

# Python como herramienta de automatización de cálculo

## Caso aplicado en envolventes trifásicas

Federico Benelli

IPQA

25 de julio de 2022

## 1 Introducción

- Descripción del problema
- Programa original
  - Archivo de entrada
  - Archivo de salida

## 2 Definición de casos de estudio

- Metodología

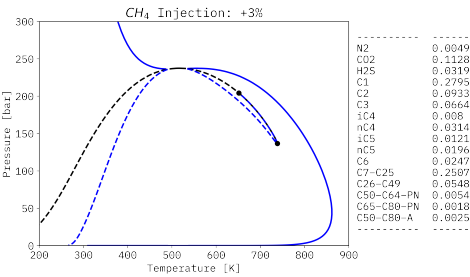
## 3 Resolución en Python

- Lectura de archivos Excel
- Generación de archivos
- Ejecución de programa Fortran
- Generación de gráfico
- Reacomodo de archivos
- Tipos de interacción
- Gráficos finales

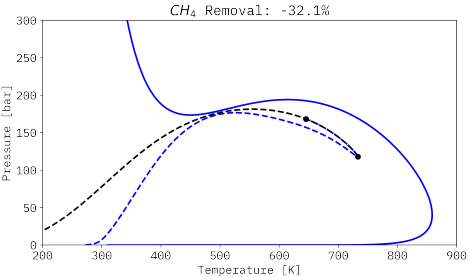
2022-07-25

# Python como herramienta de automatización de cálculo

- Introducción
  - Descripción del problema
  - Programa original
    - Archivo de entrada
    - Archivo de salida
- Definición de casos de estudio
  - Metodología
- Resolución en Python
  - Lectura de archivos Excel
  - Generación de archivos
  - Ejecución de programa Fortran
  - Generación de gráfico
  - Reacomodo de archivos
  - Tipos de interacción
  - Gráficos finales



Figura



Figura

Problema: Necesidad de generar y correr múltiples casos en un tiempo relativamente acotado.

En un congreso reciente se presentó un trabajo consistente en la categorización de casos particulares de sistemas multicomponentes con preñencia de equilibrios trifásicos.

Para poder realizar esta categorización fue necesario realizar múltiples ejecuciones de un programa que calcula las envolventes correspondientes.

# Programa original

Descripción

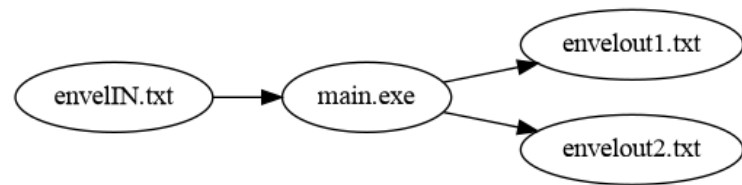


Figura: Funcionamiento básico programa

2022-07-25

## Python como herramienta de automatización de cálculo

└─ Introducción

└─ Programa original

└─ Programa original

Programa original  
Descripción



Figura: Funcionamiento básico programa

Para el cálculo de estas envolventes se utilizó un programa previamente desarrollado, con unas modificaciones básicas para facilitar la automatización del proceso de generación de gráficos.

# Introducción

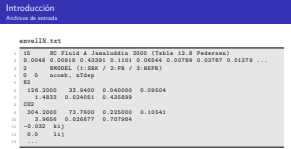
Archivos de entrada

envelIN.txt

```
1 15      NC Fluid A Jamaluddin 2000 (Table 12.8 Pedersen)
2 0.0048 0.00919 0.43391 0.1101 0.06544 0.00789 0.03787 0.01279 ...
3 2      NMODEL (1:SRK / 2:PR / 3:RKPR)
4 0 0    ncomb, nTdep
5 N2
6 126.2000 33.9400 0.040000 0.09504
7 1.4833 0.024051 0.435899
8 C02
9 304.2000 73.7600 0.225000 0.10541
10 3.9656 0.026677 0.707984
11 -0.032 kij
12 0.0 lij
13 ...
```

# Python como herramienta de automatización de cálculo

2022-07-25



El programa que realiza los cálculos utiliza como input archivos de texto simple con una estructura predefinida. El archivo debe llamarse envelin y encontrarse en la misma carpeta que el ejecutable.

envelout3.txt				
1	T(K)	P(bar)	D(mol/L)	
2	310.0000	0.1529E-23	0.5930E-25	
3	311.5589	0.2517E-23	0.9715E-25	
4	313.7960	0.5098E-23	0.1954E-24	
5	316.0554	0.1028E-22	0.3912E-24	
6	318.3371	0.2065E-22	0.7801E-24	
7	...			
8				
9	Number of critical points found:			1
10	T(K)	P(bar)	D(mol/L)	
11	639.1631	0.2328E+03	0.4516E+01	225

Tras la ejecución, el programa genera un archivo de salida por cada envoltente calculada. Llamados envelout<N>.txt

envelout3.txt			
1	T(K)	P(bar)	D(mol/L)
2	310.0000	0.1529E-23	0.5930E-25
3	311.5589	0.2517E-23	0.9715E-25
4	313.7960	0.5098E-23	0.1954E-24
5	316.0554	0.1028E-22	0.3912E-24
6	318.3371	0.2065E-22	0.7801E-24
7	...		
8			
9	Number of critical points found:		
10	T(K)	P(bar)	D(mol/L)
11	639.1631	0.2328E+03	0.4516E+01 225

# Definición de casos de estudio

Se partió de los tres casos de estudio originales, a partir de los cuales se generaron variantes.

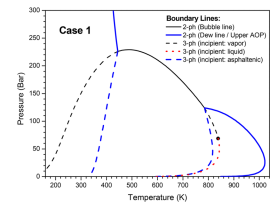


Figura: Caso 1

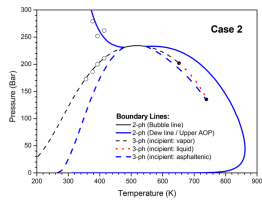


Figura: Caso 2

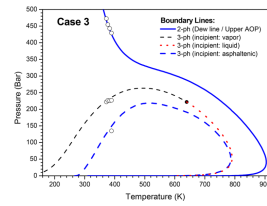


Figura: Caso 3

2022-07-25

## Python como herramienta de automatización de cálculo

Definición de casos de estudio

Definición de casos de estudio

Se partió de los tres casos de estudio originales, a partir de los cuales se generaron variantes.



Figura: Caso 1



Figura: Caso 2



Figura: Caso 3

# Metodología

Definir rangos!

16		Modificar simetría del sistema									
	N2	CO2	H2S	C1	C2	C3	iC4	nC4	iC5	nC5	
Original	0.0049	0.11369	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.0198	
1.(-) asimetría	0.0049	0.045476	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.053907	
2.(-) asimetría	0.0049	0.056845	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.0482225	
3.(+) asimetría	0.0049	0.1421125	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.00558875	0
	0.056845	0.0284225	0.068214								
	0.0284225	0.01421125	0.034107								
16		Modifica CH4 y compensa iC5 nC5; C6									
	N2	CO2	H2S	C1	C2	C3	iC4	nC4	iC5	nC5	
Original	0.0049	0.11369	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.0198	
1.(-) asimetría	0.0049	0.11369	0.0322	0.18238	0.09409	0.06699	0.0081	0.0317	0.04259667	0.05019667	0
2.(-) asimetría	0.0049	0.11369	0.0322	0.12158667	0.09409	0.06699	0.0081	0.0317	0.06286111	0.07046111	0
3.(-) asimetría	0.0049	0.11369	0.0322	0.08105778	0.09409	0.06699	0.0081	0.0317	0.07637074	0.08397074	0
4.(-) asimetría	0.0049	0.11369	0.0322	0.05403852	0.09409	0.06699	0.0081	0.0317	0.08537716	0.09297716	0
4.(-) asimetría	0.0049	0.11369	0.0322	0.03602568	0.09409	0.06699	0.0081	0.0317	0.09138144	0.09898144	0
5.(+) asimetría	0.0049	0.11369	0.0322	0.02401712	0.09409	0.06699	0.0081	0.0317	0.09538429	0.10298429	0
6.(+) asimetría	0.0049	0.11369	0.0322	0.01601141	0.09409	0.06699	0.0081	0.0317	0.09805286	0.10565286	0
7.(+) asimetría	0.0049	0.11369	0.0322	0.01067428	0.09409	0.06699	0.0081	0.0317	0.09983191	0.10743191	0
8.(+) asimetría	0.0049	0.11369	0.0322	0.00711618	0.09409	0.06699	0.0081	0.0317	0.10101794	0.10861794	0
9.(+) asimetría	0.0049	0.11369	0.0322	0.00474412	0.09409	0.06699	0.0081	0.0317	0.10180863	0.10940863	0
10.(+) asimetría	0.0049	0.11369	0.0322	0.00316275	0.09409	0.06699	0.0081	0.0317	0.10233575	0.10993575	0
11.(+) asimetría	0.0049	0.11369	0.0322	0.0021085	0.09409	0.06699	0.0081	0.0317	0.10268717	0.11028717	0

Figura: Archivo Excel

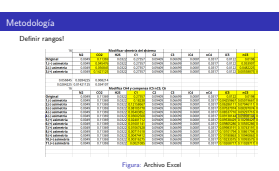
2022-07-25

Python como herramienta de automatización de cálculo

Definición de casos de estudio

Metodología

Metodología



Fue necesario definir un gran número de casos de estudio, con el fin de barrer la mayor cantidad posibles combinaciones (lógicas).

Para esto se generaron archivos Excel con los rangos de concentraciones a estudiar. (Captura Excel)



# Resolución en Python

## Introducción

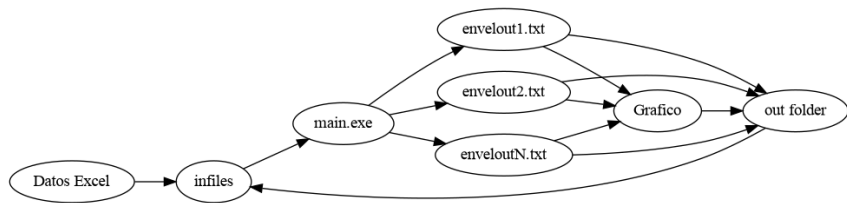


Figura: Esquema general

2022-07-25

Python como herramienta de automatización de cálculo  
└ Resolución en Python  
└ Resolución en Python



Figura: Esquema general

Se desarrolló un pequeño script en Python para automatizar todo el proceso de generación de gráficos para su posterior análisis.

## Lectura de datos en Excel

*Estructura de tabla:*

- Número de compuestos.
- Título.
- Etiqueta de fila.
- Concentraciones.

16	Modificator simetriei del sistem									
	N2	CO2	H2S	C1	C2	C3	IC4	nC4	ICS	nC5
Original	0.0049	0.11369	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.0198
1.-) asimetria	0.0049	0.045476	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.053907
2.-) asimetria	0.0049	0.056845	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.0482225
3.-) asimetria	0.0049	0.1421125	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.00555875
	0.056845	0.0284225	0.068214							
	0.0284225	0.0411215	0.034017							
16	Modificator CH4 y compensa ICS nC5: C6									
	N2	CO2	H2S	C1	C2	C3	IC4	nC4	ICS	nC5
Original	0.0049	0.11369	0.0322	0.27357	0.09409	0.06699	0.0081	0.0317	0.0122	0.0198
1.-) asimetria	0.0049	0.11369	0.0322	0.18238	0.09409	0.06699	0.0081	0.0317	0.04259657	0.0019667
2.-) asimetria	0.0049	0.11369	0.0322	0.1215867	0.09409	0.06699	0.0081	0.0317	0.06268111	0.07046111
3.-) asimetria	0.0049	0.11369	0.0322	0.0810578	0.09409	0.06699	0.0081	0.0317	0.07637074	0.08397074
4.-) asimetria	0.0049	0.11369	0.0322	0.04303852	0.09409	0.06699	0.0081	0.0317	0.08537316	0.09297316
5.-) asimetria	0.0049	0.11369	0.0322	0.0402348	0.09409	0.06699	0.0081	0.0317	0.09138148	0.09893148
5.+ ) asimetria	0.0049	0.11369	0.0322	0.02401712	0.09409	0.06699	0.0081	0.0317	0.09532849	0.10294249
6.-) asimetria	0.0049	0.11369	0.0322	0.01601141	0.09409	0.06699	0.0081	0.0317	0.09805282	0.10565282
7.+ ) asimetria	0.0049	0.11369	0.0322	0.01067428	0.09409	0.06699	0.0081	0.0317	0.09983919	0.10863919
8.-) asimetria	0.0049	0.11369	0.0322	0.00711618	0.09409	0.06699	0.0081	0.0317	0.10171944	0.10871944
9.-) asimetria	0.0049	0.11369	0.0322	0.00422412	0.09409	0.06699	0.0081	0.0317	0.10186681	0.10846681
10.+ ) asimetria	0.0049	0.11369	0.0322	0.00316275	0.09409	0.06699	0.0081	0.0317	0.10233575	0.10993575
11.-) asimetria	0.0049	0.11369	0.0322	0.00021085	0.09409	0.06699	0.0081	0.0317	0.10268717	0.10287717

Figura

2022-07-25

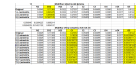
## Python como herramienta de automatización de cálculo

- └ Resolución en Python
  - └ Lectura de archivos Excel
    - └ Resolución en Python

Para la lectura de archivos Excel, se utilizó la librería `openpyxl`, la cual permite la rápida lectura. En cada archivo Excel se iteró por filas, identificando el comienzo de cada tabla, leyendo su título correspondiente.

*Estructura de tabla*

- Estructura de tabla:
- Número de compuestos.
  - Título.
  - Etiqueta de fila.
  - Concentraciones.



Figure

# Resolución con Python

## Generación archivos

Durante la lectura del archivo Excel se realizaron los chequeos y acciones.

- En tabla? -> Crear carpeta outfiles/nombretabla
- Fila con contenido? ->
  - Leer concentraciones
  - Modificar archivo input con las nuevas concentraciones
  - Copiar nuevo archivo input a outfiles/nombretabla/caso.archivonuevo

2022-07-25

## Python como herramienta de automatización de cálculo

### └ Resolución en Python

#### └ Generación de archivos

##### └ Resolución con Python

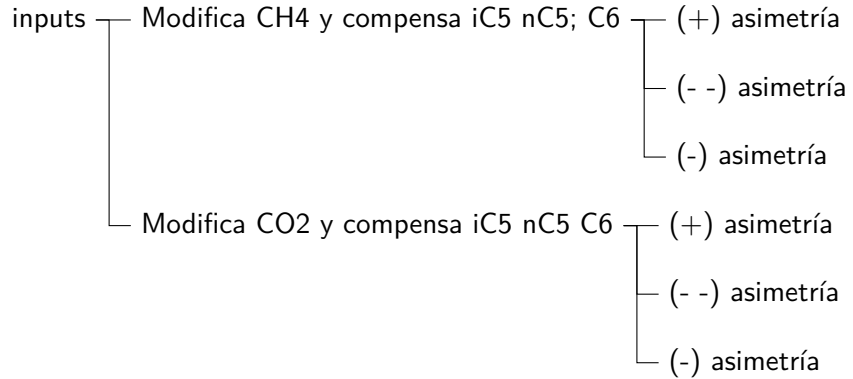
Durante la lectura del archivo Excel se realizaron los chequeos y acciones.

- En tabla? -> Crear carpeta outfiles/nombretabla
- Fila con contenido? ->
  - Leer concentraciones
  - Modificar archivo input con las nuevas concentraciones
  - Copiar nuevo archivo input a outfiles/nombretabla/caso.archivonuevo

# Resolución con Python

Generación de archivos de entrada

Finalizada la lectura de el archivo se obtiene una estructura de archivos de entrada



como:

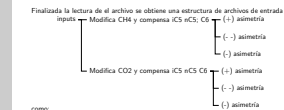
2022-07-25

## Python como herramienta de automatización de cálculo

└ Resolución en Python

└ Generación de archivos

└ Resolución con Python



## Resolución con Python

## Ejecución de compilado Fortran

```

1 def run_fortran(i, out_dir):
2     """Función run_fortran, recibe un número de iteración y
3     un directorio de salida"""
4
5     os.system("./main.exe")
6
7     # Juntar todos los archivos de salida en una lista
8     envelopes = sorted(glob("envelout*"))
9
10    # envelopes = ["envelout1.txt", "envelout2.txt", ...]
11    ...

```

2022-07-25

## Python como herramienta de automatización de cálculo

## └ Resolución en Python

## └ Ejecución de programa Fortran

## └ Resolución con Python

```

1 def run_fortran(i, out_dir):
2     """Función run_fortran, recibe un número de iteración y
3     un directorio de salida"""
4
5     os.system("./main.exe")
6
7     # Juntar todos los archivos de salida en una lista
8     envelopes = sorted(glob("envelout*"))
9
10    # envelopes = ["envelout1.txt", "envelout2.txt", ...]
11    ...

```

Una vez con todos los archivos de salida generados, es necesario ejecutar el programa Fortran. Al ser una aplicación de consola, es un paso sencillo donde se puede ejecutar como un comando de sistema

# Resolución con Python

## Generación de gráfico

```
1  ...
2  # Funcion make_figure -> Recibe la lista de envolventes y
3  # el titulo a darle al grafico
4  make_figure(envelopes, title='\n'.join(out_dir.split("/")[-2:]))
```

2022-07-25

## Python como herramienta de automatización de cálculo

### └ Resolución en Python

#### └ Generación de gráfico

#### └ Resolución con Python

```
'''
# Funcion make_figure -> Recibe la lista de envolventes y
# el titulo a darle al grafico
make_figure(envelopes, title='\n'.join(out_dir.split("/")[-2:]))
'''
```

Una vez con los archivos de salida, se llama a una función que recibe estos nombres y un nombre de gráfico.

# Generación de gráficos

## Función make figure

```

1 def make_figure(envelopes, title):
2     """From a set of envelopes, plot them."""
3     plt.clf()
4
5     for i, file in enumerate(envelopes):
6         ax = plt.subplot()
7         show_crit(file, '', ax)
8         if i == 0:
9             plot_envelope(file, 'dew', ax)
10        elif i == 1:
11            plot_envelope(file, 'bub', ax)
12        elif i > 1:
13            plot_envelope(file, i-1, ax)

```

2022-07-25

## Python como herramienta de automatización de cálculo

### └ Resolución en Python

#### └ Generación de gráfico

#### └ Generación de gráficos

### Generación de gráficos

#### Función make figure

```

1 def make_figure(envelopes, title):
2     """From a set of envelopes, plot them."""
3     plt.clf()
4
5     for i, file in enumerate(envelopes):
6         ax = plt.subplot()
7         show_crit(file, '', ax)
8         if i == 0:
9             plot_envelope(file, 'dew', ax)
10        elif i == 1:
11            plot_envelope(file, 'bub', ax)
12        elif i > 1:
13            plot_envelope(file, i-1, ax)

```

Dentro de la función make figure se itera por cada archivo, para el cual se llama a la función plot envelope, la cual realiza la lectura de los datos en el archivo y traza la línea correspondiente.

# Generación de gráficos

## Función plot envelope

```
1 def plot_envelope(file, label, ax):
2     ts = []
3     ps = []
4     with open(file) as f:
5         for line in f.readlines()[1:]:
6             if line.split() == []:
7                 break
8             x, y = line.split()[1:2]
9             ts.append(float(x))
10            ps.append(float(y))
11
12    ax.plot(ts, ps, label=label)
```

2022-07-25

## Python como herramienta de automatización de cálculo

## └ Resolución en Python

## └ Generación de gráfico

## └ Generación de gráficos

```
1 def plot_envelope(file, label, ax):
2     ts = []
3     ps = []
4     with open(file) as f:
5         for line in f.readlines()[1:]:
6             if line.split() == []:
7                 break
8             x, y = line.split()[1:2]
9             ts.append(float(x))
10            ps.append(float(y))
11
12    ax.plot(ts, ps, label=label)
```



# Resolución con Python

## Racomodo de archivos

```
1  shutil.move('envelopes.png', out_dir)
2  for envel in envelopes:
3      shutil.move(envel, out_dir)
```

2022-07-25

## Python como herramienta de automatización de cálculo

### └ Resolución en Python

#### └ Reacomodo de archivos

#### └ Resolución con Python

```
> shutil.move('envelopes.png', out_dir)
> for envel in envelopes:
>     shutil.move(envel, out_dir)
```

Una vez que se generó el gráfico, se mueven los archivos de salida y el gráfico a la carpeta correspondiente.

# Resolución con Python

Interacción flexible

```
1 - Correr Excel completo:
2 >>> python run_env23.py main.exe full caso.xlsx all
3
4 - Correr Hoja específica:
5 >>> python run_env23.py main.exe full caso.xlsx HOJA
6
7 - Correr envellN.txt directamente:
8 >>> python run_env23.py main.exe single
```

2022-07-25

## Python como herramienta de automatización de cálculo

└ Resolución en Python

└ Tipos de interacción

└ Resolución con Python

```
- Correr Excel completo:
>>> python run_env23.py main.exe full caso.xlsx all
- Correr Hoja específica:
>>> python run_env23.py main.exe full caso.xlsx HOJA
- Correr envellN.txt directamente:
>>> python run_env23.py main.exe single
```

Para facilitar la operación, también se agregaron distintas opciones de ejecución para evitar tener que calcular todo el Excel cada vez.

# Gráficos finales

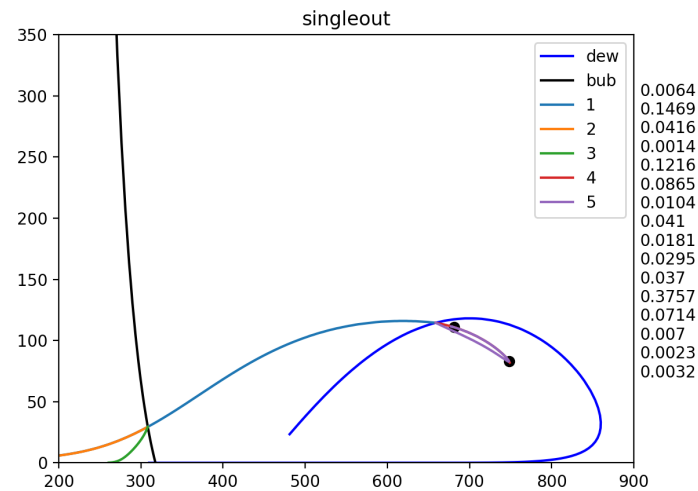


Figura 5. Bubble point vs  $Q^*$

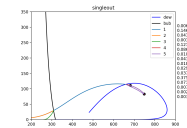
2022-07-25

## Python como herramienta de automatización de cálculo

└ Resolución en Python

└ Gráficos finales

└ Gráficos finales



## Gráficos finales

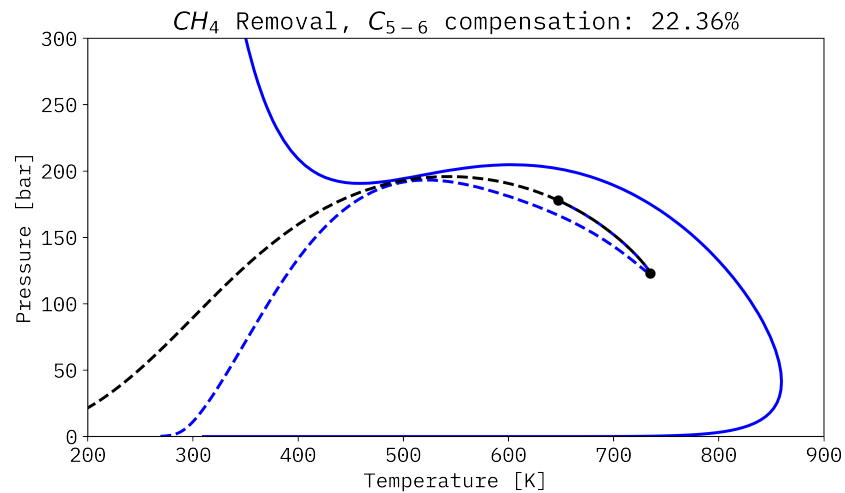


Figura: Envolvente caso 4

2022-07-25

## Python como herramienta de automatización de cálculo

└ Resolución en Python

└ Gráficos finales

└ Gráficos finales

Gráficos finales

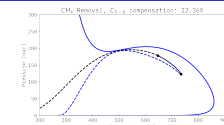


Figura: Envolvente caso 4

Muchas gracias!

Coming soon

- Fortran Refactoring: Estandarizando códigos...

2022-07-25

Python como herramienta de automatización de cálculo

└ Resolución en Python

└ Gráficos finales

└ Coming soon

Coming soon

- Fortran Refactoring: Estandarizando códigos...