Reconstructing the ground displacement time series for injection induced geomechanical activity using LSTM autoencoders

Saumik Dana[a]

[a] *University of Southern California,*

Abstract

We use LSTM autoencoders to reconstruct the displacement time series for grid points on the top surface of a faulting due to water injection problem. The reason is that we aim to use this LSTM autoencoder based model instead of the high fidelity model in the Bayesian inference framework to estimate injection rate from displacement input.

Keywords: LSTM, autoencoder, reconstruction, time series, ground displacement, injection rate, fault, coupled flow and geomechanics

1. Introduction

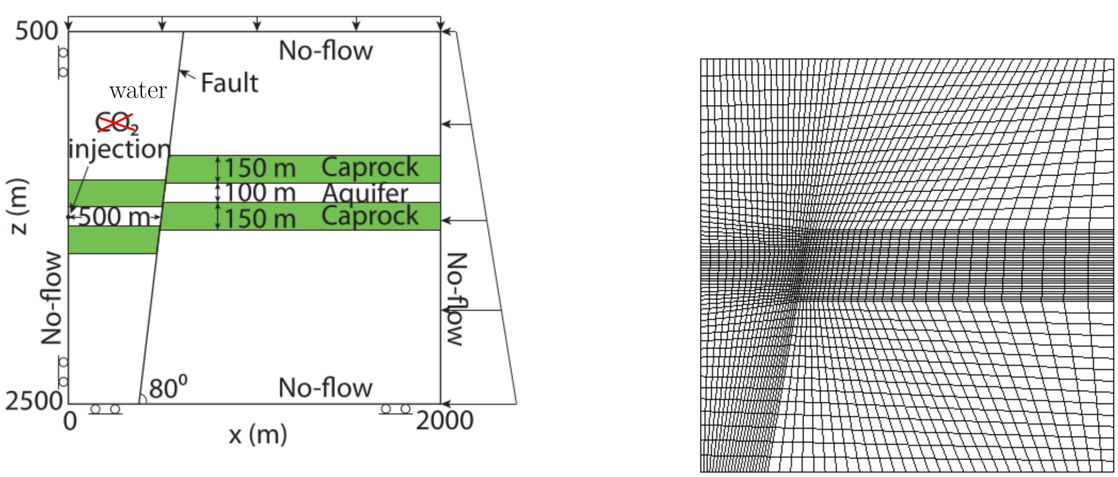

Figure 1: Model of the water injection plane strain case (adapted from [1]) and the mesh

In coupled flow and geomechanics problems [2–10], the most important aspect is the estimation of subsurface properties, either from InSAR data or geophones/seismograms on the surface or from sensors at the well(s). For estimation from surface data, we need to run an inference framework, and Bayesian is robust. To run Bayesian though, we need multiple (sometimes millions of) forward model runs, which will become infeasible if the model is sophisticated. Hence, reduced order models are needed, and we deploy LSTM autoencoders for that purpose. In this work, we go from the forward model to the simulations to the time series data for the top surface data, and then LSTM autoencoder based reconstruction

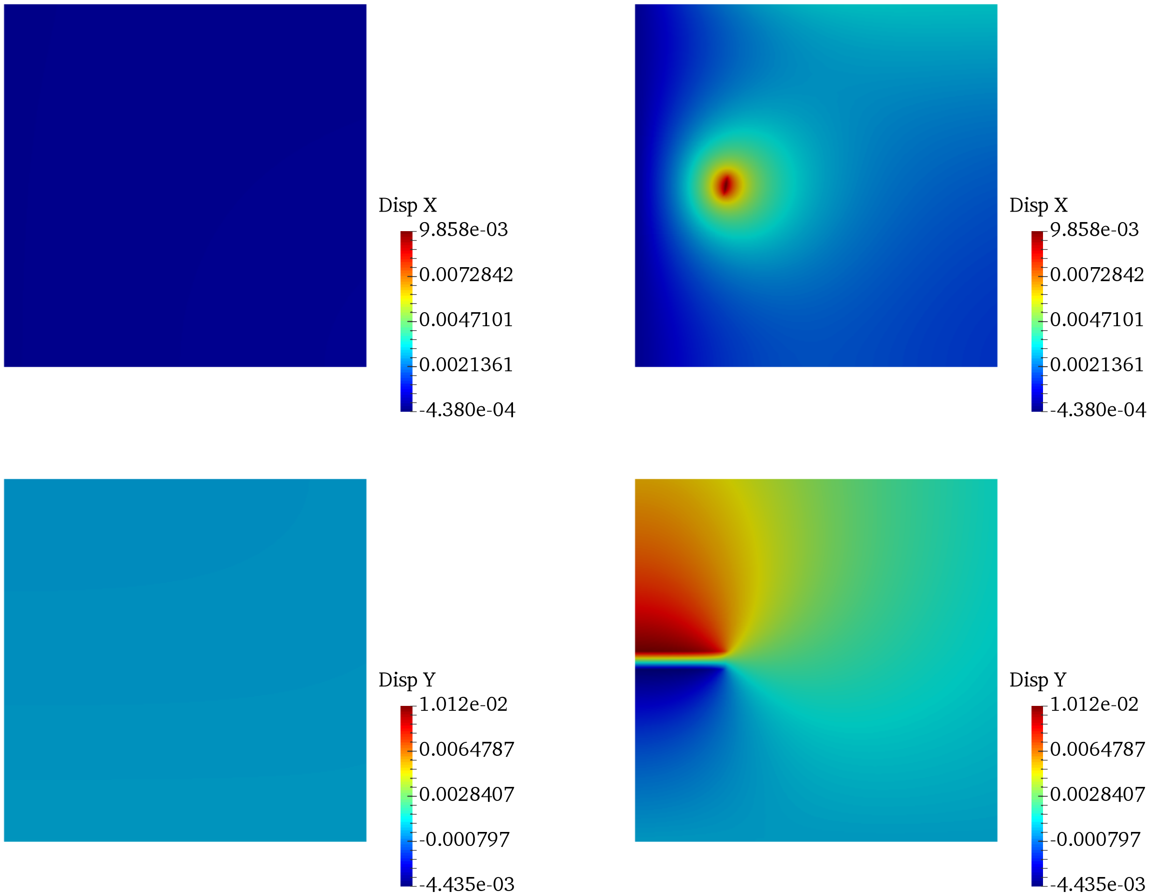

Figure 2: Snapshots of x- and y- displacement at t=0 and t=60 days respectively for an injection rate of 100 MSCF/day

for a archetypal faulting due to injection problem. This document is structured as follows: the forward model is given in the Appendix, Section 2 has the LSTM autoencoder based reconstruction.

We consider a two-dimensional plane-strain model with the fault under normal faulting conditions, that is, the vertical principal stress due to gravity is the largest among the three principal stresses (Fig. 1). The initial fluid pressure at $500\,m$ depth is $5\,MPa$ and $24.63\,MPa$ at $2500\,m$, considering a hydrostatic gradient ($9.81\,MPa/km$) and an atmospheric pressure of $0.1\,MPa$ at the ground surface. The rock density is $2260\,kg/m^3$, so the lithostatic gradient is $22.17\,MPa/km$. Assuming a porosity of 0.1, the initial vertical stress is $11.085 \times 0.9 + 5 \times 0.1 = 10.4765\,MPa$ at $500\,m$, and $22.17 \times 2.5 \times 0.9 + 24.63 \times 0.1 = 52.3455\,MPa$ at $2500\,m$. We choose a value of 0.7 for the ratio of horizontal to vertical initial total stress. The average bulk density is $\rho_b = 2260 \times 0.9 + 1000 \times 0.1 = 2134 kg/m^3$, the average sonic compressional and shear velocities are $V_p = 730$ m/s and $V_s = 420$ m/s respectively. The Biot coefficient is assumed to be $b = 1.0$. The friction coefficient drops linearly from static friction $\mu_s = 0.5$ to dynamic friction $\mu_d = 0.2$ over $d_c = 5\,mm$. Snapshots of the displacement evolution are given in Fig. 2.

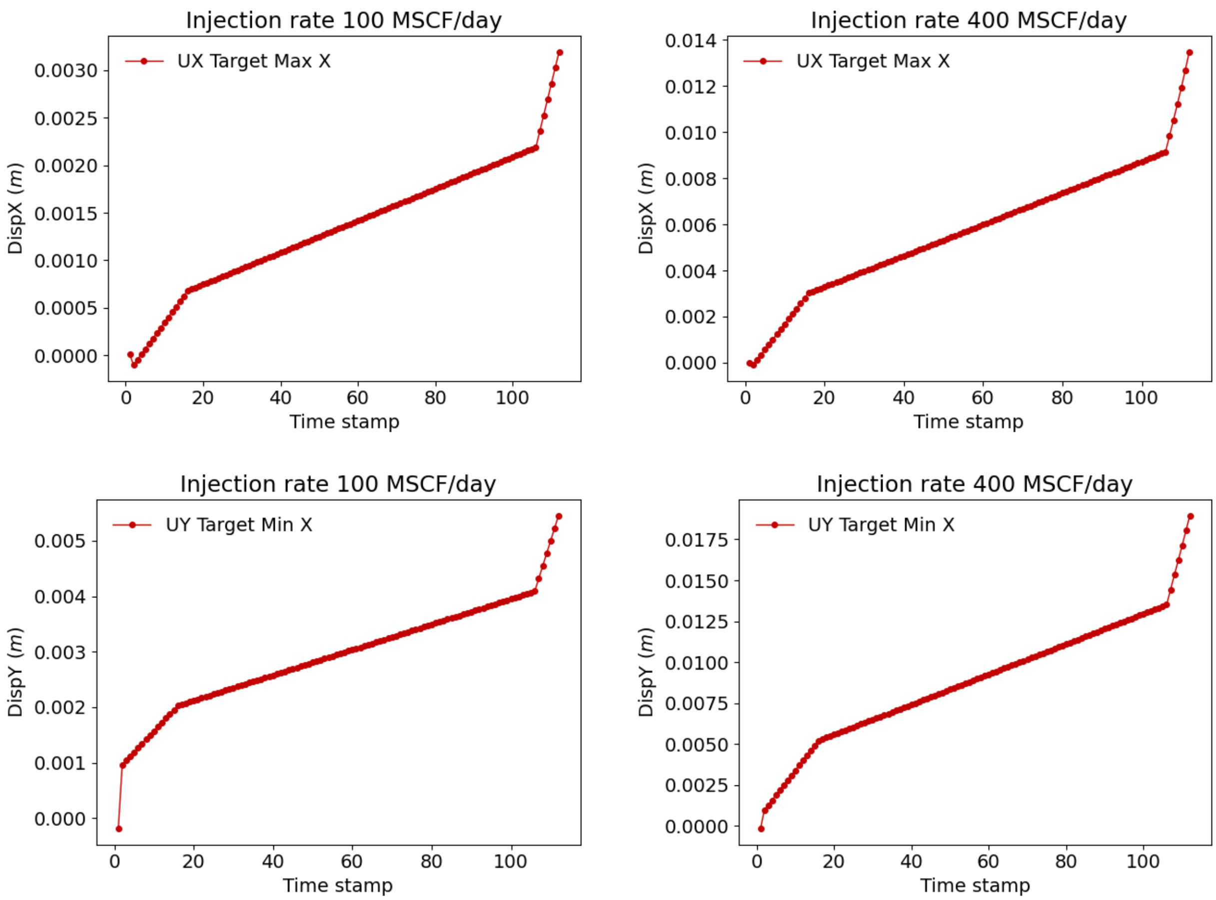

Figure 3: Displacement time series ground truth for u_x at $(2000, 500)$ and u_y at $(0, 500)$ for injection rates on either end of the data spectrum

The simulator spits out vtk files for each time stamp in the simulation. Our job is to extract the displacement values corresponding to grid points on the top surface, and construct a time series as the ground truth, as shown in Fig. 3.

2. Reconstruction using LSTM autoencoders

The full model generates a displacement time series for each value of injection rate as explained in Appendix A. A reduced order model would effectively mean reconstructing this time series using LSTM autoencoder, 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 1. We use a window size of 28 for 112 time steps, hidden state size same as window size, number of LSTM layers per encoder and decoder is 1, and we deploy the Adam optimizer. We observe from Fig. 4 that LSTM autoencoders are meant for oscillating time series, because they have this compulsive behavior to want to capture oscillations. Nevertheless, the mean behavior is captured correctly, and that is good enough for inference in the robust Bayesian framework. We have to keep in mind though,

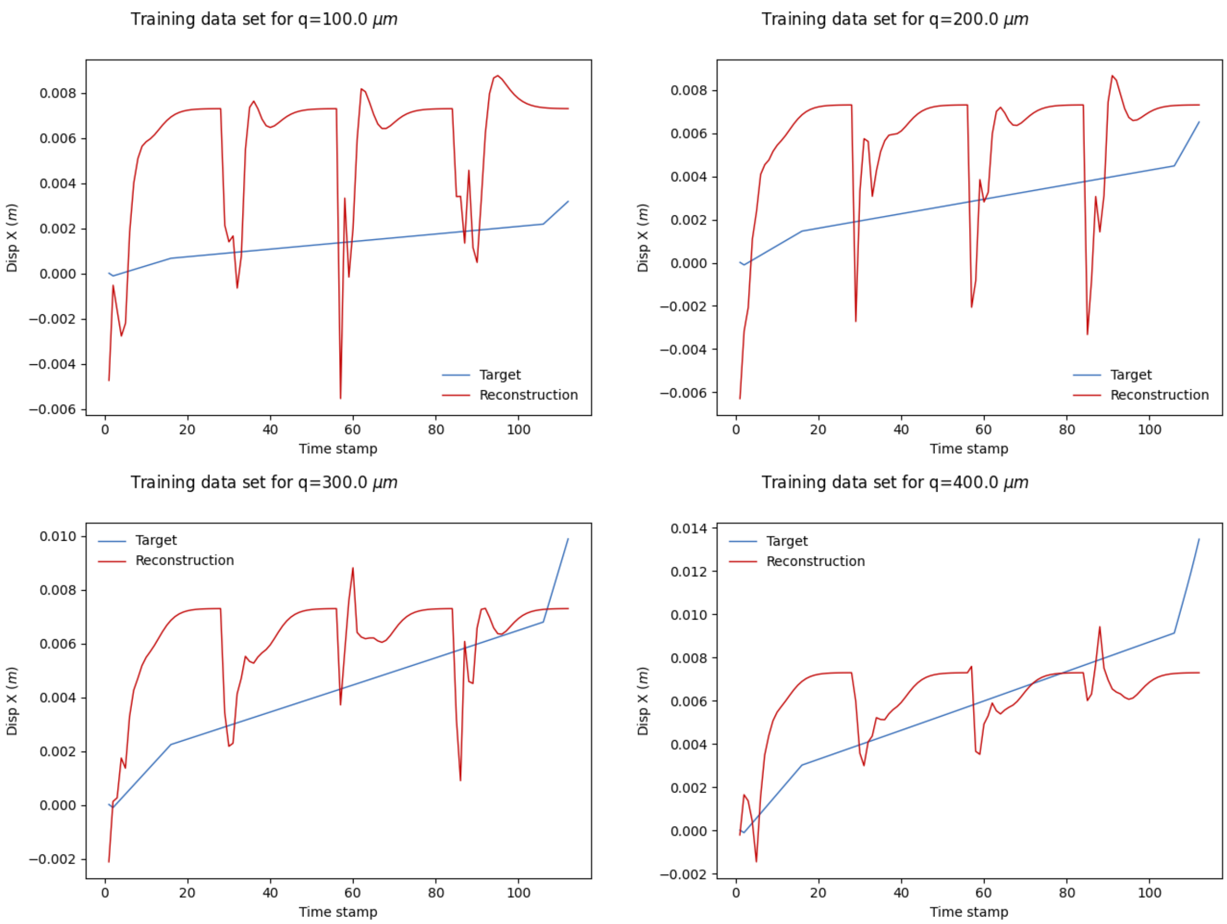

Figure 4: Reconstruction of displacement time series ground truth for u_x at $(2000, 500)$ for different injection rates

that this is a toy problem, and the ground displacement time series due to seismic activity is typically always oscillatory, as is evident from seismogram data. So we stick to this deep learning architecture for problems in this realm going forward.

```
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 1: LSTM autoencoder code snippet

Appendix A. Forward model

The governing PDE for displacement $\mathbf{u} : \Omega \times [0, T] \to \mathbb{R}^3$ is the linear momentum balance given by

$$\nabla \cdot \boldsymbol{\sigma} + \rho_b \mathbf{g} = \mathbf{0} \tag{A.1}$$

with the constitutive laws relating poroelastic stress tensor $\boldsymbol{\sigma}$, effective stress tensor $\boldsymbol{\sigma}'$, strain tensor $\boldsymbol{\epsilon}$, volumetric strain $\epsilon = tr(\boldsymbol{\epsilon})$ and pore pressure p given by

$$\begin{aligned} \boldsymbol{\sigma} &= \boldsymbol{\sigma}' - bp\mathbf{I} \\ \boldsymbol{\sigma}' &= \lambda \epsilon \mathbf{I} + 2G\boldsymbol{\epsilon} = \boldsymbol{D}\boldsymbol{\epsilon} \end{aligned} \tag{A.2}$$

with boundary and initial conditions given by

$$\begin{aligned} \mathbf{u} = \overline{\mathbf{u}} \text{ on } \Gamma_D,\ \ \dot{\mathbf{u}} = \dot{\overline{\mathbf{u}}} \text{ on } \Gamma_D,\ \ \boldsymbol{\sigma}^T \mathbf{n} = \overline{\mathbf{t}} \text{ on } \Gamma_N \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}),\ \ \dot{\mathbf{u}}(\mathbf{x}, 0) = \dot{\mathbf{u}}_0(\mathbf{x}) \end{aligned}$$

As shown in Fig. A.5, slip on the fault is the displacement of the positive side relative to the negative side:

$$(\boldsymbol{u}_+ - \boldsymbol{u}_-) - \boldsymbol{d} = \mathbf{0} \text{ on } \Gamma_f, \tag{A.3}$$

Recognizing that fault tractions are analogous to the boundary tractions, we add in the contributions from integrating the Lagrange multipliers $\boldsymbol{l} \equiv \boldsymbol{\sigma}' \boldsymbol{n}$ over the fault surface in the conventional finite element formulation to get

$$\begin{aligned} &\int_\Omega \nabla \boldsymbol{\eta} : (\boldsymbol{\sigma}' - bp\mathbf{I})\, d\Omega - \int_\Omega \boldsymbol{\eta} \cdot \rho_b \boldsymbol{g}\, d\Omega - \int_{\Gamma_N} \boldsymbol{\eta} \cdot \bar{\boldsymbol{t}}\, d\Gamma \\ &+ \int_{\Gamma_{f_+}} \boldsymbol{\eta} \cdot (\boldsymbol{l} - bp_+ \boldsymbol{n})\, d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta} \cdot (\boldsymbol{l} - bp_- \boldsymbol{n})\, d\Gamma = 0 \end{aligned} \tag{A.4}$$

We use the Mohr-Coulomb theory [11] to define the stability criterion for the fault and

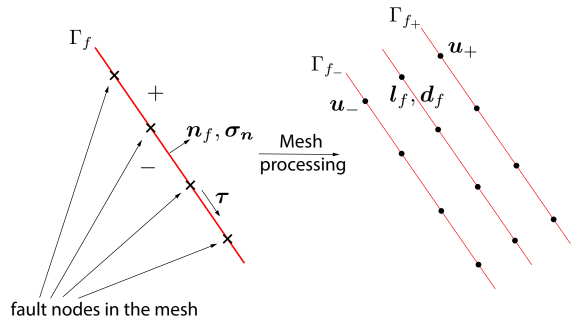

Figure A.5: A fault surface Γ_f is processed to create three surfaces: the positive side surface Γ_{f+} containing $\boldsymbol{u}_+$, the negative side surface Γ_{f-} containing $\boldsymbol{u}_-$, and the slip surface containing the fault effective traction vector $\boldsymbol{l}$ and the slip vector $\boldsymbol{d}$

define a fault pressure $p_f = \frac{p_- + p_+}{2}$. The shear stress and frictional stresses on the fault are

$$\tau = |\boldsymbol{l} - \sigma_n' \boldsymbol{n}| \equiv |\boldsymbol{l} - (\boldsymbol{l} \cdot \boldsymbol{n})\boldsymbol{n}|$$
$$\tau_f = \begin{cases} \tau_c - \mu_f \boldsymbol{l} \cdot \mathbf{n}, & \boldsymbol{l} \cdot \mathbf{n} < 0, \\ \tau_c, & \boldsymbol{l} \cdot \mathbf{n} \geq 0 \end{cases}$$

where τ_c is the cohesive strength of the fault, μ_f is the coefficient of friction which evolves as

$$\mu_f = \begin{cases} \mu_s - (\mu_s - \mu_d)\frac{|\boldsymbol{d}|}{d_c}, & |\boldsymbol{d}| \leq d_c, \\ \mu_d, & |\boldsymbol{d}| > d_c \end{cases} \tag{A.5}$$

where d_c is a critical slip distance. The fields are approximated as follows:

$$\boldsymbol{u} \approx \boldsymbol{u}_h = \sum_{b=1}^{n_{\text{node}}} \eta_b \boldsymbol{U}_b, \quad \boldsymbol{l} \approx \boldsymbol{l}_h = \sum_{b=1}^{n_{f,\text{node}}} \eta_b \boldsymbol{L}_b, \quad \boldsymbol{d} \approx \boldsymbol{d}_h = \sum_{b=1}^{n_{f,\text{node}}} \eta_b \boldsymbol{D}_b,$$

where n_{node} is the total number of nodes and $n_{f,\text{node}}$ is the number of Lagrange nodes. After substitution of the finite element approximations into the weak form of the problem, we obtain the fully discrete equations in residual form for all nodes a and lagrange nodes $\bar{a}$:

$$\begin{aligned} \boldsymbol{R}_{u,a}^{Stat} = & \int_\Omega \boldsymbol{B}_a^T (\boldsymbol{\sigma'}_h^{n+1} - b p_h^{n+1} \mathbf{1}) d\Omega - \int_\Omega \boldsymbol{\eta}_a^T \rho_{b,h}^{n+1} \boldsymbol{g} d\Omega - \int_{\Gamma_N} \boldsymbol{\eta}_a^T \bar{\boldsymbol{t}} d\Gamma \\ & + \int_{\Gamma_{f_+}} \boldsymbol{\eta}_a^T (\boldsymbol{l}_h^{n+1} - b p_{f,h}^{n+1} \boldsymbol{n}) d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}_a^T (\boldsymbol{l}_h^{n+1} - b p_{f,h}^{n+1} \boldsymbol{n}) d\Gamma = \mathbf{0} \end{aligned} \tag{A.6}$$

$$\boldsymbol{R}_{l,\bar{a}}^{Stat} = \int_{\Gamma_{f_+}} \boldsymbol{\eta}_{\bar{a}}^T \boldsymbol{u}_{h_+}^{n+1} d\Gamma - \int_{\Gamma_{f_-}} \boldsymbol{\eta}_{\bar{a}}^T \boldsymbol{u}_{h_-}^{n+1} \, d\Gamma - \int_{\Gamma_f} \boldsymbol{\eta}_{\bar{a}}^T \boldsymbol{d}_h^{n+1} d\Gamma = \mathbf{0} \tag{A.7}$$

We find the system Jacobian matrix by isolating the term for the increments in displacements and Lagrange multipliers at time step $n+1$. The system of linear equations is:

$$\left[\begin{array}{ccc:c} \boldsymbol{K}_{rr} & \boldsymbol{K}_{r+} & \boldsymbol{K}_{r-} & \mathbf{0} \\ \boldsymbol{K}_{+r} & \boldsymbol{K}_{++} & \mathbf{0} & \boldsymbol{C}_+^T \\ \boldsymbol{K}_{-r} & \mathbf{0} & \boldsymbol{K}_{--} & -\boldsymbol{C}_-^T \\ \hdashline \mathbf{0} & \boldsymbol{C}_+ & -\boldsymbol{C}_- & \mathbf{0} \end{array}\right] \left[\begin{array}{c} d\boldsymbol{U}_r \\ d\boldsymbol{U}_+ \\ d\boldsymbol{U}_- \\ \hline d\boldsymbol{L} \end{array}\right] = - \left[\begin{array}{c} \boldsymbol{R}_{u,r}^{Stat} \\ \boldsymbol{R}_{u,+}^{Stat} \\ \boldsymbol{R}_{u,-}^{Stat} \\ \hline \boldsymbol{R}_l^{Stat} \end{array}\right] \tag{A.8}$$

Algorithm 1: Time-marching in poroelastostatics

$n \leftarrow 0$; $t \leftarrow 0$;
$\boldsymbol{d}_h^0 \leftarrow \boldsymbol{0}$; // Initialize fault slip at virgin state
$\boldsymbol{u}_h^0 \leftarrow \boldsymbol{u}_h^{Prestep}$; // Initial condition based on a elastic prestep solve
while $t < T$ **do**
    /* time marching till final time */
    while *Not converged* **do**
        /* Staggered solution algorithm loop */
        Solve flow problem for pressures;
        $\boldsymbol{d}_h^{n+1} \leftarrow \boldsymbol{d}_h^n$; // Initialize fault slip for next time step
        Solve system of equations using GMRES; // Krylov subspace solver
        $\boldsymbol{L} \leftarrow \boldsymbol{L} + d\boldsymbol{L}$; // Update lagrange multipliers
        $\boldsymbol{U} \leftarrow \boldsymbol{U} + d\boldsymbol{U}$; // Update displacements
        for *Loop over lagrange nodes* **do**
            $\tau \leftarrow |\boldsymbol{l}_h^{n+1} - (\boldsymbol{l}_h^{n+1} \cdot \boldsymbol{n})\boldsymbol{n}|$; // Obtain shear stress on fault
            $\tau_f \leftarrow \tau_f|_{\boldsymbol{d}_h^n}$; // Compute fault friction based on the slip value at previous time step
            while $\tau > \tau_f$ **do**
                /* Satisfy fault constitutive law */
                $\boldsymbol{U}_+ \leftarrow \boldsymbol{U}_+ - \boldsymbol{K}_{++}^{-1}\frac{\tau-\tau_f}{\tau}\boldsymbol{L}$;
                $\boldsymbol{U}_- \leftarrow \boldsymbol{U}_- + \boldsymbol{K}_{--}^{-1}\frac{\tau-\tau_f}{\tau}\boldsymbol{L}$;
                $\boldsymbol{d}_h^{n+1} \leftarrow \boldsymbol{u}_{h_+}^{n+1} - \boldsymbol{u}_{h_-}^{n+1}$; // Update fault slip
                $\tau_f \leftarrow \tau_f|_{\boldsymbol{d}_h^{n+1}}$; // Update fault friction
            $\boldsymbol{d}_h^{n+1} \leftarrow \boldsymbol{u}_{h_+}^{n+1} - \boldsymbol{u}_{h_-}^{n+1}$; // Update fault slip
    $n \leftarrow n + 1$;
    $t \leftarrow t + \Delta t$;

where the top row corresponds to displacement nodes excluding the fault positive and negative side nodes. In many quasi-static simulations it is convenient to compute a static problem with elastic deformation prior to computing a transient response. The heavy lifting for the time marching is done at the Krylov solver [12] stage to solve the system of Eqs. (A.8) as shown in Algorithm 1.

References

[1] F. Cappa, J. Rutqvist, Impact of CO_2 geological sequestration on the nucleation of earthquakes, Geophys. Res. Lett. 38 (2011) L17313.

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

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

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

[5] S. Dana, M. F. Wheeler, 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 (2018) 788–806.

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

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

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

[9] S. Dana, J. Ita, M. F. Wheeler, 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 (2020) 221–239.

[10] S. Dana, Addressing challenges in modeling of coupled flow and poromechanics in deep subsurface reservoirs, Ph.D. thesis, The University of Texas at Austin, 2018.

[11] J. C. Jaeger, N. G. W. Cook, Fundamentals of Rock Mechanics, Chapman and Hall, London, 1979.

[12] H. A. Van Der Vorst, Krylov subspace iteration, Computing in science & engineering 2 (2000) 32–37.