CO2BLOCK-to-python user guide

CO2BLOCK-to-python provides an estimate of the CO_2 storage capacity of a saline aquifer, this software is a direct translation into Python of the original programme in MATLAB. For more details on the background theory, please refer to De Simone and Krevor (2021) on the original CO2BLOCK repository:

co2block/CO2BLOCK: A tool for estimates of storage resource capacity in saline aquifers (github.com)

This tool is written in Python and is composed of six scripts. Users only need to open and modify the script *CO2BLOCK-in-Jupyter.ipynb* or *CO2BLOCK-in-Spyder.py* (depending on the extension required for the platform used to run the code). There is no need to open and modify other scripts, this is not recommended.

Downloading the Anaconda package

We recommend using Anaconda as the platform for opening and running this Python code, and provide instructions for opening the code files in either Spyder or Jupyter Notebook (both are provided in the Anaconda package):

- Visit anaconda.com and install the latest version of Anaconda (at least 'Python 3.7')
- Accept the license agreement and choose the installation location on your PC
- Once installed, search for 'Anaconda Navigator' in the Start Menu in Windows and launch either Jupyter Notebook or Spyder (or another Integrated Development and Learning Environment)

Running the code

Download the zip folder of code scripts and save the individual scripts on the PC.

If using Spyder: open Spyder environment, select the 'open file' icon in top left and select *CO2BLOCK-in-Spyder.py*

If using Jupyter Notebook: launch Jupyter Notebook from the Anaconda Navigator (should open your computer directory in the browser), navigate the directory and select *CO2BLOCK-in-Jupyter.ipynb*

Editing inputs

It is necessary to provide an input file of aquifer parameters which must be formatted exactly as the provided example file *example_data.xlsx* (column order must not be changed). The below table shows the parameters required and the parameters that can be calculated using default values, however filling in as many known parameters as possible will allow for the most accurate prediction of storage capacity.

The Excel file must be saved in the same folder as the Python scripts so that it can be called by the *CO2BLOCK-in-Jupyter.ipynb* script. If it is not saved in this folder then the user must define the directory in this script instead.

Required parameters	**Default parameters**	**Default values**
Domain BC (open/closed)	Rock compressibility, c_r	5×10^{-4} MPa^{-1}
Shallowest depth, ζ	Brine compressibility, c_w	3×10^{-4} MPa^{-1}
Mean depth, ζ_m	Brine salinity, χ	180000 ppm
Thickness, H	Shallowest pressure, p_0	10 MPa/km $\times\zeta$
Surface area, A	Mean pressure, $\overline{p_0}$	10 MPa/km $\times\zeta_m$
Permeability, κ	Mean Temperature, $\overline{T_0}$	$33°$/km $\times\zeta_m + 15°$
Porosity, ϕ	CO_2 density, ρ_c	calculated with respect to $\overline{p_0}$ and $\overline{T_0}$, according to Redlich and Kwong (1949) (with the parameters proposed by Spycher et al. (2003))
	CO_2viscosity, μ_c	calculated with respect to $\overline{p_0}$ and $\overline{T_0}$, according to Altunin and Sakhabetdinov (1972)
	Brine viscosity, μ_w	calculated with respect to $\overline{T_0}$ and χ, according to Batzle and Wang (1992)
	Vertical stress, σ_v	23 MPa/km $\times\zeta_m$
	Stress ratio, k_0	0.7
	Friction coefficient, φ	$30°$
	Cohesion, C	0 MPa
	Tensile strength, S_0	0 MPa

Users must define the name of the Excel data file in the *CO2BLOCK-in-Jupyter.ipynb* (or *CO2BLOCK-in-Spyder.py*) script. Users must also set some parameters in this script. These are as follows:

- correction = can be set to 'on' or 'off' to apply correction for the error in superposition over large numbers of wells ('off' is the most conservative option)
- dist_min = minimum inter-well distance [km]
- dist_max = maximum inter-well distance [km]. If 'auto' is set then the tool will calculate maximum distance according to the reservoir area
- nr_dist = number of inter-well distances to explore (essentially the number of iterations – higher number will slow the programe down)
- nr_well_max = maximum number of wells in reservoir. If 'auto' is set, the tool automatically calculates this based on reservoir area and minimum inter-well distance

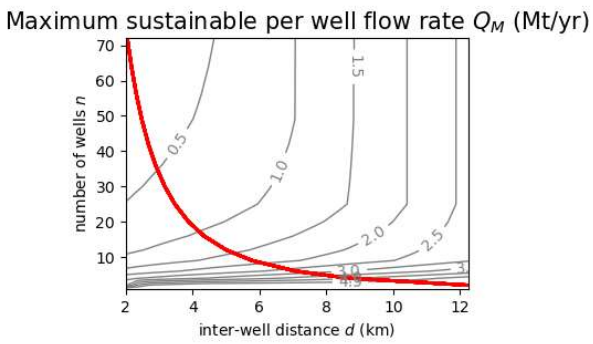
- rw = well radius [m]
- time_yr = time of injection [years]
- maxQ = maximum sustainable injection rate per well due to technological limitations (Mton/years)

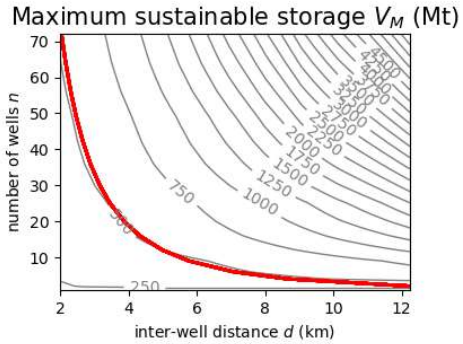
Output

Results are the maximum sustainable injection rate per well [Mton/yr] and the maximum sustainable injected mass [Mton] which are output in Excel tables and graphs as shown.

The contours give the outcome for various numbers of wells and inter-well distances. The red line shows the limit due to size of the reservoir so anything below the red line is viable, the tables print all plausible values for flow rate and storage after the constraints due to reservoir size have been applied.

These figures correspond to the example data provided in the Excel file attached.





number_of_wells	Q_M_for_d_2000_m	Q_M_for_d_2353_m
1	4.9995	4.9995
2	2.939132945	4.069453454
4	2.359490433	2.46777783
6	1.63398807	1.720015122
9	1.130037374	1.19563736
12	0.91205692	0.970958481
16	0.731319946	0.783103708

number_of_wells	V_M_for_d_2000_m	V_M_for_d_2353_m
1	149.985	149.985
2	176.3479767	244.1672073
4	283.1388519	296.1333397
6	294.1178525	309.602722
9	305.110091	322.8220873
12	328.3404911	349.5450532
16	351.033574	375.8897799

PERENCO CCS Subsurface Team

- Alice Simmonds
- Jeremy Fortun
- Wei Cher Feng
- Christopher Furby
- Eric Blouin

References

De Simone and Krevor (2021). A tool for first order estimates and optimisation of dynamic storage resource capacity in saline aquifers. International Journal of Greenhouse Gas Control, 106, 103258.

Original scripts in MATLAB: co2block/CO2BLOCK: A tool for estimates of storage resource capacity in saline aquifers (github.com)