**Методические указания по Автотесту**

В структуру автотеста входит родительский класс – **autotest\_class.py**, подклассы (**autotest\_pipe\_subclass.py, autotest\_pvt\_subclass.py, autotest\_gaslift\_subclass.py, autotest\_esp\_subclass.py**) наследуемые от родительского, разделенные по типам расчета (calc\_type: pipe, pvt, well, esp), а также модуль хранящий специальную функцию, которая соединяет все классы между собой ­– **autotest\_core.py**.

**Установка Автотеста**

Инструкция по установке Pipesim в Anaconda:

* скопировать папочку с пакетами [по ссылке](https://yadi.sk/d/hOVjOuK2oZvvyA) в папку Lib/site-packages с заменой.
* подгрузить базу насосов - pipesim\_2411.sdf через интерфейс pipesim. Options -> Catalog. Выбрать from file и указать путь к pipesim\_2411.sdf, выбрать опцию Duplicates overwrite и нажать Import
* скачать по ссылке выше json-базу насосов
* от имени администратора выполнить в cmd

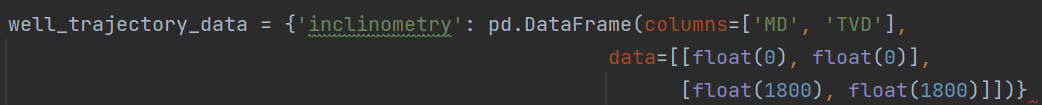
conda install -c conda-forge traits

conda install -c conda-forge isodate

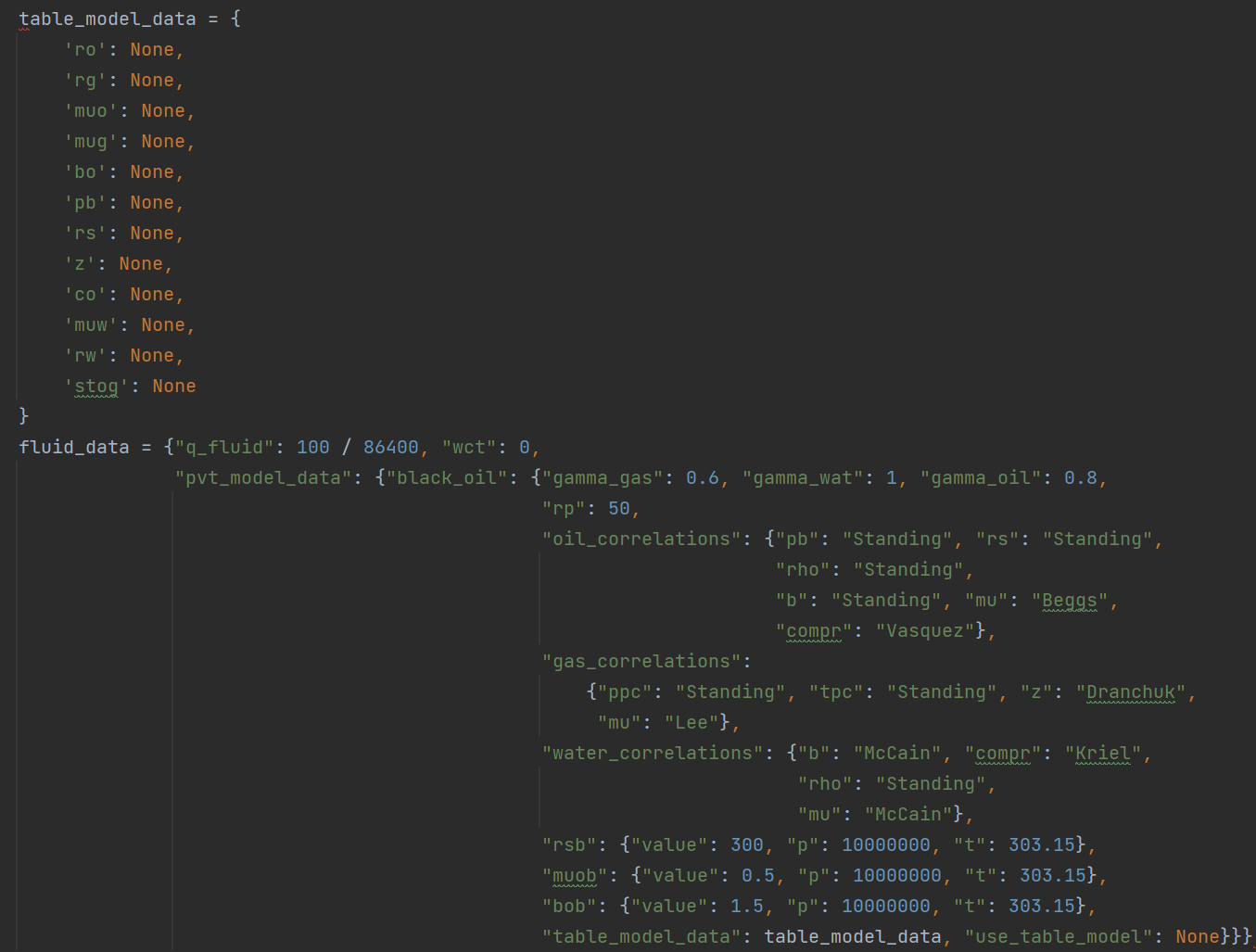
**Интерфейс**

В первую очередь необходимо импортировать autotest\_core.py, которая содержит функцию calc\_autotest. В качестве исходных данных требуются:

1) **well\_trajectory\_data** – словарь с траекторией скважины

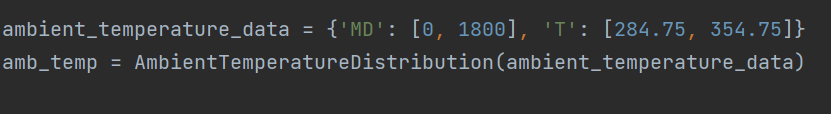


2) **fluid\_data** – словарь с PVT свойствами флюида

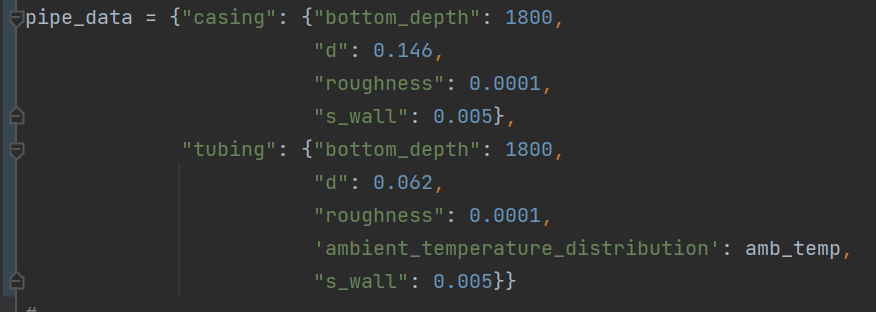


В случае если необходимо провести расчет с использованием таблицы PVT, укажите True в **use\_table\_model**

3) **ambient\_temperature\_data** – словарь с распределением температуры по стволу скважины



4) **pipe\_data** – словарь с конструкцией скважины

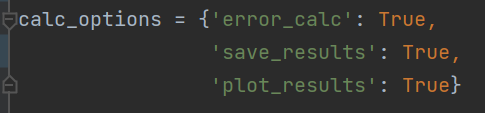


5) **calc\_type** – тип расчета (pipe, pvt, well, esp)

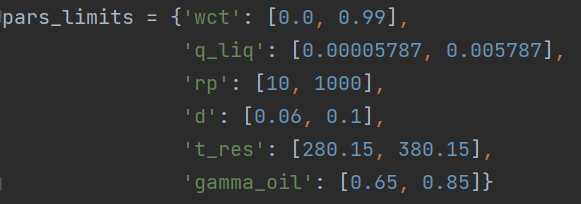
6) **file\_path** – путь, куда сохраниться эксель с результатами тестирования

7) **model\_path** – путь, куда сохраниться файл из pipesim в формате pips.

8) **calc\_options** – словарь с возможностью выбора опций, таких как расчет ошибки (error\_calc), save\_results, который позволяет сохранить распределения параметров по глубине скважины в отдельный эксель (**Применяется только при единичном расчете sample = False**), а также plot\_results, позволяющий строить графики параметров.

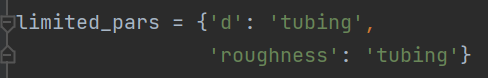


9) **pars\_limits** – словарь, который применяется только при массовом тестировании (sample = True). В нем можно задать границы параметров, в пределах которых будут генерироваться значения.



В зависимости от типа расчета, можно варьировать различные параметры.

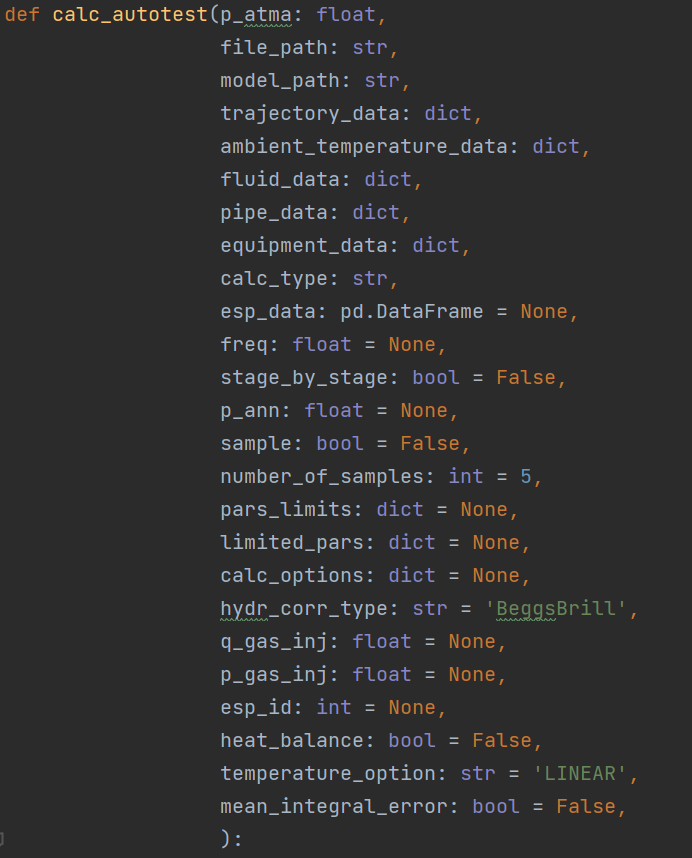
10) **limited\_pars –** словарь, в который вписываются те параметры, исключительно которые должны меняться. Например в pars\_limits варьируется диаметр **d**, но мы хотим, чтобы диаметр относился только к НКТ. Поэтому необходимо вписать ограничение на диаметр в limited\_pars в виде:



В противном случае будет меняться и диаметр обсадной колонны.

**Запуск автотеста**

В модуль **Autotest\_core.py** входит функция, производящая запуск автотеста – **calc\_autotest**.



**p\_atma:** давление, атма  
**file\_path:** путь сохранения файла с результатами  
**model\_path:** путь сохранения файлов в формате pips  
**trajectory\_data:** словарь с таблицей с инклинометрией, dict  
**ambient\_temperature\_data:** словарь с таблицей распределения температуры  
**fluid\_data:** словарь с параметрами флюида, dict  
**pipe\_data:** словарь с параметрами труб, dict  
**equipment\_data:** словарь с параметрами оборудования, dict  
**calc\_type:** тип расчета (pvt, well, pipe, esp)

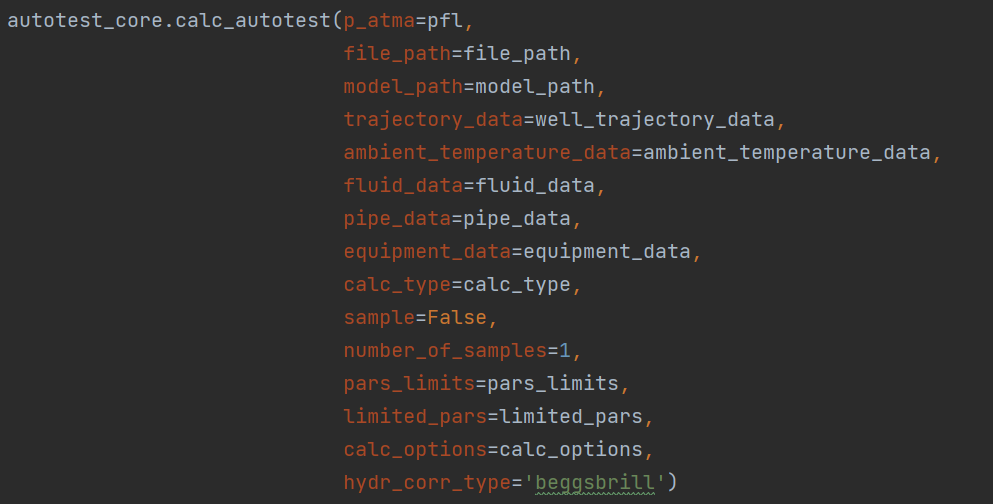
**esp\_data:** паспортные данные ЭЦН - pd.Series или dict

**freq:** частота ЭЦН  
**stage\_by\_stage:** расчет ЭЦН от ступени к ступени  
**p\_ann:** давление в затрубном пространстве  
**sample:** если True, то расчет произведется на произвольном наборе данных.  
 В противном случае рассчитается одиночный - main  
**number\_of\_samples:** количество набота данных (по умолчанию - 5)  
**pars\_limits:** словарь с параметром и его границами, в которых будет генерироваться данные, dict  
**limited\_pars:** словарь с однозначными соответствиями параметров для объектов, которые могут повторяться, dict  
 Например: {'d': 'tubing'} - тогда диаметр заменится только в tubing  
**calc\_options:** словарь с параметрами расчета, dict

**hydr\_corr\_type:** тип гидравлической корреляции  
**q\_gas\_inj:** закачка газлифтного газа, ст. м3/с (задается в случае если тип расчета well)  
**p\_gas\_inj:** давление закачки газлифтного газа, Па (задается в случае если тип расчета well)

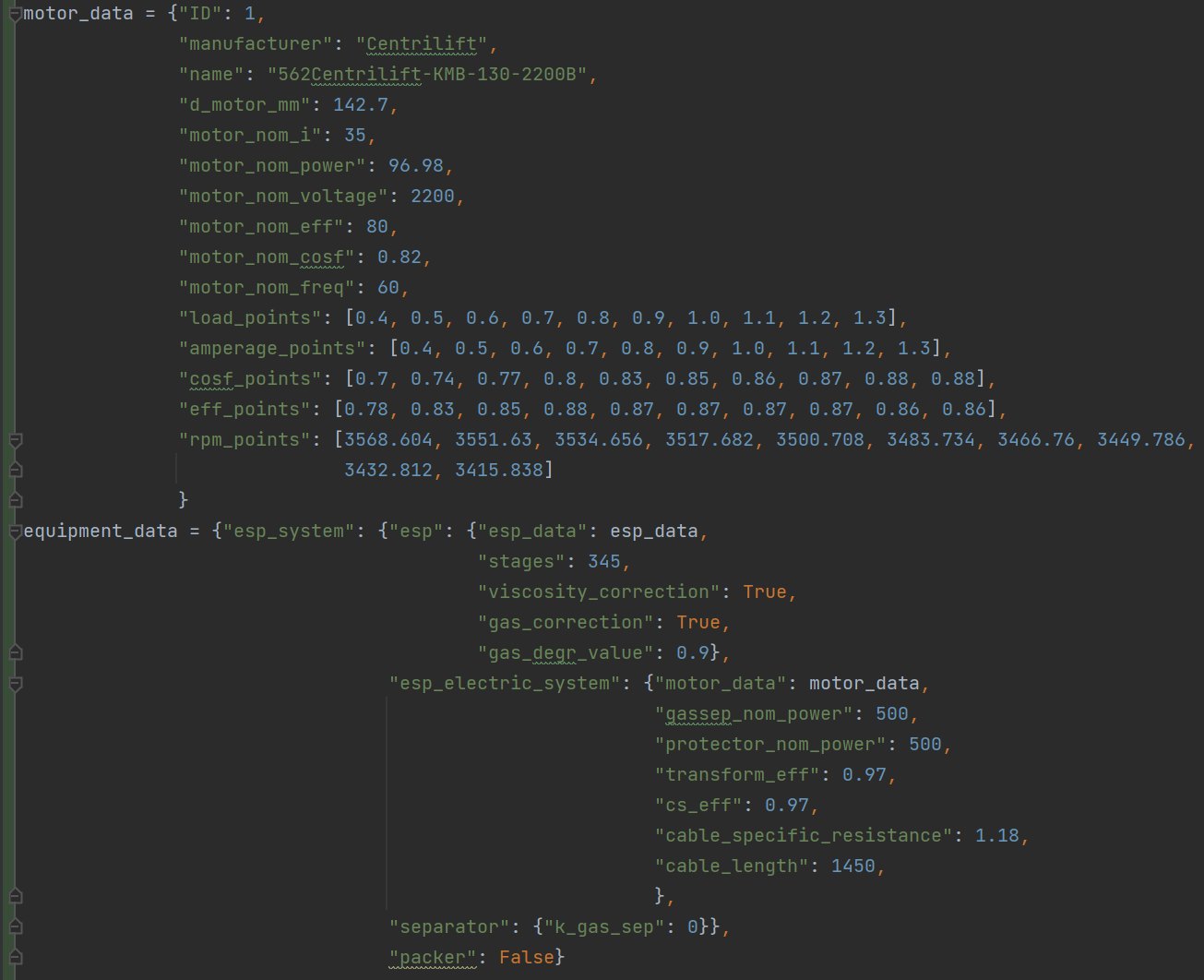
**esp\_id:** id насоса  
**heat\_balance:** опция учета теплопотерь, по умолчанию - False  
**temperature\_option:** 'CONST' - темпертарура константа по всей скважине,  
 'LINEAR' - линейное распределение темпратуры  
 (по умолчанию - LINEAR)  
**flow\_direction:** -1 - расчет от h\_start,  
 1 - расчет к h\_start  
 (по умолчанию - 1)  
**h\_start:** Точка от которого начнется расчет. 'top' или 'bottom' (по умолчанию top)  
**mean\_integral\_error:** True - среднеинтегральная ошибка,  
 False - относительная ошибка  
 (по умолчанию - False)

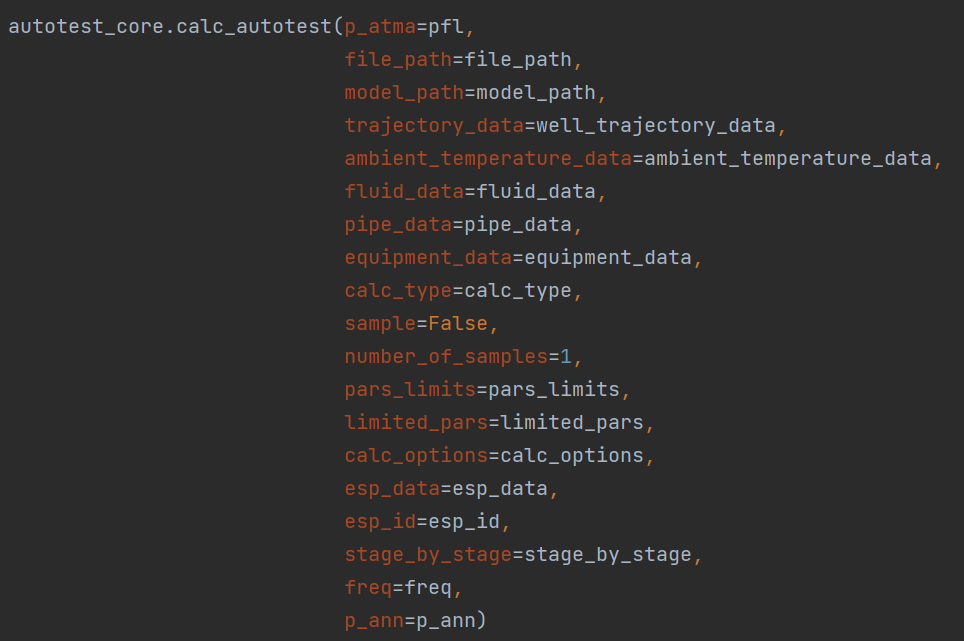
1) Если считаем трубу (calc\_type = ‘**pipe**’) или PVT свойства (calc\_type = ‘**pvt**’). В equipment\_data обязательно установление пакера ‘packer’: True.



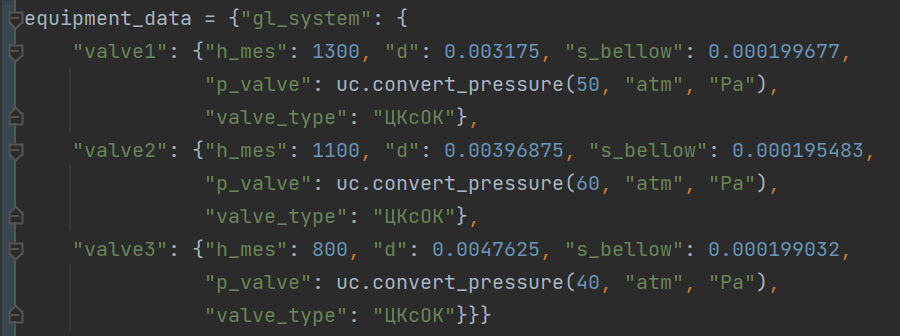
2) если считаем скважину свойства (calc\_type = ‘well’):

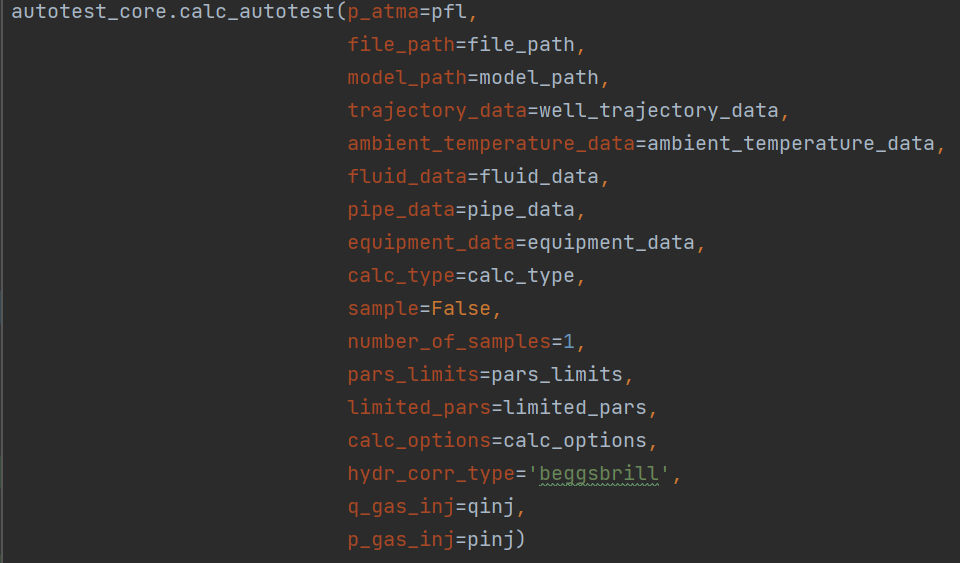
В зависимости от исходных данных, если в **equipment\_data** расположен **esp\_system**, значит рассчитываем скважину с ЭЦН. Также необходимо загрузить esp\_data – паспортные данные ЭЦН - pd.Series или dict.





Если же в **equipment\_data** расположен **gl\_system**, значит рассчитываем скважину с газлифтом.



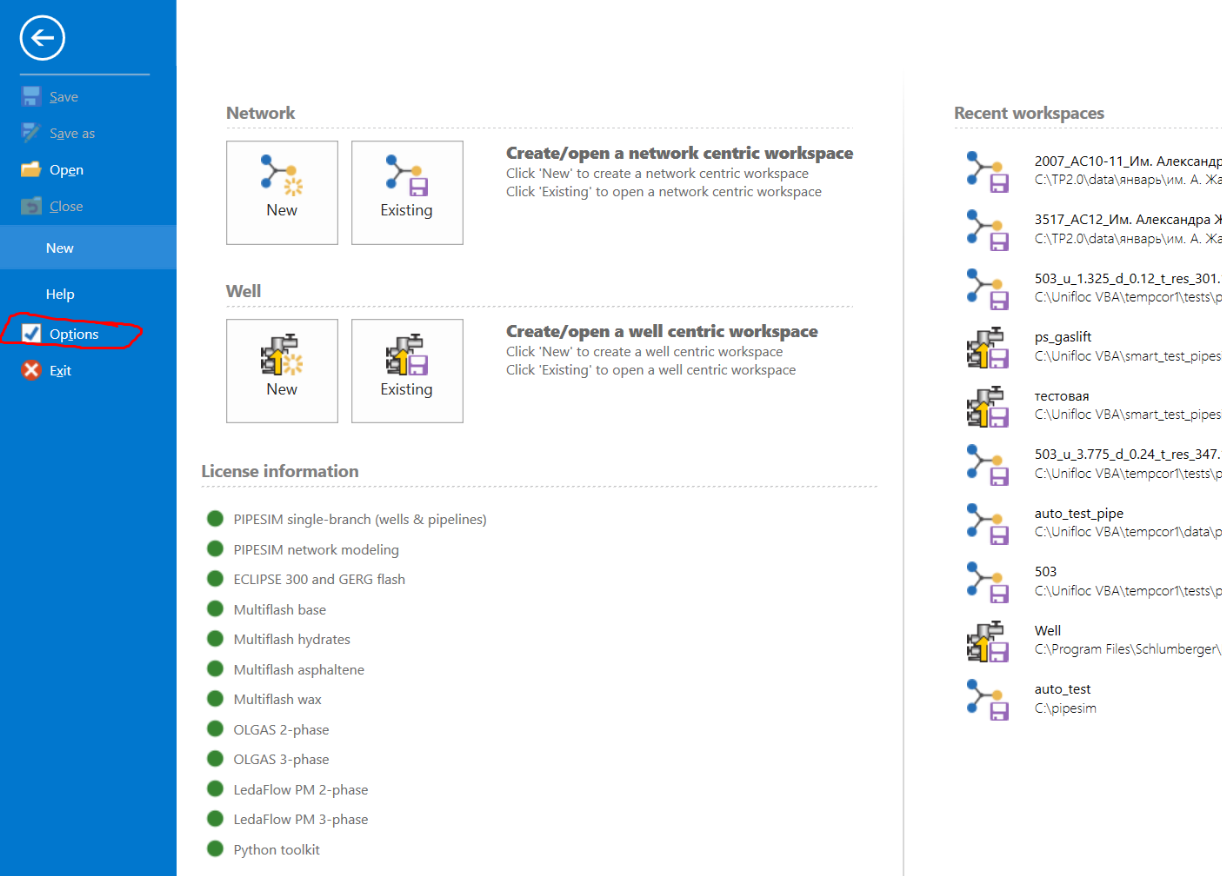


3) Тип расчета ЭЦН (calc\_type = ‘esp’)

Рассчитывает ЭЦН вне скважин. Вызов функции происходит аналогично с well (скважина с ЭЦН)

Для начала необходимо загрузить в библиотеку Pipesim новый каталог насосов. Файл с каталогом называется «esp\_db.sdf».

Для начала открываем Pipesim, далее заходим в Options.



Catalog – From file – Нажать на кнопку обзора (…) – Выбрать файл «esp\_db\_new.sdf». Далее Import. После небольшой загрузки, каталог насосов установится в Pipesim.

