The Journal of Open Source Software

Pywaterflood: Well connectivity analysis through capacitance-resistance modeling

Frank Male [1¶]

1 Pennsylvania State University, University Park, PA, USA ¶ Corresponding author

DOI: 10.21105/joss.06191

Software

- Review
- Repository
- Archive

Editor: Jayaram Hariharan

Reviewers:

- @mgcooper
- @amandersillinois

Submitted: 16 September 2023
Published: 14 March 2024

Summary

Well connectivity analysis has many applications for subsurface energy, covering any project where nearby wells are expected to influence one another, whether they are in oil or gas fields, geothermal fields, or an aquifer. After completing a well connectivity analysis, reservoir managers know the degree and time-dependence of influence of one well's behavior on another's. With this knowledge, they can better allocate injections and plan well interventions. Capacitance Resistance Models are useful for performing well connectivity analysis with limited information about the geology of the reservoirs involved (Yousef et al., 2006). They are so-called because the equations describing well influence mimic a network of capacitors and resistors (Holanda et al., 2018).

`Pywaterflood` is a Python package that uses Capacitance Resistance Modeling (CRM) to estimate well connectivity. It is a portmanteau of "Python" and "waterflood," where a waterflood is an oil reservoir with water injection designed to increase reservoir pressure and move oil towards producing wells. The `CRM` submodule forms the bulk of this package. It can perform CRM with differing levels of complexity, from assuming that producing and injecting wells share one universal time constant, to each producer has the same time constant with all injectors, to each producer-injector pair has an its own time constant. CRM was developed by Yousef et al. (2006).

The `MPI` (Multiwell Productivity Index) submodule uses a geometrical model of well influence (Valko et al., 2000), extended and applied to reservoirs with both injecting and producing wells (Kaviani & Valkó, 2010). As a geometrical model, it can assist in planning reservoirs before any production or injection has begun.

Statement of need

Interwell connectivity analysis is important for understanding the geology of subsurface systems. This can be used to improve oil recovery efficiency (Albertoni & Lake, 2003), better sequester CO_2 (Tao & Bryant, 2015), and optimize geothermal fields (Akin, 2014). Holanda et al. (2018) enumerate four uses for CRM results:

1. Finding sealing faults and high-flow-connectivity pathways
2. Investigating connectivity between adjacent reservoirs and reservoir compartments
3. Measuring the per-well effectiveness of fluid injection
4. Optimizing injection, either through redirecting fluid to different wells or to inform the placement of new wells

`Pywaterflood` uses a reduced-physics model to match connections between injecting and producing wells. As explained in Holanda et al. (2018), CRM provides a method for connectivity analysis that is more sophisticated than empirical decline analysis but also more approachable than full reservoir simulation.

There is another publicly available tool for CRM analysis of reservoirs like `pywaterflood`: Sayarpour (2008). However, that tool comes in the form of an Excel workbook with no associated license. This python package, with performance parts written in Rust, provides more extensibility and better performance than an Excel file. There are other programs for performing waterflood analysis with CRM in the industry, but they are not open sourced and publicly available. A survey of Github reveals the following examples of CRM: a matlab script, a proxy-CRM model "highly inspired by" pywaterflood, and another python script.

The `pywaterflood` library can perform the following tasks:

1. Estimate connectivity between wells in fluid or pressure communication with `CRM`
2. History-match and forecast the production of wells in waterfloods, CO_2 floods, or geothermal fields with `CRM`
3. Provide purely geometric estimates of well connectivity before production data is available with `MPI`

In the period from 22 January 2024 to 21 February 2024, the pywaterflood package was downloaded from PyPI 772 times. It has been used for the author's work in analyzing waterfloods in two papers in preparation.

Background

The governing equation for CRM to predict production (q) at a particular time is

$$q(t_n) = q(t_0)e^{-\left(\frac{t_n - t_0}{\tau}\right)} + \sum_i \sum_{k=1}^{n} \left(\left(1 - e^{-\frac{\Delta \Delta t_k}{\tau_i}} \right) \left(w_i(t_k) - J_i \tau_i \frac{\Delta p_i(t_k)}{\Delta t_k} \right) e^{-\frac{t_n - t_k}{\tau_i}} \right).$$

It has three components that feed $q(t_n)$, the production from a well at the n'th period in time:

- $q(t_0)$: production from fluid expansion, decaying exponentially
- $w_i(t_k)$: injected fluid for the previous periods for the i'th injector
- Δp_i: changes in pressure for previous periods for the i'th injector

Acknowledgements

I am thankful for Jerry Jensen and Larry Lake for their mentorship, introduction to Capacitance-Resistance Modeling, and presentation of several interesting problems for `pywaterflood`. Ian Duncan was responsible for useful discussions and further problems to apply CRM to. Danial Kaviani provided advice for the MPI submodule. Software development funding was provided by the Department of Energy grant "Optimizing Sweep based on Geochemical and Reservoir Characterization of the Residual Oil Zone of Hess Seminole Unit" (Principal investigator: Ian Duncan), the State of Texas Advanced Resource Recovery program (PI: William Ambrose, then Lorena Moscardelli), and by Pennsylvania State University faculty funds.

This project relies on the following open-source Python packages: NumPy (Harris et al., 2020; Walt et al., 2011), SciPy (Virtanen et al., 2020), and pandas (McKinney, 2010). It also uses the Rust crates ndarray, numpy, and pyo3.

References

Akin, S. (2014). Optimization of reinjection allocation in geothermal fields using capacitance-resistance models. *Thirty-Ninth Workshop on Geothermal Reservoir Engineering. Stanford University*. https://pangea.stanford.edu/ERE/pdf/IGAstandard/SGW/2014/Akin.pdf

Albertoni, A., & Lake, L. W. (2003). Inferring interwell connectivity only from well-rate fluctuations in waterfloods. *SPE Reservoir Evaluation & Engineering*, *6*(01), 6–16. https://doi.org/10.2118/83381-PA

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Holanda, R. W. de, Gildin, E., Jensen, J. L., Lake, L. W., & Kabir, C. S. (2018). A state-of-the-art literature review on capacitance resistance models for reservoir characterization and performance forecasting. *Energies*, *11*(12). https://doi.org/10.3390/en11123368

Kaviani, D., & Valkó, P. P. (2010). Inferring interwell connectivity using multiwell productivity index (MPI). *Journal of Petroleum Science and Engineering*, *73*(1-2), 48–58. https://doi.org/10.1016/j.petrol.2010.05.006

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Sayarpour, M. (2008). *Development and application of capacitance-resistive models to water/carbon dioxide floods*. The University of Texas at Austin. https://repositories.lib.utexas.edu/handle/2152/15357

Tao, Q., & Bryant, S. L. (2015). Optimizing carbon sequestration with the capacitance/resistance model. *SPE Journal*, *20*(05), 1094–1102. https://doi.org/10.1016/j.egypro.2013.06.290

Valko, P. P., Doublet, L., & Blasingame, T. (2000). Development and application of the multiwell productivity index (MPI). *SPE Journal*, *5*(01), 21–31. https://doi.org/10.2118/51793-PA

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … SciPy 1.0 Contributors. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2

Walt, S. van der, Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Yousef, A. A., Gentil, P., Jensen, J. L., & Lake, L. W. (2006). A capacitance model to infer interwell connectivity from production-and injection-rate fluctuations. *SPE Reservoir Evaluation & Engineering*, *9*(06), 630–646. https://doi.org/10.2118/95322-PA