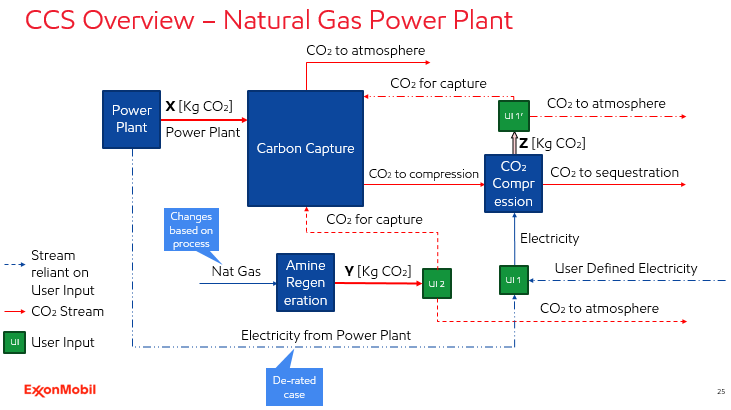
# Intro:

The current version of CCS has been added to these pathways: ng\_power\_greet, ng\_power\_aspen, coal\_power\_aspen, and coal\_power\_greet. For these pathways, the emissions that are considered are: CO2 and SOx emissions from burning natural gas or coal for amine regeneration, CO2 emissions from additional electricity needed for compression, and all emissions that are included in data for transportation of CO2 by pipeline.



This is a process diagram for CCS in SESAME. This is a schematic for a power plant, specifically, a natural gas power plant. Each box here represents a section of code that deals with each of these parts of CCS. The box labeled ‘Power Plant’ represents the code related to the original base case power plant with the dark red arrow representing the CO2 emissions associated with the production of a the user’s input amount of electricity. X does not change whether if the user selects the use of CCS or not and the ‘Power Plant’ box is the original natural gas to power code that has not been modified.

This X CO2 stream is then fed to another piece of code where the user can define a % capture of this stream of CO2, and the CO2 that isn’t captured is then released into the atmosphere. The captured CO2 must then be compressed before being transported to the location where it will be sequestered. This compression step requires an input of electricity, and, based on a user input, the electricity can come either from the power plant itself or the user can define another source of electricity by defining the lifecycle carbon intensity of that electricity.

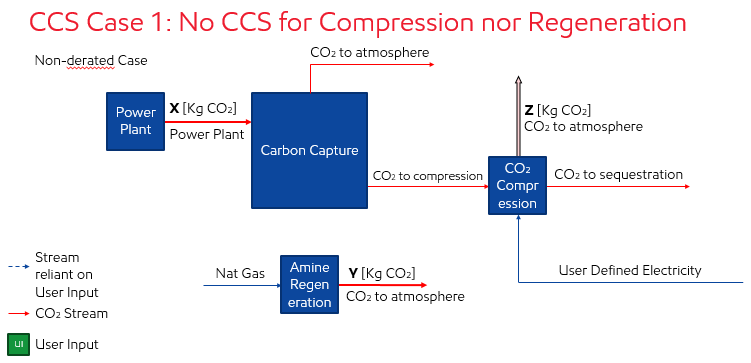
If a user selects that the electricity comes from the power plant itself, then it represents a de-rated case, and the emissions that are associated with that electricity, which are labeled as the Z stream here, are indistinguishable from the emissions that are related to the power plant, thus those emissions must go through the same process as the X stream in this diagram. By defining electricity from power plant at UI 1, the user is also defining that the emissions association with compression go through the capture process.

If the user defines that the electricity does not come from the power plant, then they define the lifecycle carbon intensity of that electricity and this then constrains the decision at UI 1’ to releasing the emissions associated with that electricity to the atmosphere. It is important to note that the user will define the lifecycle, cradle-to-grave emissions of this electricity, which they can do by running another SESAME run with the details of what they want this other electricity to be, and then take the result from that SESAME run and input it into a run that contains CCS.

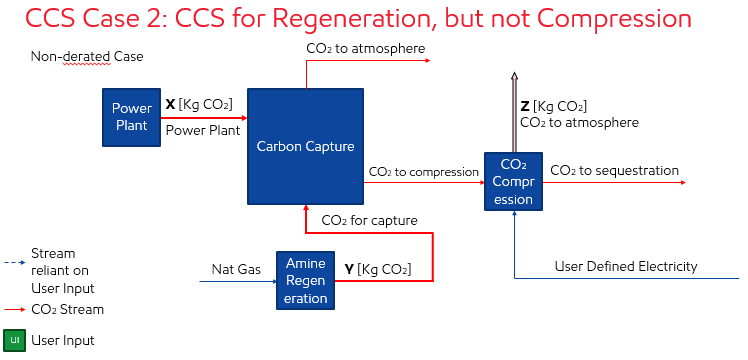
The last piece of the puzzle is the amine regeneration. After a certain amount of CO2 has been processed, the amine must be regenerated and this process requires steam. In a natural gas power plant, the steam is produced by burning natural gas which creates more emissions. However, what is burned changes depending on the specific scenario you’re considering, for example, in a coal power plant, they would likely burn coal to produce this steam, and this is reflected in the coal CCS code. Within SESAME, the user can define whether that stream, which is labeled Y, is released to the atmosphere or also goes through a CO2 capture, and again the user can define the capture percent. The complication with this is that the amount of electricity and natural gas that is used is based on the amount of CO2 that is being captured, but these two streams have their own emissions associated with them. Below are detailed the four permutations possible with the user inputs and the relevant equations that were solved to be implemented into SESAME.

Note: Technically the % capture of the compression (Y stream) is set to be the same as the % capture from power plant (X stream) but I’m keeping it as separate in the equations below and in the SESAME code just in case there are future implementation of SESAME where it makes sense to set a different % capture for the compression electricity so it’ll be easy to implement.

# Case 1:



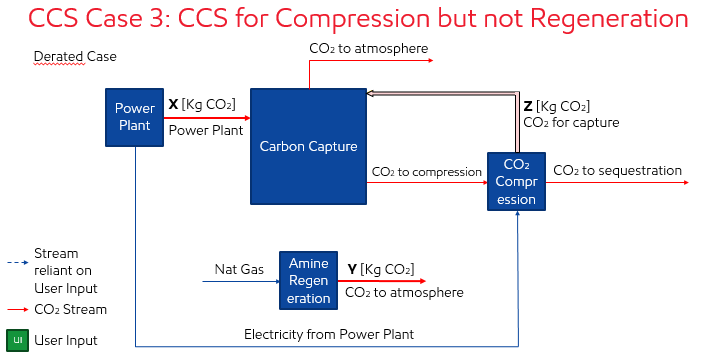
# Case 2:



a

b

Plug in Y from into this equation to calculate Z

Case 3: 

a

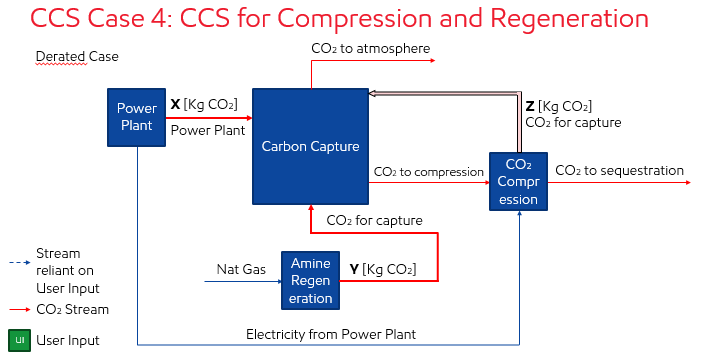
b

c

d

Plug in this calculated value for Z into the equation above

# Case 4:



a

b

c

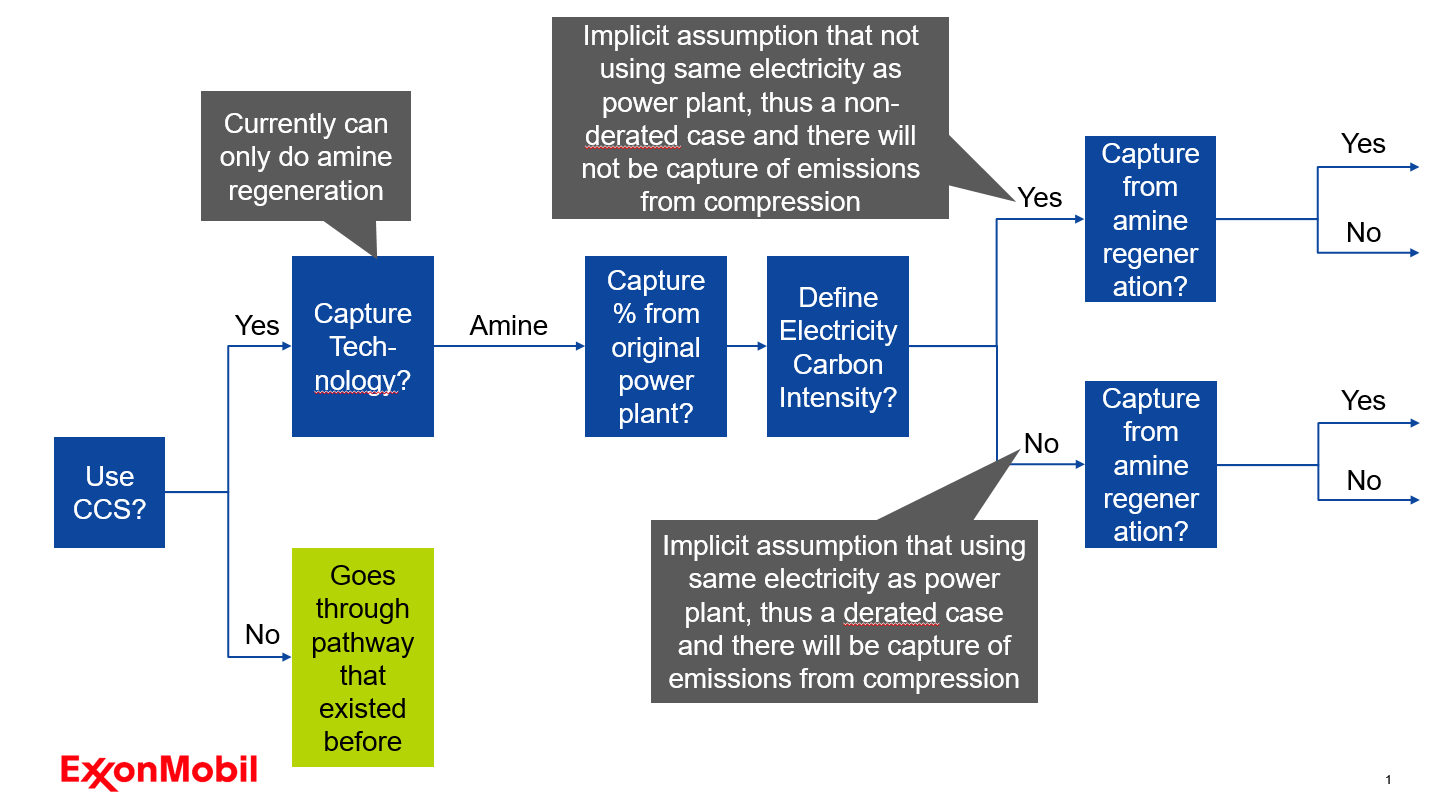
d

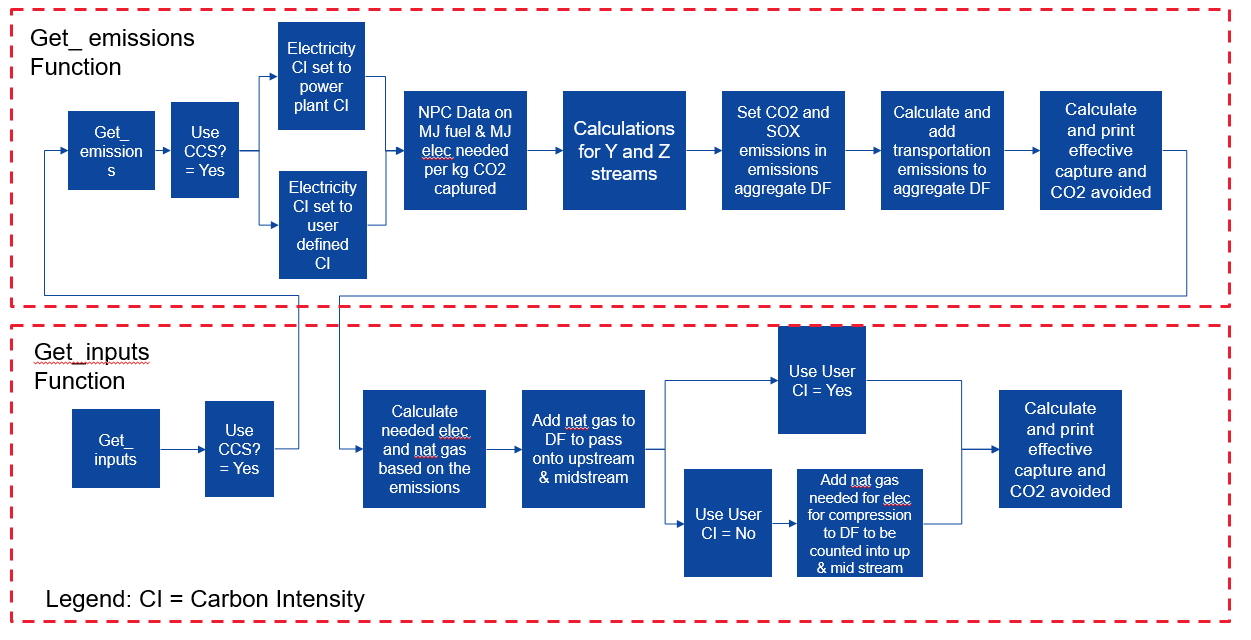
e

f

Plug in this calculated value for Y into the equation above

# All Cases





Note: the letter variables detailed in the equations above have no physical meaning and are the same variables that are used in each of the four cases within the SESAME code. The letter variables are defined such that the equations implemented in SESAME are simpler to read and type.

For all cases, three variables are printed when the CCS module is active: (1) the total kg of CO2 captured, (2) the effective CO2 captured, and (3) the kg of CO2 avoided. These are defined as follows:

Total kg of CO2 captured is calculated within each of the cases based on what streams are captured at what percent.

# Natural Gas CCS

The CO2 emissions from burning natural gas are calculated based on the required amount of energy from natural gas needed and the carbon content of natural gas from GREET. The SOx emissions are also accounted for when burning natural gas to produce steam for regeneration according to the sulfur content data of natural gas, also from GREET.

Note: since ASPEN doesn’t originally account for SOx emissions, the implementation to account for SOx in the ASPEN pathways is slightly different than the implementation in the GREET pathway in that I had to create a new dictionary entry for SOx within emissions[‘aggregate’] instead of updating the value in that dictionary for SOx

# Coal CCS

The basics for the Coal CCS are the same as for the Natural Gas CCS, and the equations for the four cases are the same with slight differences coming from the definitions of the energy, sulfur, and carbon content of the different types of coal (Bituminous, Subbituminous, Lignite, and Mix for Electricity from GREET). Due to different values for these three variables based on the type of coal selected, there is an additional section of code that selects the correct value based on the type of coal appropriate.

# CCS\_LCA\_TEA\_data.xlsx

* NPC Data tab: includes the consolidated table with information from all the relevant tables in the NPC report that I could find for both LCA (highlighted in yellow) and TEA values for a variety of considered facilities. Includes Estimated CO2 pipeline costs and volume-weighted storage costs. Also includes the conversions done to create the csv files for LCA inputs for CCS for natural gas power plant, coal power plant, ethanol plant, and hydrogen plant.
* Checking SESAME tab: includes the tests for the four cases present for CCS with the hand calculated values in the colorful case boxes with inputs defined at the top most box (the nat gas and elec boxes were used to check the accuracy of the equations themselves and assume that all streams are being captured at 85%).
* Nat\_gas CC Results tab: results for cases 1:4 for natural gas runs from SESAME with specific inputs stated if there is more than one option in SESAME. Also includes the water fall charts on the right for case 1 and 4.
* Coal Results tab: same type of results as the Nat\_gas CC Results tab, but for coal runs in SESAME\*

\*please note that the values in this sheet will be slightly off because of an error that was caught after creating the graphs where for the incremental natural gas needed to pass into the upstream and midstream modules due to compression, a conversion was incorrect and was being divided by 1E5 instead of 1E6. However the amount of natural gas needed for this is small and thus the overall results shouldn’t change by that much.

# Future Improvements/Work

* Account for emissions other than CO2 from additional electricity needed for compression
  + If the electricity from compression comes from the power plant itself you would need to account for this. Otherwise, if the user is defining their own electricity carbon intensity, you can account for this by having the user input lifecycle kg CO2-**eq**/kWh instead of just kg CO2/kWh (since it’s kg CO2-eq, you can pass it into SESAME as a value for CO2 and it should work, as long as there’s not a plan at some point to give a breakdown of the contribution associated with different emissions)
* In the coal\_power\_aspen pathway add integration between the process and midstream stages so that the user isn’t asked twice what the type of coal they want to consider
* Implementation of CCS for hydrogen, ethanol, and natural gas processing pathways
  + I’ve already created the files for electricity and energy needed as inputs for hydrogen and ethanol in the same format as what I have implemented for natural gas and coal to power and added them to SESAME in the appropriate folders within the process stage, but the actual code needs to be added to the hydrogen and ethanol process files to be able to read and process the data
* Add loss of CO2 on pipeline? (should be pretty simple)
* Allow flexibility to burn something other than the fuel used in the power plant itself for regeneration
* Compare values from SESAME CCS to Athena values

# Things to keep an eye out for

* If the structure of the folders within the pathway folder changes, you will need to change the line under pipeline CO2 transportation emissions which has pd.read\_csv(Path[:-len(“…”)] + “…”. The file pathway represented by the … will need to be updated.

Notes by Rebecca Grekin in summer 2020