.814644224123982843814531252060

DTLN =41.355369661565342229133206017450


58.84
58.82
58.8
58.78
58.76
58.74
58.72
TCs
1
1.2
1.4
1.6
1.8
2
2.2
2.4
2.6
2.8
3
Iteraciones

Implementación con GAMS.

GAMS es un software para la optimización de modelos implementados por usuario. El mismo implica la búsqueda de máximo o mínimo de una determinada función a través de variables, parámetros y funciones que permite establecer la relación entre ellos (modelo) como así de condiciones limitantes de las variables (restricciones). En un modelo como el del intercambiador no existe en realidad una variable a optimizar pero aún así puede establecerse una variable muda con un valor constante. De este modo el resultado final es, ya no la optimización de la variable muda sino la resolución de las ecuaciones del sistema, esto es, el modelo a resolver. A continuación el código implementado en cuestión:

Cabe mencionar la necesidad que hubo de limitar el valor de la variable TCS a la cual se le asignó un límite inferior de 40 no siendo necesario hacerlo en las otras variables (como se muestra en la salida del simulador). A continuación del código se listan las ecuaciones y variables tal como las presenta GAMS. Dichos resultados son consistentes con los anteriores.

```
* Integración III
* Intercambiador de calor

VARIABLES
Q,muda,TFS,dtln,TCS;

PARAMETERS
FC /100/
CPC /1.4/
TCE /100/
FF /150/
CPF /1.8/
TFE /20/
UA /120/


EQUATIONS
R0,R1,R2,R3,R4;

R0.. muda =E= 3;
R1.. Q =E= CPC*FC*(TCE-TCS);
R2.. Q =E= CPF*FF*(TFS-TFE);
R3.. dtln =E= ((TCE-TFS)-(TCS-TFE))/log((TCE-TFS)/(TCS-TFE));
R4.. Q =E= UA*dtln;

TCS.lo = 40;

MODEL BM /ALL/;
SOLVE BM USING NLP MINIMIZING muda;
```

Resultados:

	LOWER	LEVEL	UPPER	MARGINAL
---- EQU R0	3.000	3.000	3.000	1.000
---- EQU R1	14000.000	14000.000	14000.000	EPS
---- EQU R2	-5400.000	-5400.000	-5400.000	EPS
---- EQU R3	.	.	.	EPS
---- EQU R4	.	.	.	EPS

	LOWER	LEVEL	UPPER	MARGINAL
---- VAR Q	-INF	**5765.950**	+INF	.
---- VAR muda	-INF	**3.000**	+INF	.
---- VAR TFS	-INF	**41.355**	+INF	.
---- VAR dtln	-INF	**48.050**	+INF	.
---- VAR TCS	***40.000***	**58.815**	+INF	.