The Journal of Open Source Software

Bluebonnet: Scaling solutions for production analysis from unconventional oil and gas wells

Frank Male [1,2]¶, **Michael P. Marder**[2], **Leopoldo M. Ruiz-Maraggi**[2], and **Larry W. Lake**[2]

1 Pennsylvania State University, University Park, PA, USA **2** University of Texas at Austin, TX, USA ¶ Corresponding author

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: unpublished

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC BY 4.0).

DRAFT

Summary

Unconventional oil and gas wells are only productive due to extensive hydraulic fracturing treatments. Therefore, the character of their production over time is greatly influenced by engineering decisions. However, it can be difficult to separate the engineering decisions from the effects due to fluid properties. Also, during production these wells might be producing oil, gas, and water simultaneously, with each phase interacting with the others. Numerical tools are necessary to fully capture the effects of fluid properties on production.

`Bluebonnet` is a Python package that uses dimensionally scaled solutions of a pressure diffusivity equation to analyze, history-match, and forecast production of tight-oil and shale gas wells. `Bluebonnet` has been developed to help researchers and petroleum engineers analyzing production data from unconventional (shale gas and tight oil) wells. It provides the user with a set of tools to evaluate production performance of tight-oil and shale gas wells. These tools are:

1. `fluids` calculates pressure-volume-temperature properties for oil, water, and gas phases.
2. `flow` builds physics-based production curves and estimates hydrocarbon recovery factors.
3. `forecast` fits and forecasts unconventional production.

The `fluids` submodule estimates the formation volume factors, solubility ratios, and viscosity for the oil, water and gas phases given the reservoir temperature, oil API gravity, gas specific gravity, and initial gas/oil ratio. Figure 1 illustrates the plots of the *(a)* formation volume factors and *(b)* viscosities for the oil, gas, and water phases using the `fluids` submodule.

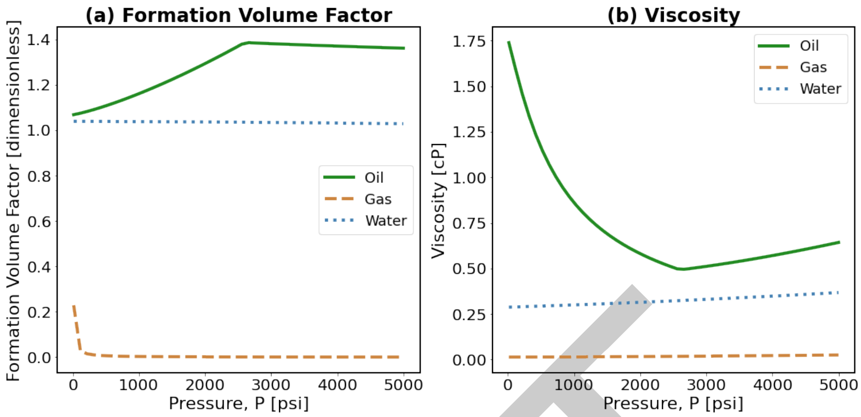

Figure 1: Plots of *(a)* formation volume factors and *(b)* viscosities for the oil, gas, and water phases using the `fluids` submodule.

The `flow` submodule solves the pressure diffusivity equation to provide estimates of the hydrocarbon production over time and the hydrocarbon recovery factors. This module allows the user to estimate production for shale gas wells using a scaled solutions of the single-phase real gas diffusivity equation (Male, 2015; Patzek et al., 2013). In addition, this module simulates production for tight-oil and gas condensate wells using a two-phase scaled solution of the pressure diffusivity equation (Ruiz Maraggi et al., 2022a). The `flow` submodule also allows users to capture production variations due to changes in bottomhole pressure.

Figure 2 shows the gas recovery factors for single-phase ideal gas, real gas, and multiphase scaled flow solutions using the `flow` submodule.

DRAFT

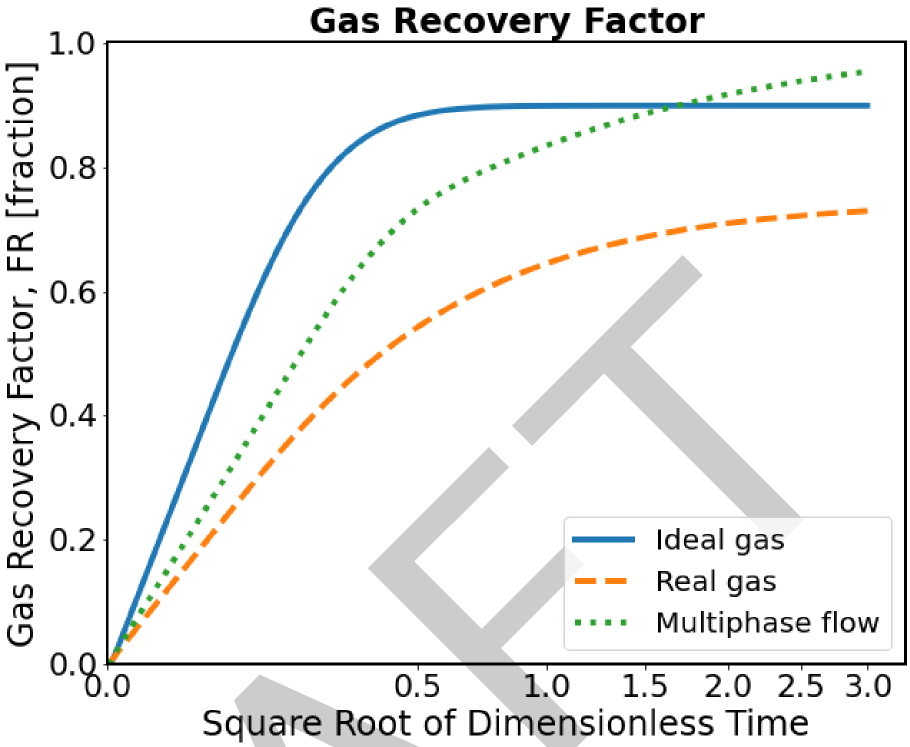

Figure 2: Plots of the gas recovery factors for ideal gas, real gas, and multiphase flow solutions of the pressure diffusivity equation using the `flow` submodule.

The `forecast` submodule performs history matches and forecasts the production of unconventional wells using the scaling solutions present in the `flow` module. Figure 3 illustrates the history-match of a gas well using the single-phase real gas flow solution.

DRAFT

The Journal of Open Source Software

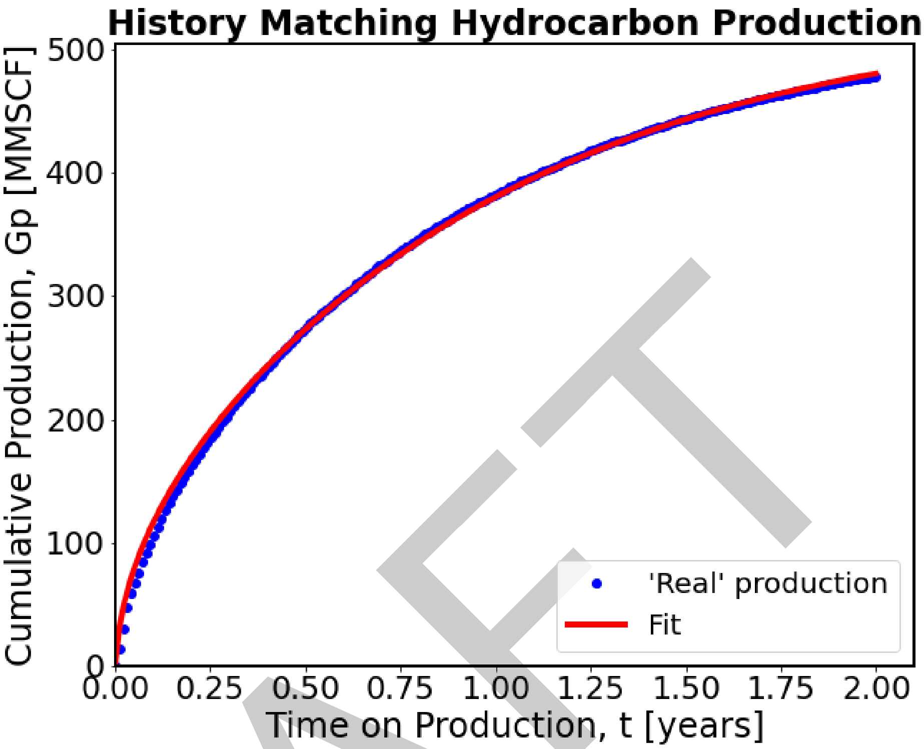

Figure 3: History-match of a shale gas well (blue dotted curve) using the single-phase real gas flow solution (solid red curve).

The `forecast` submodule also allows users to history-match and forecast production of wells subject to variable bottomhole pressure conditions using a modification of the approach developed by Ruiz Maraggi et al. (2022b).

Figure 4 illustrates the *(a)* history-match of the gas well #20 from the SPE data repository (Petroleum Engineers, 2021), subject to variable bottomhole flowing pressure conditions *(b)*.

DRAFT

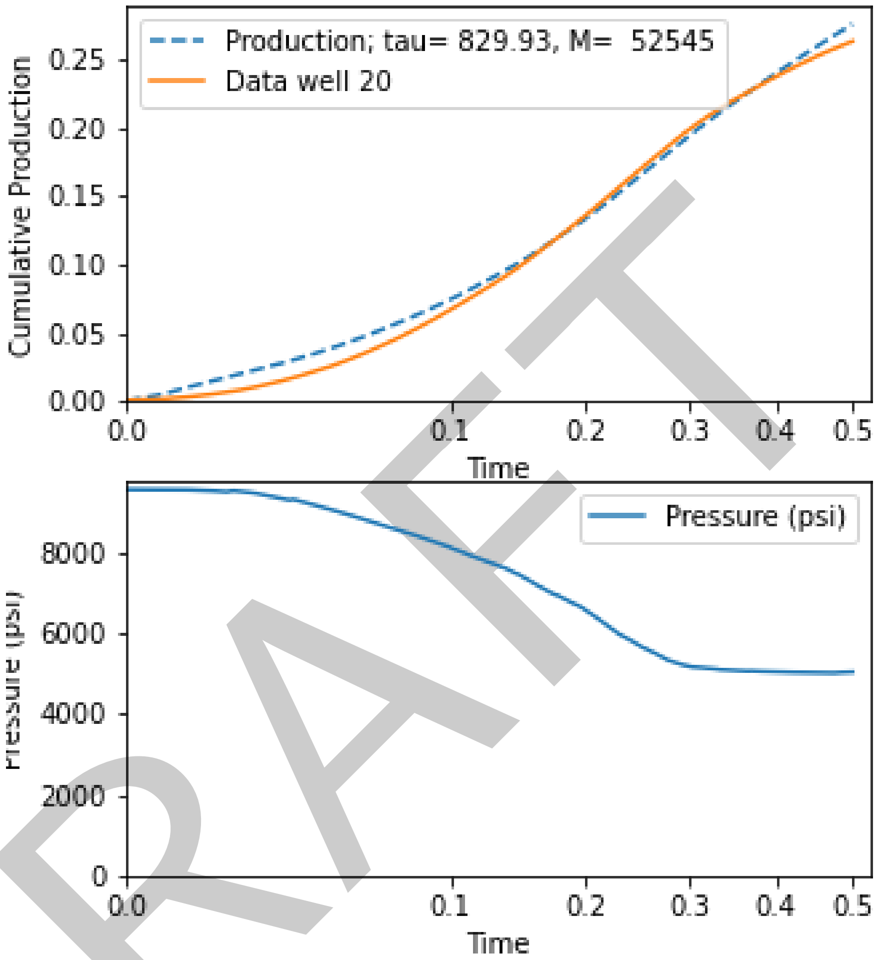

Figure 4: Plots for the *(a)* history-match of the gas well #20 from Petroleum Engineers (2021), subject to variable bottomhole flowing pressure conditions *(b)*.

DRAFT

Statement of need

`Bluebonnet` is a Python package using petroleum engineering methods to perform production analysis of hydrofractured wells. Parts of this code were first developed to assist in determining U.S. shale gas reserves (Male, 2019; Patzek et al., 2013).

There are no free open-source tools that use physics-based scaled flow solutions of the diffusivity equation to perform decline-curve and rate-transient analysis for unconventional reservoirs like `bluebonnet`. The goal for producing this software package is to provide researches and reservoir engineers with a free and open source tool suitable to analyze production from unconventional (tight oil and shale gas) reservoirs.

The present library can be used for the following tasks:

1. Estimate fluid properties of reservoir fluids.

2. Build type curves and recovery factors for shale gas and tight-oil reservoirs.
3. History-match and forecast the production of shale gas and tight-oil wells.
4. Perform Rate-transient analysis (rate-time-pressure) of unconventional reservoirs.

Acknowledgements

This library would not exist without Tad Patzek introducing several of the authors to the problem of unconventional production forecasting and kindly providing code samples of the pressure diffusivity equation. We thank ExxonMobil for funding this project with the grant "Heterogeneity and Unconventional Production" (PI: Michael Marder). Valuable comments and criticism came from discussions with Gary Hunter, Emre Turkoz, Zaheera Jabeen, and Deniz Ertas.

This project relies on the following open-source Python packages: NumPy (Harris et al., 2020; Walt et al., 2011), SciPy (Virtanen et al., 2020), matplotlib (Hunter, 2007), and pandas (McKinney, 2010).

The authors would like to thank the Society of Petroleum Engineers (SPE) for providing open access to production data from unconventional wells through the SPE Data Repository, Data Set 1 (Petroleum Engineers, 2021) used to illustrate the application of this package.

References

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

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Male, F. (2015). *Application of a one dimensional nonlinear model to flow in hydrofractured shale gas wells using scaling solutions* [PhD thesis]. University of Texas at Austin.

Male, F. (2019). Assessing impact of uncertainties in decline curve analysis through hindcasting. *Journal of Petroleum Science and Engineering*, *172*, 340–348. https://doi.org/10.1016/j.petrol.2018.09.072

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

Patzek, T. W., Male, F., & Marder, M. (2013). Gas production in the Barnett Shale obeys a simple scaling theory. *Proceedings of the National Academy of Sciences*, *110*(49), 19731–19736. https://doi.org/10.1073/pnas.1313380110

Petroleum Engineers, S. of. (2021). *SPE Data Repository, Dataset # 1*. https://www.spe.org/en/industry/data-repository

Ruiz Maraggi, L. M., Lake, L. W., & Walsh, M. P. (2022a). A Two-Phase Flow Model for Reserves Estimation in Tight-Oil and Gas-Condensate Reservoirs Using Scaling Principles. *SPE Reservoir Evaluation & Engineering*, *25*(01), 81–98. https://doi.org/10.2118/199032-PA

Ruiz Maraggi, L. M., Lake, L. W., & Walsh, M. P. (2022b). Rate-Pseudopressure Deconvolution Enhances Rate-Time Models Production History-Matches and Forecasts of Shale Gas Wells. *SPE Reservoir Evaluation & Engineering*, 1–20. https://doi.org/10.2118/208967-PA

DRAFT

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

DRAFT