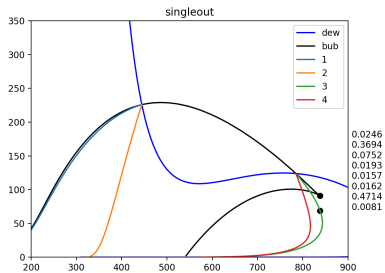


Casos de estudio: Envolventes incompletas

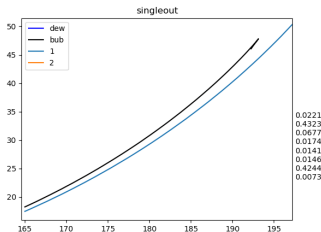
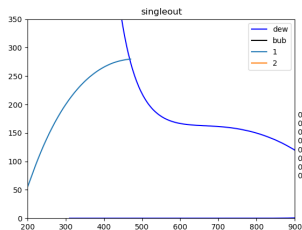
## Caso 1

# Caso 1



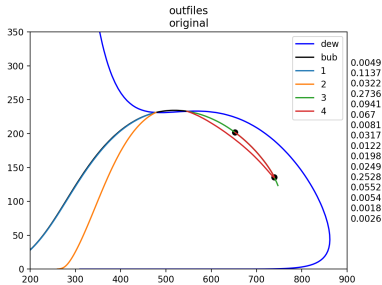
# Envolvente arrepentida

Una envolvente de burbuja se arrepiente de su recorrido por lo que el algoritmo asume un equivocado caso 3



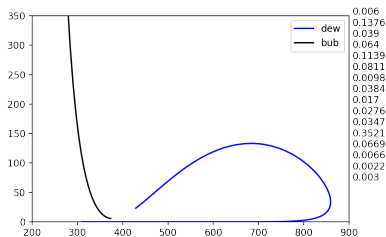
## Caso 2

## Caso 2



De aqui se tomaron dos casos particulares

# Curva de rocío igual a curva de alta presión LL



Se puede observar que se calcularon las dos líneas cuando en realidad debería de ser todo una misma.

Habría que ver de hacer más estricto el corte de la envolvente de rocío, cosa de que recorra un poco más antes de que corte la línea

## Curva de rocío igual a curva de alta presión LL

```
(choice == 2 .and. i > 1) then  
: TODO: If this is the way the low p dew line finishes, I think this should be more strict  
if (P < Pv(i - 1) .and. P < maxP/5 .and. T > 300) then  
  run = .false. ! to finish envelope going to low T bubble  
end if
```

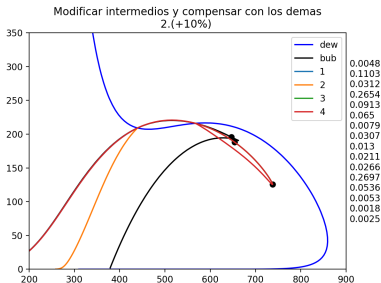
Además de esto, nunca se le asigna un valor a la variable PlowT (inicializadora de curva trifásica), lo que no permite que se calculen las líneas trifásicas. Esto tiene fácil solución ahora que en el código actualizado se guardan todos los puntos, así que se podría simplemente extraerlo de ahí.

En este caso la curva de alta presión termina arriba de 1 bar (~5bar) por lo que el algoritmo no lo considera en ningún caso, no se hace el cálculo de curva de burbuja necesario para luego buscar cruces e inicializar trifásicas.



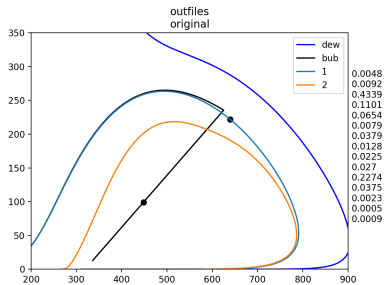
# Envolvente ambiciosa

Una envolvente trifásica que recorre más de lo que debería, inclusive pasa por la línea de equilibrio bifásico.



## Caso 3

# Caso 3

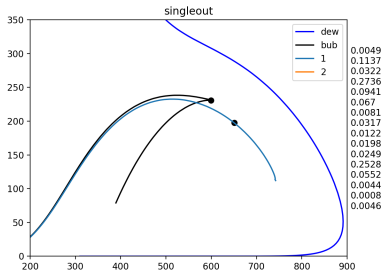


## Curva trifásica no inicializa

En este caso la segunda curva trifásica (LAOP) no puede inicializarse.

Más allá de la corrección de la inicialización, creo que sería importante tener en cuenta si llegamos a considerar que el comportamiento de tipo cuatro debería de calcularse de su propia manera particular, ya que posee una línea entera continua en lugar de ser dos líneas separadas, hubo algunos casos en los cuales la línea inferior continuaba sobre la superior hasta su origen.

# Curva trifásica no inicializa



## Curva trifásica no inicializa

En una variante del *caso 1* (en el que tampoco se inicializa la segunda curva trifásica), se vio que un cálculo Flash devuelve una composición mayor a uno en la fase w.

# Curva trifásica no inicializa

```
Wstep = WCN * EXP(0.01*(10000*(1-100)/P) / P) * 0.000010 #10000 #10000 #10000
xx = Z/(1-beta+beta*Wstep)
s = sum(xx)
end do
NFAC(n) = 1/xx(n)
P = P1 + (P2-P1) * (1-S1) / (S2-S1)
NFAC1 = NFAC(1:n) ! Item converged first po
P = P1*0.7
FIRST = .true.
spec = 'TP'
call flash(spec, FIRST, smodel, n, z, tcn, pem, om
             Kis,or,KHn,Tstain,lijo,t,p,v,xx,W,tho_x,tho_y,beta,iter)
call TERM(p,1,1,T,P,xx,Vx,PHILOx,DLPHIPy,DLPHITy,FUGNy)
```

En ese Flash, al tratar de mantener a beta dentro del rango de betamin y betamax, se llega a un step igual a cero.

# Curva trifásica no inicializa

```
125 | beta = beta + step
126 |
127 | if (beta < 0) beta = beta + step
128 | while (beta < min(re_beta, 0.0) || re_step < re_0.0) { next_beta (GUARANTEED) 3/3/23
129 |   step = step/2
130 |   beta = beta - step
131 | }
```

Con el ajuste, se llega a que la fase de asfaltenos no es la mayoritaria en los flashes (no estoy seguro si es relevante)



# Curva trifásica no inicializa

Pold = aux	[0]	
end do	[1]: 0.023200707451397521	
if (abs(dif) > 0.1) then	[2]: 0.01450460047024350449	
FIRST = .true.	[3]: 0.10120314514122791	
Told = T	[4]: 0.030648185614653135	
T = T + 10.0	[5]: 0.025041120720404490	
end if	[6]: 0.025047956792396203	
do while (abs(dif) > 0.1)	[7]: 0.75224455212036878	
call flash(spec, FIRST, nr	[8]: 0.026008243023856041	
K1j_or_K0n, Tst	Hold Alt key to switch to editor language browser	deltain, &
call TERMO(n, 1, 1, T, P, xx, Vx, PHILOGx, DLPHIPy, DLPHITy, FUGNy)		(, iter)
call TERMO(n, 1, 1, T, P, y, Vy, PHILOGy, DLPHIPy, DLPHITy, FUGNy)		
dif = T - Told		

La envolvente trifásica sigue sin converger, en el cálculo del primer punto pareciese que converge sólo una fase.

# Curva trifásica no inicializa

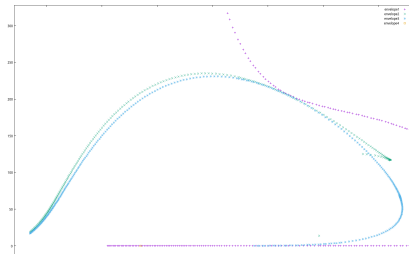
```
delS = 0.001
nd if
x = z/(1 - beta + beta*KF
= KFAct*xx
= KFsep*xx
pId = 0.d0
FdS = 0.d0
FdS(2*n + 3) = -1.d0
rite (2, *) ' T
f (Comp3ph) write (3, *)

while (run)
i = i + 1 ! number of
! Newton starts here
delX = 1.0
iter = 0
max_iter = 100
reps = 0

do while (maxval(abs(delX)) > 1.d-5 .and. iter <= max_iter)
iter = iter + 1
```

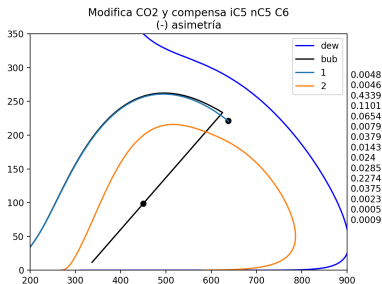
```
[1]: 6.3284805458512639e-16
[2]: 2.8816046562169099e-15
[3]: 2.7627821429692436e-15
[4]: 7.9845163512832435e-16
[5]: 4.71481876802807e-15
[6]: 3.812961099197277e-15
[7]: 9.9029855389311262e-15
[8]: -4.3688317384408978e-15
[9]: 0.29234788272825836
[10]: 0.33293313578050854
[11]: 0.36862297987182533
[12]: 0.39322880017367955
[13]: 0.4173914293215385
[14]: 0.38893059078213814
[15]: 0.53847688515971949
[16]: -0.006317286961213915
[17]: 0
[18]: -0.18927996529788985
[19]: 0.0066728026702785873
```

Se vio que la inicializacion de la segunda trifasica es desde el punto que normalmente es final de la envolvente (baja P y “alta” T)



# Envolvente trifásica muere en el crítico

Una envolvente trifásica llega a calcular el punto crítico pero no continua para finalizar el recorrido completo



# Envolvente trifasica muere antes de crítico

Una envolvente trifásica no pudo llegar siquiera detectar el punto crítico (línea verde).

