Bayesian inference of deep learning based reduced order model: How does it compare to Bayesian inference of full model?

Saumik Dana[1*] and Karthik Reddy Lyathakula[2]

[1*]University of Southern California, Los Angeles, CA, USA.
[2]North Carolina State University, Raleigh, NC, USA.

*Corresponding author(s). E-mail(s): saumik@utexas.edu;
Contributing authors: klyatha@ncsu.edu;

Abstract

Going from data to build a robust differential/algebraic equation governing the phenomena requires calibration of the coefficients which can have a large range of possible values. The Bayesian inference framework allows narrowing down the range, albeit with a potential snag if the forward model being built is too sophisticated. A reduced order model is needed to make Bayesian inference feasible in such cases, and deep learning in the form of LSTM autoencoders are deployed. We compare the Bayesian inference of a full model to that with a reduced order model to get a proof of concept

Keywords: Time series reconstruction, LSTM, encoder, decoder, reduced order model, Bayesian inference

1 Introduction

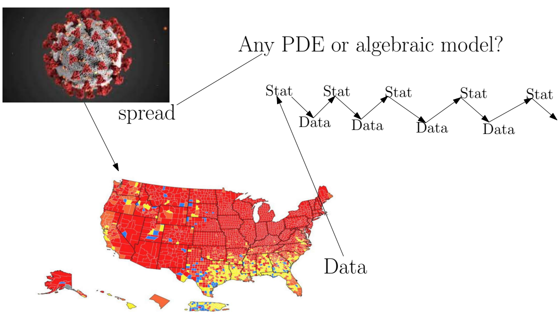

Fig. 1: We will potentially keep hovering between data and statistics if a robust forward model is not generated

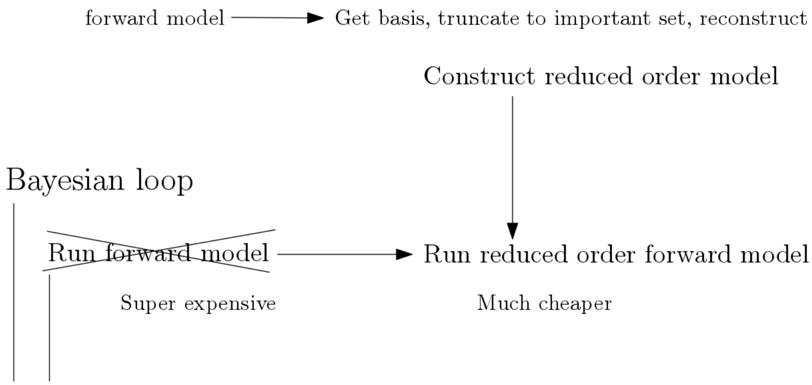

Fig. 2: The big picture

Any forward model in computational physics that comes with uncertainty in one or more model parameter values has to pass through an inference framework to narrow down the range of possible values for that parameter. In this day and age, with the preponderance of available data, and no model (differential/algebraic) as such to govern the underlying physics for quite a few problems (COVID-19 for instance, Fig. 1), it boils down to patching together a model with a combination of differential and algebraic operators, and then estimating a good range for the coefficients. On the other hand, there are ways in which no such effort is put in, and the data is processed through a statistical blackbox and the deliverables are delivered. In the realm of computational physics and finance as well, it makes sense to go about doing the former, as that effort will lend more intuition in the field for the future. The thing is though, the inference framework if Bayesian, requires multiple passes of such a forward model, and if the forward model being constructed is sophisticated, then a reduced order model is needed, as shown in Fig. 2. Now, the math behind reduced order modeling can be complicated, so recourse is taken to the burgeoning field of deep learning to help us construct these reduced order models. The question is though, how well does the Bayesian inference perform with a reduced order model as opposed to a full model? We examine that question in this work by following these steps

- ✓ We simulate the forward model
- ✓ We add noise to the forward model simulation output
- ✓ We run the noisy high fidelity model output in the Bayesian inference framework
- ✓ We generate inference estimates
- ✓ We construct a reduced order model using deep learning with the noisy data as target
- ✓ We run the reduced order model in the Bayesian framework
- ✓ We generate inference estimates again
- ✓ We compare the inference estimates of Bayesian with full model against Bayesian with reduced order model

1.1 Bayesian inference

The Bayesian inference framework works on the basic tent of uncovering a distribution centered around the true value and starts off with an initial guess for the distribution also called "prior" to eventually get to the most accurate distribution possible also called "posterior" [1, 2]. The Bayes theorem in a nutshell is:

$$Posterior = \frac{Prior \times Likelihood}{\int Prior \times Likelikehood} \tag{1}$$

The prior is typically taken to be a Gaussian distribution and the likelihood carries information about the forward model. The denominator at the bottom requires numerical computation, and standard quadrature based techniques become incredibly expensive for sophisticated forward models. Monte Carlo sampling is invoked in the Markovian realm where each sample is related to the previous sample in a Markovian chain way. The Markov chain Monte Carlo (MCMC) sampling is achieved using the adaptive metropolis algorithm [3], which requires forward model evaluation for every sample that is accepted by the algorithm. A code snippet is provided in Listing 1.

```
class MCMC:
    """Class for MCMC sampling"""
    .
        def sample(self):
        # function for sampling using adaptive Metropolis algorithm
        # return: accepted samples
        .
        for isample in range(0,self.nsamples):
            q_new = np.reshape(np.random.multivariate_normal(qparams[:,-1],Vold),(-1,1))
            accept,SSqnew=self.acceptreject(q_new,SSqprev,self.std2[-1])
            .
```

Listing 1: Adaptive metropolis code snippet

1.2 Deep learning based reduced order model

The full model generates an acceleration time series for each value of a critical slip distance parameter of the forward model as shown in Appendix A and plotted in Fig. 3 for one value of model parameter. A reduced order model would effectively mean reconstructing this time series using LSTM encoder decoder network [4] as shown in Fig. 4, and the optimal deep learning parameters to best reconstruct the time series. The deep learning piece is built on the PyTorch framework, and all simulations are run on a basic AMD Ryzen 3 3200U with Radeon Vega Mobile Gfx $\times$ 4 processor. A code snippet is provided in Listing 2. We use a batch size of 20, hidden state size of 10, number of LSTM layers per encoder and decoder is 1, and we deploy the Adam optimizer. Now, the question remains as to whether such a reconstructed time

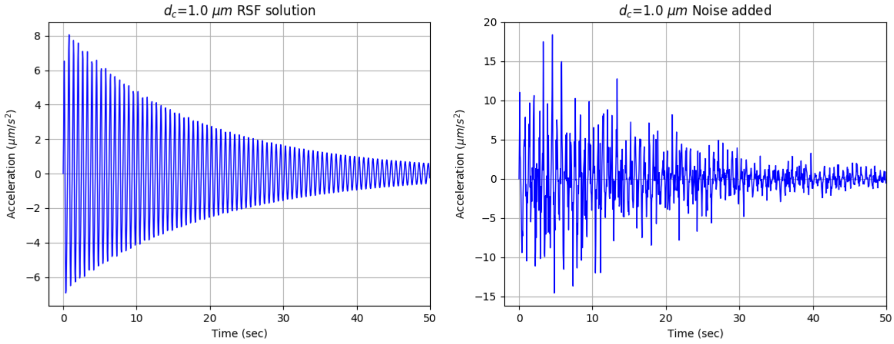

Fig. 3: System response

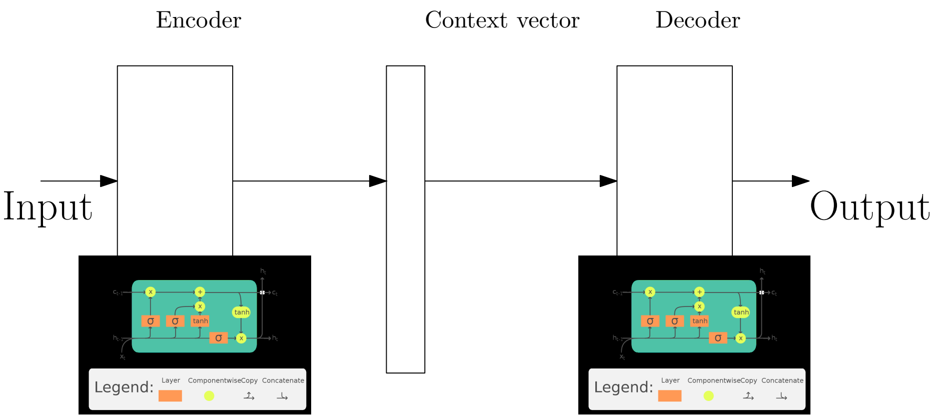

Fig. 4: LSTM encoder decoder

series would work well in the Bayesian inference framework? Have we lost too many features in the process? We delve into these things in this work, as we mentioned in the Introduction.

```
class lstm_encoder(nn.Module):
    # Encodes time-series sequence
    def __init__(self, input_size, hidden_size, num_layers = 1):
        super(lstm_encoder, self).__init__()
        self.lstm = nn.LSTM(input_size = input_size, hidden_size= hidden_size, num_layers = num_layers)

    def forward(self, x_input): # called internally by PyTorch
        lstm_out, self.hidden = self.lstm(x_input.view(x_input.shape[0], x_input.shape[1], self.input_size))
        return lstm_out, self.hidden

class lstm_decoder(nn.Module):
    # Decodes hidden state output by encoder
```

```
    def __init__(self, input_size, hidden_size, num_layers = 1):
        super(lstm_decoder, self).__init__()
        self.lstm = nn.LSTM(input_size = input_size, hidden_size = hidden_size, num_layers = num_layers)
        self.linear = nn.Linear(hidden_size, input_size)

    def forward(self, x_input, encoder_hidden_states): # called internally by PyTorch
        lstm_out, self.hidden = self.lstm(x_input.unsqueeze(0), encoder_hidden_states)
        output = self.linear(lstm_out.squeeze(0))
        return output, self.hidden
```

Listing 2: LSTM autoencoder code snippet

2 Numerical simulations

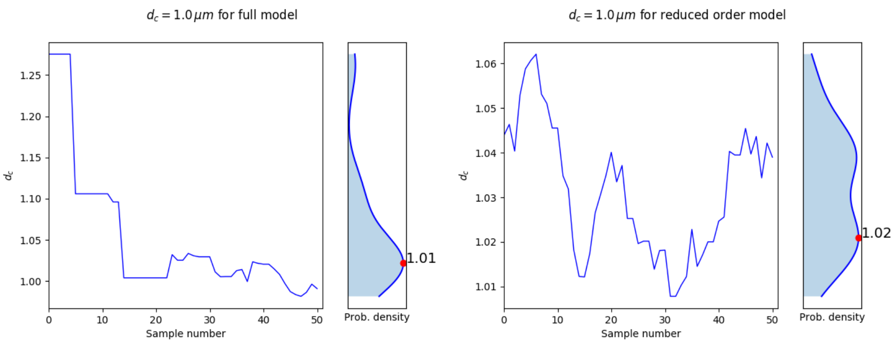

Fig. 5: Noisy data of Fig. 3. Comparison of Bayesian/MCMC inference for full and DL based reduced order models for 100 samples with half the samples burnt-in (as is usual practice in Bayesian/MCMC)

We see in Fig. 5 that the inference for the deep learning reconstructed time series compares well with that for the original time series. It is important to note that the Monte Carlo concept is meant to sample randomly with the Markovian concept ensuring that the samples follow a certain condition, so any reconstruction which captures important features of the original solution would do a good job in this Bayesian/MCMC framework. We add more noise to the RSF solution as shown in Fig. 6 to check the efficacy of the deep learning reconstruction. We observe from Figs .7 - 9 that with increase in the number of samples, the DL reconstructed time series also does fairly well in the Bayesian/MCMC framework. At this point, it is important to point what these "features" of the time series are after all. If you look at Fig. 3, you will observe that the RSF solution is a trignometric function with exponentially decreasing amplitude passed through a spring slider damper which does not

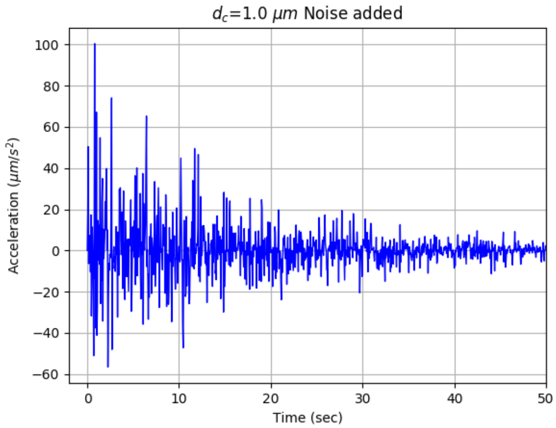

Fig. 6: More noise added to RSF solution

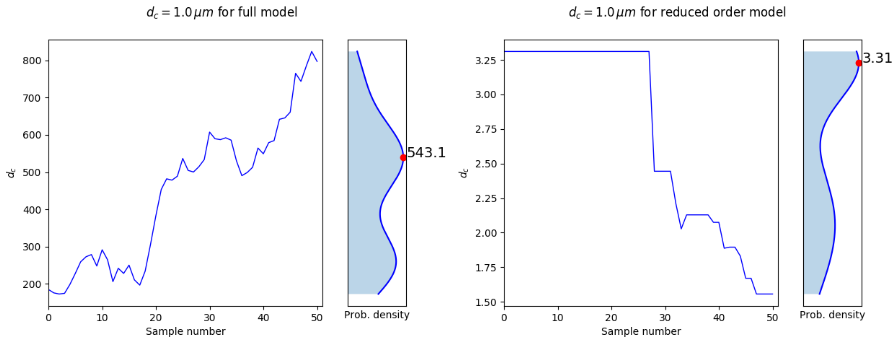

Fig. 7: Noisy data of Fig. 6. Comparison of Bayesian/MCMC inference for full and DL based reduced order models for 100 samples with half the samples burnt-in (as is usual practice in Bayesian/MCMC)

distort the signal too much. The "features" would then be the sinusoidal wave, exponentially decreasing amplitude, and solution dying down to a whimper after a certain time. In real life problems, there would be numerous features which would characterize the solution, and that is where the efficacy of the DL reconstructed time series would be tested to a good amount. As it stand right now, the proof of concept works well, and offers a lot of promise in the realm of constructing forward models from data.

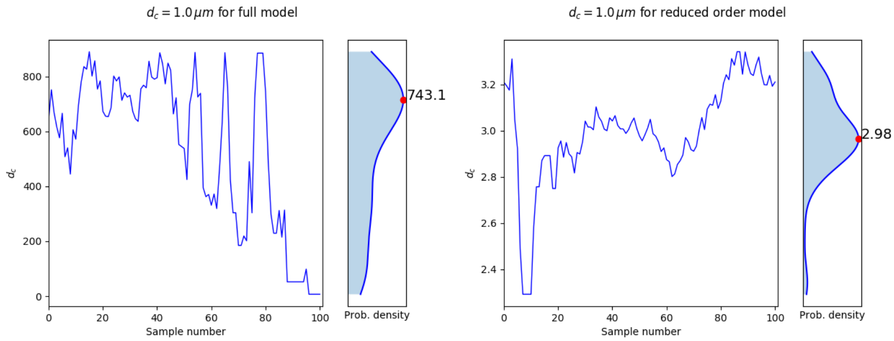

Fig. 8: Noisy data of Fig. 6. Comparison of Bayesian/MCMC inference for full and DL based reduced order models for 200 samples with half the samples burnt-in (as is usual practice in Bayesian/MCMC)

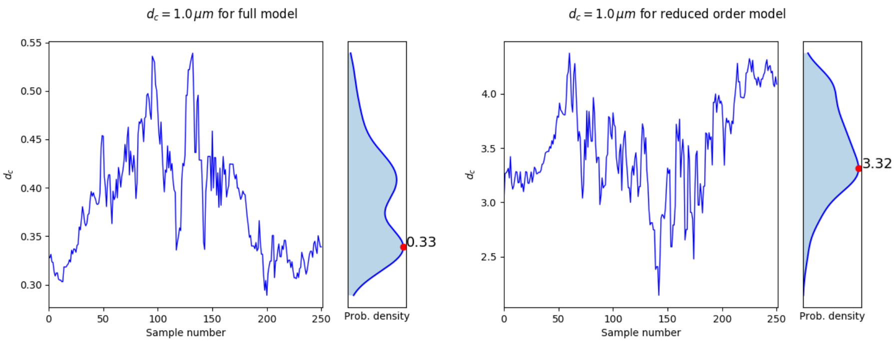

Fig. 9: Noisy data of Fig. 6. Comparison of Bayesian/MCMC inference for full and DL based reduced order models for 500 samples with half the samples burnt-in (as is usual practice in Bayesian/MCMC)

Appendix A Forward model details

The quantification of fault slip is achieved using the rate- and state-friction (RSF) model for modeling earthquake cycles on faults [5–10], and given by

$$\begin{aligned} \mu &= \mu_0 + A \ln\left(\frac{V}{V_0}\right) + B \ln\left(\frac{V_0 \theta}{d_c}\right), \\ \dot{\theta} &= 1 - \frac{\theta V}{d_c} \end{aligned} \tag{A1}$$

where $V = |\frac{d\boldsymbol{d}}{dt}|$ is the slip rate magnitude, $a = \frac{dV}{dt}$ which we hypothesize is of the same order as recorded by seismograph, μ_0 is the steady-state friction coefficient at the reference slip rate V_0, A and B are empirical dimensionless constants, θ is the macroscopic variable characterizing state of the surface and d_c is a critical slip distance over which a fault loses or regains its frictional strength after a perturbation in the loading conditions [11]. The RSF model is a piece of the coupled flow and geomechanics puzzle [12–18]. We rewrite Eq. (A1) as

$$\begin{aligned} V &= V_0 \exp\left(\tfrac{1}{A}\left(\mu - \mu_0 - B\ln\left(\tfrac{V_0\theta}{d_c}\right)\right)\right), \\ \dot{\theta} &= 1 - \tfrac{\theta V}{d_c} \end{aligned} \tag{A2}$$

As shown in Fig. A1, we model a fault by a slider spring system [19–21]. The friction coefficient of the block is given by

$$\mu = \frac{\tau}{\sigma} = \frac{\tau_l - k\delta - \eta V}{\sigma} \tag{A3}$$

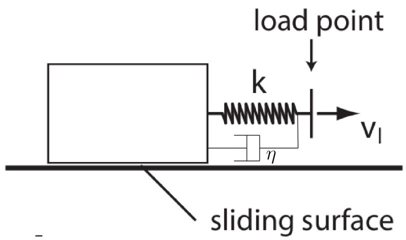

Fig. A1: Spring slider damper idealization of fault behavior

where σ is the normal stress, τ the shear stress on the interface, τ_l is the remotely applied stress acting on the fault in the absence of slip, -$k\delta$ is the stress relaxation due to fault slip [22] and η is the radiation damping coefficient [23]. We consider the case of a constant stressing rate $\dot{\tau}_l = kV_l$ where V_l is the load point velocity. The stiffness is a function of the fault length l and elastic modulus E as $k \approx \frac{E}{l}$. With $k' = \frac{E}{l\sigma}$, we get

$$\begin{aligned} \dot{\mu} &\approx k'(V_l - V) - k''\dot{V} \\ &= k'\left(V_l - V_0 \exp\left(\frac{1}{A}\left(\mu - \mu_0 - B\ln\left(\frac{V_0\theta}{d_c}\right)\right)\right)\right) \\ &\quad - k''V_0 \overbrace{\exp\left(\frac{1}{A}\left(\mu - \mu_0 - B\ln\left(\frac{V_0\theta}{d_c}\right)\right)\right)}^{\bullet} \end{aligned} \tag{A4}$$

where $k'' = \frac{\eta}{\sigma}$. Once the phenomenological form of $\dot{\mu}$ is known, we get the acceleration time series as $a \equiv \dot{V}$ as,

$$\dot{V} = \frac{V}{A}\left(\dot{\mu} - \frac{B}{\theta}\dot{\theta}\right)$$

The ballpark values are:

- ✓ Elastic modulus $E = 5 \times 10^{10}\, Pa$
- ✓ Critical fault length $l = 3 \times 10^{-2}\, m$
- ✓ Normal stress $\sigma = 200 \times 10^6 Pa$
- ✓ Radiation damping coefficient $\eta = 20 \times 10^6 Pa/(m/s)$
- ✓ $A = 0.011$
- ✓ $B = 0.014$
- ✓ $V_0 = 1\mu m/s$

- ✓ $\theta_0 = 0.6$
- ✓ $\mu_0 = \mu_0 = 0.6$
- ✓ $t_{start} = 0,\, t_{end} = 50\,s,\, dt = 0.05\,s$

from which the effective stiffness and damping are obtained as

- ✓ $k' = \frac{E}{l\sigma} = \frac{5\times10^{10}}{3\times10^{-2}\times2\times10^{8}}[1/m] \approx 10^{-2}[1/\mu m]$
- ✓ $k'' = \frac{\eta}{\sigma} = \frac{2\times10^{7}}{2\times10^{8}}[s/m] = 10^{-7}[s/\mu m]$

The influence of critical slip distance on system response to a load point perturbation of the form

$$V_l = V_0(1 + \exp\left(-t/20\right) sin(10t)) \tag{A5}$$

is shown in Fig. 3.

References

[1] Dana, S., Reddy, K.: Bayesian inference and markov chain monte carlo based estimation of a geoscience model parameter. arXiv preprint arXiv:2201.01868 (2021)

[2] Dana, S., Lyathakula, K.R.: Uncertainty quantification in friction model for earthquakes using bayesian inference. arXiv preprint arXiv:2104.11156 (2021)

[3] Haario, H., Saksman, E., Tamminen, J.: An adaptive metropolis algorithm. Bernoulli, 223–242 (2001)

[4] Sak, H., Senior, A., Beaufays, F.: Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. arXiv preprint arXiv:1402.1128 (2014)

[5] Ruina, A.L.: Slip instability and state variable friction laws. Geophys. Res. Lett. **88**, 359–370 (1983)

[6] Scholz, C.H.: Mechanics of faulting. Ann. Rev. Earth Planet. Sci. **17**, 309–334 (1989)

[7] Marone, C.: Laboratory-derived friction laws and their application to seismic faulting. Ann. Rev. Earth Planet. Sci. **26**, 643–696 (1998)

[8] Wei, M., Shi, P.: Synchronization of earthquake cycles of adjacent segments on oceanic transform faults revealed by numerical simulation in the framework of rate-and-state friction. Journal of Geophysical Research: Solid Earth **126**(1), 2020–020231 (2021)

[9] Jia, Y., Tang, J., Lu, Y., Lu, Z.: The effect of fluid pressure on frictional stability transition from velocity strengthening to velocity weakening

and critical slip distance evolution in shale reservoirs. Geomechanics and Geophysics for Geo-Energy and Geo-Resources **7**(1), 1–13 (2021)

[10] Weiqiang, Z., Allison, K., Dunham, E., Yang, Y.: Fault valving and pore pressure evolution in simulations of earthquake sequences and aseismic slip. Nature communications **11**, 4833 (2020)

[11] Palmer, A.C., Rice, J.R.: The growth of slip surfaces in the progressive failure of over-consolidated clay. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences **332**(1591), 527–548 (1973)

[12] Dana, S., Wheeler, M.F.: Convergence analysis of fixed stress split iterative scheme for anisotropic poroelasticity with tensor biot parameter. Computational Geosciences **22**(5), 1219–1230 (2018)

[13] Dana, S., Wheeler, M.F.: Convergence analysis of two-grid fixed stress split iterative scheme for coupled flow and deformation in heterogeneous poroelastic media. Computer Methods in Applied Mechanics and Engineering **341**, 788–806 (2018)

[14] Dana, S., Ganis, B., Wheeler, M.F.: A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs. Journal of Computational Physics **352**, 1–22 (2018)

[15] Dana, S., Srinivasan, S., Karra, S., Makedonska, N., Hyman, J.D., O'Malley, D., Viswanathan, H., Srinivasan, G.: Towards real-time forecasting of natural gas production by harnessing graph theory for stochastic discrete fracture networks. Journal of Petroleum Science and Engineering **195**, 107791 (2020)

[16] Dana, S., Jammoul, M., Wheeler, M.F.: Performance studies of the fixed stress split algorithm for immiscible two-phase flow coupled with linear poromechanics. Computational Geosciences, 1–15 (2021)

[17] Dana, S., Ita, J., Wheeler, M.F.: The correspondence between voigt and reuss bounds and the decoupling constraint in a two-grid staggered algorithm for consolidation in heterogeneous porous media. Multiscale Modeling & Simulation **18**(1), 221–239 (2020)

[18] Dana, S.: Addressing challenges in modeling of coupled flow and poromechanics in deep subsurface reservoirs. PhD thesis, The University of Texas at Austin (2018)

[19] Rice, J.R., Gu, J.-c.: Earthquake aftereffects and triggered seismic phenomena. Pure and Applied Geophysics **121**(2), 187–219 (1983)

[20] Gu, J.-C., Rice, J.R., Ruina, A.L., Simon, T.T.: Slip motion and stability of a single degree of freedom elastic system with rate and state dependent friction. Journal of the Mechanics and Physics of Solids **32**(3), 167–196 (1984)

[21] Dieterich, J.H.: Earthquake nucleation on faults with rate-and state-dependent strength. Tectonophysics **211**(1-4), 115–134 (1992)

[22] Kanamori, H., Brodsky, E.E.: The physics of earthquakes. Reports on Progress in Physics **67**(8), 1429 (2004)

[23] McClure, M.W., Horne, R.N.: Investigation of injection-induced seismicity using a coupled fluid flow and rate/state friction model. Geophysics **76**(6), 181–198 (2011)