1. Quasi-static crustal deformation

When a region is subjected to a gradual loading process, such as tectonic stress changes, viscoelastic relaxation, and pore pressure changes, it deforms in a quasi-static manner. Every snapshot of a quasi-static process, as opposed to a dynamic process, satisfies stress equilibrium. The inertial force is considered negligible, since the net force is small, and the time scale is large.

For linear constitutive law and small strain problems, the absolute and incremental versions of the linear equation, Smith and Griffiths, 2004.

\[
K_n U_n = f_n, \text{ absolute. } K_n \Delta U_n = \Delta F_n, \text{ incremental.}
\]  
(1)

where, \( K \) is the system stiffness matrix, \( U \) is the solution vector and \( F \) is the nodal force, including fluid source. The subscript \( n \) is the time index. In this study, we use the incremental equation. The solution space \( \Delta U_n \) of a poroelastic problem contains the nodal displacement and pressure, \( \Delta U_n = \begin{bmatrix} \Delta u_n \\ \Delta p_n \end{bmatrix} \), whereas the solution for an elastic problem is only the nodal displacement.

The stiffness matrix \( K_n \) and RHS function \( \Delta F_n \) are also different for the elastic and poroelastic problems, Eq. (2).

\[
(K_n, \Delta F_n) = \begin{bmatrix} K_e & \Delta f_e \\ -H^T & \Delta K_e + S_p \end{bmatrix} \begin{bmatrix} \Delta u_e \\ \Delta p \end{bmatrix}, \text{ elastic.}
\]
\[
(K_n, \Delta F_n) = \begin{bmatrix} K_e & \Delta f_e \\ -H^T & \Delta K_e + S_p \end{bmatrix} \begin{bmatrix} \Delta u_e \\ \Delta p \end{bmatrix}, \text{ poroelastic.}
\]

where, \( K_e \) is the elastic stiffness matrix, depending on the elastic constants of the solid. \( K_n \) is the fluid stiffness matrix, depending on the fluid flow conductivity. \( H \) is the coupling matrix, depending on the Biot's coefficient. \( S_p \) is the storage matrix, depending the solid compressibility and porosity, and the fluid compressibility. Smith and Griffiths (2004) provide the detailed formulation for these matrices and vectors. Note, the stiffness matrix \( K_n \) is constant for evenly spaced time step \( \Delta t \). In a later section, we show that for Newtonian viscoelasticity, \( K_n \) is, although modified, still independent of time. \( f_n \) and \( q_n \) are nodal force and fluid source respectively. The detailed formulations of these matrices and RHS vectors are given in Appendix A.

2. Poroelastic model and benchmark

Unstable pressure is caused by using linear elements, known as the Ladyzenskaja-Babuska-Brezzi restrictions. The local pressure projection scheme, Bochev and Dohrmann (2006), is implemented to stabilize the pore pressure,

\[
F_{e+1} = F_{e+1} - \int_\Omega (N - (1/n_e)I_n)^T (N - (1/n_e)I_n)(2G)d\Omega,
\]

where

\[
F_{e+1} = \begin{bmatrix} 0 \\ H_i p_i \end{bmatrix}, \quad K_{e+1} = K_{e+1} + \begin{bmatrix} 0 & 0 \\ 0 & H_j \end{bmatrix}, \quad \text{where}
\]

References are provided in the text.
N is the shape function of an element, $n_e$ is the node count of the element, and $G$ is the shear modulus.

To benchmark the poroelastic models, we compare the Deform results against the well known Mandel solution. Fig. 1 illustrates the Mandel problem.

At time $t=0$, the boundaries at $r = \pm a$ of a homogeneously pressurized poroelastic medium have a pressure dropped from $p_0$ to zero. And, a compressional loading $\sigma_z = p_0$ is placed on a rigid plate at the matrix top. The pore fluid will then flow towards the side boundaries accompanied by matrix deformation. Because the matrix is axially symmetric with rigid top and bottom, the pressure, deformation and stress only vary in $r$ and time, not in $z$. These variables have closed forms in the Laplace frequency domain, Kurashige et al. (2005). Here, we apply numerical inverse Laplace transform, Talbot’s method, to have the semi analytical solution in the time domain.

There are two difficulties to simulate the Mandel problem with a quasi-static model. First, the initial pressure imposed by the Mandel problem results a singular pressure gradient at the side boundaries. Second, implementation of the rigid loading plate requires solving a contact problem. Thanks for the pressure stabilization method as described by the 2D and 3D models, at $r=0$, against the Cartesian and cylindrical Mandel solutions respectively. Where the normalized time $\tau_z$ is given by

$$\tau_z = \frac{at}{c}$$

$K$ being the solid bulk modulus and $S$ being the storage coefficient. Because of the difference between the model and Mandel problem in the initial state as mentioned, the comparisons show greater discrepancy at $\tau_z = 0$ than at later times.

Note that, the initial state is not only missed by the numerical solution but also missed by the analytical solution, using the inverse Laplace transform, Kurashige et al. (2005). We put the time zero pressure there just for reference. Since the analytical solution is dimensionless, we have to normalize the numerical pressure in order have a comparison, similar to Jha and Juanes (2014).

Fig. 4 plots the normalized pressure as a function of time at $(r) = 0$ by the 2D and 3D models, at $z=0$, against the Cartesian and cylindrical Mandel solutions respectively due to dimensional effect.

The 2D (Cartesian) and 3D (cylindrical) models (Mandel solutions) show significantly different pressure curves.

To demonstrate the relation between the mesh resolution and the results, especially the pressure peak at $t=0$ and near $r=a$, we make two meshes for the same 3D model. One of them has 10 cells along the radius, and another one has 20 cells along the radius. Fig. 5 compares the two numerical pressures against the analytical result. The pressure peak becomes lower and closer to $r=a$ as we refine the mesh. The finer mesh has pressure in between the coarser mesh and analytical pressures. This suggests that the numerical solution is approaching the analytical one with the mesh refinement. However, due to the linear continuum nature of the code, the peak adjacent to the zero pressure boundary, although becoming lower, would remain for a refined mesh. Another way to improve the agreement is to refine the mesh where the pressure gradient is large, see Appendix C.4.

### 3. Viscoelasticity

For Maxwell power law viscoelasticity, the deformation has affect on the model in both the stiffness matrix $K_e$ and the RHS function $\Delta F_e$, Eqs. (4) and (5) by Melosh and Raefsky, 1980.

$$K_e^{(n+1)} = \int_0^t B'(D) + \alpha \Delta \varepsilon_{n+1}^*(\varepsilon) \, dt = \int_0^t B'(D) + \alpha \Delta \sigma_{n+1}^*(\sigma) \, dt$$

$$\Delta F_e^{(n+1)} = \int_0^t \rho \nabla \cdot (\varepsilon) \, dt = \int_0^t \rho \nabla \cdot (\sigma) \, dt$$

where, $B$ is displacement to strain matrix depending on the element geometry, Eq. (A.10). $D$ is the element stiffness matrix depending on the elastic constants.

$$\beta = \frac{\sigma^{(n+1)}}{4\eta} C_e : \varepsilon, \quad \varepsilon \geq 1, \quad \sigma = \sum_{i<j} (\sigma_{ij}^e - \sigma_{ij}) \varepsilon(i, j) \varepsilon, \quad d = 2 \text{ or } 3,$$

$$C_e = \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix} \quad 2D,$$

$$C_e = \begin{bmatrix} 4/3 & -2/3 & -2/3 \\ -2/3 & 4/3 & -2/3 \\ 2/3 & -2/3 & 4/3 \end{bmatrix} \quad 3D.$$

### Table 1

Poroelastic model parameters: Young’s modulus $E$, Poison’s ratio $\nu$, fluid mobility $k$, Biot’s coefficient $a_b$, fluid bulk modulus $K_f$.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$E$ [Pa]</td>
<td>3.0E10</td>
</tr>
<tr>
<td>$\nu$</td>
<td>0.25</td>
</tr>
<tr>
<td>$k$ [m²/Pa/s]</td>
<td>1.0E-12</td>
</tr>
<tr>
<td>$a_b$</td>
<td>0.96</td>
</tr>
<tr>
<td>$K_f$ [Pa]</td>
<td>2.2E9</td>
</tr>
</tbody>
</table>
To model visco-poroelasticity, only the submatrix $K_e$ and the subvector $f_n$ in Eq. (2) are affected, Eq. (4), by viscoelasticity. At time zero, $\Delta t_{-1} = 0 \ (n=0)$, the stiffness matrix is the same as for a purely elastic formation.

If the stress exponent $\nu=1$, i.e. the medium is Newtonian viscoelastic, we have a constant $\beta = \frac{1}{\kappa} C$. The stiffness matrix will not vary with the time index $n$, if the model time step is evenly spaced, $\Delta t = \text{const}$. To compare Defmod against some analytically solutions, e.g. Savage and Prescott, 1978, one needs to emulate an infinite half space and implement multi-window fault constraint. Nevertheless, a comparison
against an established FEM code Abaqus™ Simulia (2008) for a viscoelastic problem is provided by Ali (2014, 2015), where the two codes produce identical results, Appendix C.4.

4. Fault model and benchmark

A fault is defined by nodes coinciding on an interior surface, and belonging to different elements on different sides of the surface. Fig. 6 demonstrates a 2D (x, z) fault model.

The constraint equation for the node pair \( (n_i, n_j) \) in Fig. 6 is given by Eq. (7)

\[
\begin{bmatrix}
    n_i & n_j & 0 & -n_s & -n_c \\
    t_x & t_z & 0 & -t_x & -t_z \\
    0 & 0 & 1 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
    u_x^{(1)} \\
    u_z^{(1)} \\
    u_x^{(3)} \\
    u_z^{(3)} \\
\end{bmatrix}
= \mathbf{0}, \text{ for slipping fault}
\]

\[
\begin{bmatrix}
    n_i & n_j & 0 & -n_s & -n_c \\
    t_x & t_z & 0 & -t_x & -t_z \\
    0 & 0 & 1 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
    u_x^{(1)} \\
    u_z^{(1)} \\
    u_x^{(3)} \\
    u_z^{(3)} \\
\end{bmatrix}
= \mathbf{0}, \text{ for locked fault}
\]

where, \( u^{(i)} \) is the nodal displacement.

The difference between a non-slipping (locked) fault and a slipping fault is shown by the last two rows of the first equation in Eq. (7), which prohibit the slipping movement. We assume that there is no separation and interpenetration, so the first row remain for a slipping fault.

For the poroelastic model with a permeable fault, Eq. (8) applies.

\[
\begin{bmatrix}
    n_i & n_j & 0 & -n_s & -n_c \\
    t_x & t_z & 0 & -t_x & -t_z \\
    0 & 0 & 1 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
    u_x^{(1)} \\
    u_z^{(1)} \\
    p^{(1)} \\
    u_x^{(3)} \\
    u_z^{(3)} \\
    p^{(3)} \\
\end{bmatrix}
= \mathbf{0}, \text{ slipping, permeable}
\]

where, \( p^{(i)} \) is the nodal pressure. For a partly permeable fault, Eq. (7) applies to the impermeable part, and Eq. (8) applies to the permeable part. For the permeable fault (part), the pressure will not jump across the fault interface that has infinitesimal width in this modeling work.

Eq. (9) combines the governing equation and the constraint equation together.

\[
\begin{bmatrix}
    K & G^T & \Delta U_n \\
    G & 0 & \lambda_n \\
\end{bmatrix}
\begin{bmatrix}
    \Delta L \\
    \lambda_n
\end{bmatrix}
= \mathbf{0}
\]

where the Lagrange Multiplier, \( \lambda_n \), is the nodal forces/sources needed to honor the constraint equation. Each row of the constraint matrix G has two exact opposite nonzero entries, Eqs. (7), (8), so \( \lambda_n \) will introduce exact opposite force/source to the node pairs on different sides of the fault. The force on the node pair (1, 3) in Fig. 6, for instance, is given by Eq. (10)

\[
f^{(1)} = -f^{(3)} = \begin{bmatrix}
    \lambda_{n,x}^{(1,3)} \\
    \lambda_{n,z}^{(1,3)} \\
    \lambda_{x,x}^{(1,3)} \\
    \lambda_{x,z}^{(1,3)} \\
    \lambda_{x,z}^{(1,3)} \\
    \lambda_{z,z}^{(1,3)} \\
\end{bmatrix}
\]

\[
f^{(1)} = -f^{(3)} = \begin{bmatrix}
    \lambda_{n,x}^{(1,3)} \\
    \lambda_{n,z}^{(1,3)} \\
    \lambda_{x,x}^{(1,3)} \\
    \lambda_{x,z}^{(1,3)} \\
    \lambda_{x,z}^{(1,3)} \\
    \lambda_{z,z}^{(1,3)} \\
\end{bmatrix}
\]
The fault shear to normal stress ratio at the node (1, 3) is equivalent to the shear to normal nodal force ratio given by Eq. (11).

\[
\frac{\sigma_{\text{fi}}^{(1,3)}}{\sigma_{\text{nn}}^{(1,3)}} = \frac{\| \lambda_{\text{fi}}^{(1,3)} \|}{\| \lambda_{\text{nn}}^{(1,3)} \|}
\]  

(11)

where, the normal stress is assumed compressive. This equality allows us to evaluate the slip tendency of a fault patch without evaluating the element-wise stress, e.g. Eq. (12) for nodes 1 and 2 in Fig. 6, Zienkiewicz, 2000.

\[
\sigma_{\text{ff}}^{(1,2)} = \frac{(T_{\text{fi}}^{(1)}T_{\text{ff}}^{(1)})_{22}}{(T_{\text{nn}}^{(1)}T_{\text{ff}}^{(1)})_{22}}, \quad \text{at node 1}
\]

\[
\frac{\sigma_{\text{ff}}^{(2)}}{\sigma_{\text{tt}}^{(2)}} = \frac{(T_{\text{fi}}^{(2)}T_{\text{ff}}^{(2)})_{22}}{(T_{\text{nn}}^{(2)}T_{\text{ff}}^{(2)})_{22}}, \quad \text{at node 2}
\]

(12)

where, the matrices B and D are the same as in Eq. (4) and Appendix A Matrix T translates the stress tensor from the (x,z) coordinates to (t, n) coordinates in Fig. 6. If the fault is curved, the translation matrix T varies with position. Stress tensor \( \sigma \) is anisotropic, homogeneous all over the domain, the Mohr-Coulomb model is known analytically.

Fig. 7 illustrates the anisotropic loading model. At the start, we incrementally load the sample at the rate of \( \Delta \sigma_{\text{ft}} = \Delta \sigma_{\text{fn}} = 0.2 \, \text{MPa/yr}. \) At year 50, we stop the loading increment in the horizontal direction, and keep the increment in the vertical direction until year 150. For all the static benchmark problems, we assume the shear stress is not enough to make the frictional contact slip, then the contact is locked. In this case, the fault orientation (curvature) and local stress tensor determine the local Coulomb stress.

Because of the fault curvature, the nodal force is varying along the fault at anisotropic stress state. At different depths, the fault has different dip angle \( \alpha \) and associated stress \( \tau \) and \( \sigma_n \). Since the stress is, although anisotropic, homogeneous all over the domain, the Mohr-Coulomb circle is known analytically.

Fig. 8 compares the Mohr-Coulomb circles calculated by Defmod against the analytical ones. The absolute numerical stress \( \tau \) and \( \sigma_n \) are again from the nodal force, i.e. Lagrange Multiplier. To translate the nodal force (Newton) to stress (Pa), we compute the node-wise force-to-stress ratio from the first step (isotropic) loading result.

Shown by Fig. 8, although the stress is different at different depths, for a given loading condition, the resulting Mohr-Coulomb circles are the same, and overlap well with the analytical ones. We use the fault vectors (strike, dip and normal) to construct the constraint equation, Eqs. (7), (8). The vectors however, only have definition on the fault (element) facets. To have the vectors at a node (pair), we calculate them for all the adjacent facets, and take average. The resulting vectors are in general not strictly equal to the exact fault vectors that are given analytically in this example. This contributes the slight misfit in Fig. 8. When the fault curvature is high and/or the element (facet) is large, this approximation may result poorly. Fortunately, the unstructured mesh used by Defmod can be locally refined.

Fig. 9 compares the stress ratio \( \tau / \sigma_n \) calculated by Defmod at different depth and time against the analytical values. The numerical stress ratio is given by the equivalent ratio of the nodal force, i.e. Lagrange Multiplier in Eq. (11).

In another example the fault is planar and has a dip angle \( \beta = 60^\circ \). We apply gravity at start and gradually add compressional traction on the top, \( \sigma_{\text{gz}} = 0.2 \, \text{MPa/yr}. \) The bottom and side walls have zero normal displacement. Fig. 10 plots the stress \( (\tau, \sigma_n) \) and associated Mohr-Coulomb circles. In this example, analytical stress \( \sigma_{\text{gn}} = \rho_{\text{fz}} + \sigma_{\text{ft}} \) and \( \sigma_{\text{n}} = \frac{\rho_{\text{fz}} + \sigma_{\text{ft}}}{1-\nu} \). For \( \nu = 0.25 \), all the stress evaluations \( (\tau, \sigma_n) \) should appear on a straight line that passes the origin, and has \( \sqrt{3}/3 \) slope, \( \theta = 30^\circ \). Since \( 2\beta - \theta = \pi/2 \), this line should be tangent with all the Mohr-Coulomb circles. This is exactly shown by the figure.

5. Dynamic model

An elastodynamic problem has the governing equation given by Eq. (13).

\[
\dot{\mathbf{u}} + \mathbf{C} \ddot{\mathbf{u}} + \mathbf{K} \mathbf{u} = \mathbf{f}.
\]  

(13)

where, \( \mathbf{M} \) is the mass matrix; \( \mathbf{C} = \alpha_i \mathbf{M} + \beta_i \mathbf{K} \) is the damping matrix; \( \alpha_i \) and \( \beta_i \) are the Rayleigh damping coefficients. The detailed matrices and RHS vector formulations are given in Appendix A.

Newmark explicit scheme has Eq. (14).

\[
\Delta \mathbf{u}_i = \mathbf{M}^{-1}(\Delta \mathbf{f}_i - \mathbf{K} \mathbf{u}_{i-1}) - \Delta \mathbf{C}(\mathbf{u}_{i-1} - \mathbf{u}_{i-2}) + 2\mathbf{u}_{i-1} - \mathbf{u}_{i-2}
\]

(14)

The incremental form is given by Eq. (15).

\[
\Delta \mathbf{u}_i = \mathbf{M}^{-1}(\Delta \mathbf{f}_i - \mathbf{K} \mathbf{u}_{i-1}) - \Delta \mathbf{C}(\Delta \mathbf{u}_{i-1} - \Delta \mathbf{u}_{i-2}) + 2\Delta \mathbf{u}_{i-1} - \Delta \mathbf{u}_{i-2}
\]

(15)

Eq. (15) provides the unconstrained dynamic solution at \( t_i \). For the constrained dynamic solution we apply the forward incremental Lagrange Multiplier method, Carpenter et al., 1991, Eq. (16).

\[
\lambda_i = (\Delta^T \mathbf{M}^{-1} \mathbf{G} \lambda_i - \mathbf{f} \mathbf{u}_{i-1}) = \Delta \mathbf{u}_{i-1} - \Delta \mathbf{M}^{-1} \mathbf{G} \mathbf{u}_{i-1}
\]

(16)

where the Lagrange Multiplier \( \lambda_i \) is the nodal force needed to constrain the solution. \( \lambda \) itself, being a stress proxy, is constrained by physical laws, e.g. yielding stress or maximum friction. The model implements the frictional fault slip by capping the friction, i.e. \( \lambda \) component perpendicular to the fault face, with the product of friction coefficient and normal stress, i.e. \( \lambda \) component perpendicular to the fault face. Here, we call this approach, Bartolomeo et al. (2010), Lagrange Multiplier (LM) capping method. This method, as appose to the traction at split node (TSN) method, Dalguer and Day (2007), does not require solving a subspace problem.

Lyzmer and Kuhlemeyer (1969) propose that the absorbing boundary for the elastodynamic model can be achieved by adding additional terms to the damping matrix, Eq. (17), for all the element facets on the boundary.

\[
c_i = \begin{cases} 
    c_i + \int J_i \rho \mathbf{V}_d \mathbf{d} \Gamma, & \text{p wave along } i^{\text{th}} \text{ axis,} \\
    c_i + \int J_i \rho \mathbf{V}_d \mathbf{d} \Gamma, & \text{for all } i - \text{axes.} 
\end{cases}
\]

(17)
6. (Quasi-)static dynamic hybrid model

A (quasi-)static-dynamic hybrid solver has been implemented by Meng (2015). Fig. 11 gives the flow chart of this solver. When the stress state, $\tau$ and $\sigma_n$, satisfies some failure criterion, the dynamic solver will take over so that the fault will slip. The constraint equations associated with the tangential direction of the slipping patches are relaxed by limiting the maximum friction, required to prevent the fault from slip, with the yielding limit, $\mu \sigma_n$, where $\mu$ is the frictional coefficient. At the end of a dynamic run, an implicit (static) solver will reevaluate the stress drop, and ensure that the fault will be no longer slipping, i.e., with zero net force, after returning to a quasi-static state.

Fig. 12 illustrates the rupture process. When the split node pairs start to shear, they will squeeze and stretch their neighbors, resulting in stress localization at the rupture fronts. This localized stress will drive the rupture to propagate further. The dynamic fault rupture, in return, will relax the shear stress on the slipping pairs, and reduce their slipping momentum. A slipping patch, in 1D (2D model), may break...
Fig. 12. Schematics of the rupture process:  \( f_i \) denotes the net nodal force at different steps, the green frames enclose the slipping patches, the red frames enclose the locked patches, and the cyan frames enclose the rupture/arrest fronts. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

into segments and have multiple rupture and arrest fronts.

Once the whole fault is stabilized, i.e. no dynamic ruptures is taking place, the model is expected to jump back to the quasi-static (parent) loop, as in Fig. 11. However, we are often interested in the wave forms on the top domain surface due to the dynamic rupture. Therefore, we keep the dynamic model running for some time to let the wave forms reach the synthetic seismometers before having the model back to the parent loop. The mode shift is often accompanied by boundary condition changes. A 3D subsurface dynamic problem usually has different boundary conditions, e.g. the sides and bottom being absorbing, and the top being traction free, than those under the static mode.

7. Benchmarking the hybrid model

Due to the complexity of the hybrid model, it is hard to find an analytical solution to compare against. Southern California Earthquake Center (SCEC) hosts a dynamic earthquake rupture code verification project, Harris et al. (2010), where different models can compare against each other by solving the same set of synthetic problems. A dynamic rupture problem can be described by a single iteration of a hybrid model, Fig. 11. When a quasi-static state, with prescribe fault stress, satisfies the slipping condition, the model will run in the dynamic (child) loop, and generate waveforms at given observation locations. The focus of the SCEC benchmark is indeed comparing the waveforms by different models.

Here, we present the result for the benchmark problem TPV205, see http://scecdata.usc.edu/cvws for detailed problem description. The model layout is given in Fig. 13. The strike slip rupture starts at a square shaped nucleation zone, with the shear stress \( \tau_{\text{ref}} \) overshooting the yielding stress, i.e. static friction \( \mu_{\text{st}} \sigma_n \). The background shear stress \( \tau_{\text{ref}} \) is between the dynamic friction \( \mu_{\text{dyn}} \sigma_n \) and static friction. A square patch to the left of the nucleation zone has shear stress \( \mu_{\text{st}} \sigma_n \) higher than the background, and a square patch to right has shear stress \( \mu_{\text{dyn}} \sigma_n \) lower than the background. The temporal friction coefficient follows the slip weakening law, Eq. (18),

\[
\mu = \begin{cases} 
\mu_{\text{st}} - \mu_{\text{dy}} (1 - D/D_i) + \mu_{\text{dy}} & \text{if } D \leq D_i \\
\mu_{\text{dy}} & \text{if } D > D_i 
\end{cases}
\]

(18)

where, \( \mu_{\text{st}} \) and \( \mu_{\text{dy}} \) are the static and dynamic friction coefficients respectively, \( D \) is the slip distance, \( \| u^{\text{slip}} - u^{\text{dyn}} \| \), and \( D_i \) is the slip weakening distance. The parameter values are listed in Table 2.

Fig. 14 compares the three component velocity calculated by Defmod against the codes EqSim (now Pylith), Aagaard et al. (2015), and Roten, Dalguer and Day (2007) and Cui et al. (2010), for the surface station 3 km off the fault, and -12 km along the fault from the epicenter. Fig. 15 shows the waveform comparison for the surface station 3 km off the fault, and 12 km along the fault from the epicenter. Fig. 16 shows the waveform comparison for the subsurface station 3 km off the fault, and -12 km along the fault from the hypocenter. Fig. 17 shows the waveform comparison for the subsurface station 3 km off the fault, and 12 km along the fault from the hypocenter. Fig. 18 shows the rupture front comparison. Within the contour of each time step, 0.5 s spaced, the slip distance is greater than 1 mm.

Both the waveform and rupture front plots suggest that Defmod, being an FE code, agrees more with the FD code Roten than with another FE code EqSim. Note that, the SCEC benchmark presents the comparisons among different numerical codes, whereas the true solutions are, and will remain, unknown. Therefore, only by comparing the SCEC problem results, it is hard to argue which code is more

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v_s )</td>
<td>3464 m/s</td>
</tr>
<tr>
<td>( v_p )</td>
<td>6000 m/s</td>
</tr>
<tr>
<td>( \rho )</td>
<td>2670 kg/m³</td>
</tr>
<tr>
<td>( \sigma_n )</td>
<td>120 MPa</td>
</tr>
<tr>
<td>( \tau_{\text{ref}} )</td>
<td>70 MPa</td>
</tr>
<tr>
<td>( \tau_{\text{dy}} )</td>
<td>81 MPa</td>
</tr>
<tr>
<td>( \tau_{\text{slip}} )</td>
<td>62 MPa</td>
</tr>
<tr>
<td>( D_i )</td>
<td>0.4 m</td>
</tr>
<tr>
<td>( \mu_{\text{st}} )</td>
<td>0.667</td>
</tr>
<tr>
<td>( \mu_{\text{dy}} )</td>
<td>0.525</td>
</tr>
</tbody>
</table>

Table 2: SCEC benchmark TPV205 parameter list.
In addition to TPV205, the similar comparisons for TPV102 are given in Appendix C.5, where the rate and state friction law, Ampuero and Rubin (2008), is implemented. The hybrid model, unlike others, will return to the quasi-static (parent) loop once the dynamic run is over. The resulting fault slip and associated stress perturbation are taken into account when examining if another seismic event shall happen in the following quasi-static steps. This is not in the scope of the SCEC benchmark. Nevertheless, we demonstrate an external loading triggered earthquake by revisiting the anisotropic loading example that would result the fault stress profile evolution given in Fig. 9. If we set the frictional coefficients $\mu_{\text{sta}} = 0.5$ and $\mu_{\text{dyn}} = 0.5$, the fault would rupture before 150 yr at depth about 2 km. When the model returns back to the quasi-static state, the perturbed displacement should show discontinuity across the fault face due to the rupture in the previous dynamic steps.

**Fig. 14.** SCEC TPV205 waveform comparison, station 1, velocity in x, y, z directions.

**Fig. 15.** SCEC TPV205 waveform comparison, station 2, velocity in x, y, z directions.

**Fig. 16.** SCEC TPV205 waveform comparison, station 3, velocity in x, y, z directions.

**Fig. 17.** SCEC TPV205 waveform comparison, station 4, velocity in x, y, z directions.

**Fig. 18.** SCEC TPV205 fault rupture front comparison.

accurate than the rest.

In addition to TPV205, the similar comparisons for TPV102 are given in Appendix C.5, where the rate and state friction law, Ampuero and Rubin (2008), is implemented.

The hybrid model, unlike others, will return to the quasi-static (parent) loop once the dynamic run is over. The resulting fault slip and associated stress perturbation are taken into account when examining if another seismic event shall happen in the following quasi-static steps. This is not in the scope of the SCEC benchmark. Nevertheless, we demonstrate an external loading triggered earthquake by revisiting the anisotropic loading example that would result the fault stress profile evolution given in Fig. 9. If we set the frictional coefficients $\mu_{\text{sta}} = 0.5$ and $\mu_{\text{dyn}} = 0.5$, the fault would rupture before 150 yr at depth about 2 km. When the model returns back to the quasi-static state, the perturbed displacement should show discontinuity across the fault face due to the rupture in the previous dynamic steps.

Fig. 19 plots the displacement increment (delta function) magnitude snapshot showing the seismic radiation due to the fault rupture, left panel, and the x-displacement showing discontinuity due the rupture, right panel. The seismic radiation is capture by the dynamic part of the hybrid solver, and the perturbed displacement is captured by the quasi-static (parent) part. When the static-to-dynamic shift happens, the side and bottom boundaries will change from loading and zero normal displacement conditions to absorbing condition. When dynamic-to-static-shift happens, the boundary conditions will change back.
8. Summary

Table 3 lists the major functionalities implemented in Defmod. Defmod models the pore fluid flow and resulting matrix deformation by a standalone approach, without coupling with any external code. A benefit of this approach is that the effort to build, maintain and upscale the model is considerably less than a two way coupled approach, Jha and Juanes (2014). The code scales uniformly across the fluid and solid parts.

The code integrates the (quasi-)static and dynamic processes into an episodic (hybrid) process, using two time scales. The perturbations introduced by the previous seismic events are taken into account in the following computing steps. This feature enables one to model the interactions between the sequentially induced earthquake events.

The code, being sophisticated, remains light weight. The scalability is inherited from the linear algebra toolkit PETSc (Balay et al., 2015), which ensures high performance across platforms from a single processor laptop to a multi-processor cluster, see Ali (2014) for detailed scalability.

Defmod, being an FEM code, is not without limitations. Only single phase, slightly compressible pore fluid is considered. This code is not meant to replace a reservoir simulator for multiphase multicomponent and/or reactive porous flow.

Acknowledgements

The author thanks Dr. Tabrez Ali at Wisconsin Madison University for making the original Defmod code open source. The author is sponsored by Shell Company (SHELL MITEI FDG MEMBER CONTRACT PT30193).

Appendix A. Integration forms of the governing equations

The elastodynamic deformation \( \mathbf{u} \) has the governing equation:

\[
\frac{\partial^2 \mathbf{u}}{\partial t^2} + \eta \frac{\partial \mathbf{u}}{\partial t} - \nabla \cdot \sigma = \mathbf{f}, \quad \mathbf{u} = \mathbf{D} \mathbf{e}, \quad \mathbf{e} = \nabla \mathbf{u},
\]

(A.1)

\( \mathbf{D} \) is elasticity matrix depending on the material properties, and \( \mathbf{f} \) is body force. Generalized-\( \sigma \) method for time discretization:

\[
\rho \mathbf{u}_{n+1} - \eta \mathbf{u}_{n+1} - \nabla \cdot \sigma_{n+1} - \mathbf{f}_{n+1} = 0,
\]

where,

\( \mathbf{u}_{n+1} = (1 - \alpha_i)\mathbf{u}_{n+1} + \alpha_i \mathbf{u}_n, \quad \mathbf{u}_{n+1} - \eta \mathbf{u}_{n+1} = (1 - \alpha_i)\mathbf{u}_{n+1} + \alpha_i \mathbf{u}_n, \quad \nabla \cdot \sigma_{n+1} = (1 - \alpha_i)\nabla \sigma_{n+1} + \alpha_i \nabla \sigma_n, \quad \mathbf{f}_{n+1} = (1 - \alpha_i)\mathbf{f}_{n+1} + \alpha_i \mathbf{f}_n.
\]

The Newmark time stepping scheme for a choice of \( \beta \) and \( \gamma \):

\[
\sigma_i = \sigma \left( \mathbf{u}_i \right).
\]

Table 3

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Method</th>
<th>Benchmark</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td></td>
<td>Smith and Griffiths (2004), Bochev and Dohrmann (2006), Kurashige et al. (2005), Melosh and Raefsky (1980)</td>
<td></td>
</tr>
<tr>
<td>Poroelasticity</td>
<td>pore pressure</td>
<td>Mandel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>stabilization</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Viscoelastic</td>
<td>Implicit</td>
<td>Abaqus</td>
<td></td>
</tr>
<tr>
<td>power law</td>
<td></td>
<td>Mohr-Coulomb</td>
<td>Zienkiewicz (2000)</td>
</tr>
<tr>
<td>(quasi)static</td>
<td>Lagrange</td>
<td></td>
<td></td>
</tr>
<tr>
<td>constraint</td>
<td>Multiplier</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Elastodynamic</td>
<td>Forward</td>
<td></td>
<td></td>
</tr>
<tr>
<td>constraint</td>
<td>Lagrange</td>
<td>SCEC</td>
<td>Carpenter et al. (1991)</td>
</tr>
<tr>
<td>absorbing</td>
<td>viscous damping</td>
<td></td>
<td>Lysmer and Kuhlemeyer</td>
</tr>
<tr>
<td>Isostasy</td>
<td>Winkler</td>
<td></td>
<td>Fryba (1995)</td>
</tr>
<tr>
<td>Fault/faulting</td>
<td>implicit/explicit</td>
<td>SCEC</td>
<td>Bartolomeo et al. (2010)</td>
</tr>
</tbody>
</table>

Fig. 19. Left: dynamic output showing displacement increment (delta function) magnitude within one time step due to the fault rupture. Right: quasi-static output showing the absolute x-displacement that is discontinuous due to the previous dynamic rupture.
\[ u_{n+1} = \frac{\gamma}{\beta \Delta t} (u_{n+1} - u_n) - \left( \frac{\gamma}{\beta} - 1 \right) \Delta t \frac{\gamma}{\beta} - 1 \right) u_n. \]  
(A.3)

After substitutions, the only variable that has subscript \( n + 1 \), i.e. needs to be sought for time \( t_{n+1} \), is the displacement \( u_{n+1} \). All the other variables are available from the previous time steps.

We put all the unknowns to the left hand side of the equation, and put to the right-hand side:

\[ \text{LHS} = c_i^{(1)} u_i + c_i^{(2)} u_i - (1 - \alpha) \Delta \sigma_{i+1} \]  
\[ \text{RHS} = c_i^{(1)} u_i + c_i^{(2)} u_i + c_i^{(3)} u_i + c_i^{(4)} u_i + c_i^{(5)} u_i + c_i^{(6)} u_i + c_i^{(7)} u_i + (1 - \alpha) \Delta \sigma_{i+1} + \alpha \Delta \sigma_i, \]

where,

\[ c_i^{(1)} = \frac{\rho(1 - \alpha u_i)}{\beta \Delta t}, \]  
\[ c_i^{(2)} = \frac{\rho(1 - \alpha u_i)}{\beta \Delta t}, \]  
\[ c_i^{(3)} = \eta \left(1 - \alpha \Delta \sigma_i \right) \gamma + \beta, \]  
\[ c_i^{(4)} = \eta \left(1 - \alpha \Delta \sigma_i \right) \gamma - 2 \beta, \]  
\[ c_i^{(5)} = \frac{\eta \left(1 - \alpha \Delta \sigma_i \right) \gamma - 2 \beta}{2 \beta}. \]  
(A.4)

To solve this equation using finite element method, we write Eq. (A.4) in the weak form by multiplying the LHS and RHS with a vector valued test function \( r(x) \), and take volume integrals over the model domain.

\[ \text{(r. LHS)}_2 = \text{(r. RHS)}_2, \quad \forall \ r \in \mathbb{L}^2(H^1(\Omega)). \]  
(A.5)

where, \( (\cdot, \cdot) \) denotes the \( \mathbb{L}^2 \) inner product integrated over domain \( \Omega \).

Integrating by part the stress divergence terms, and applying the general Stokes’ Theorem, we have:

\[ \text{(r. (V \sigma_i))}_2 = -(\nabla r, \sigma_i)_2 + (\text{(r. } \sigma_i n) n)_2, \]  
(A.6)

where, \( n \) is normal vector on the domain boundary denoted by \( \partial \Omega \).

For given boundary traction \( \sigma_i = \sigma_i n \), the governing equation is written as:

\[ a = L, \]  
\[ \text{where} = (c_i^{(1)} u_i + c_i^{(2)} u_i + c_i^{(3)} u_i + c_i^{(4)} u_i + c_i^{(5)} u_i + c_i^{(6)} u_i + c_i^{(7)} u_i) \]  
\[ + (c_i^{(4)} u_i)_2 - (\sigma_i, n)_2 + (r, (1 - \alpha \tau_{i+1} + \alpha \tau_i))_2 + (r, (1 - \alpha \tau_{i+1} + \alpha \tau_i))_2, \]  
\[ \text{for} (A.7) \]

This equation can be solved for given initial conditions \( u_0, u_0, u_0 \), boundary conditions \( \tau_i \) and body forces \( f_i \). Usually the body force is a known constant, e.g. \( f_{i+1} = f_i = \rho g \). A second order time scheme, is an implementation of the generalized-\( \alpha \) method with specific \( \beta, \gamma, a_m, \alpha \). A first order time scheme is obtained by neglecting the last terms in Eq. (A.3).

The test function, \( r(x) \), within an element \( e \) is given by the products of the shape function, \( N^{(e)}(x) \), \( \{N_1 N_2 \ldots \} \), and the nodal values, \( r^{(e)} \) \( \{r_f r_i \ldots \} \), \nabla r(x) = \nabla^{(e)}(x)r^{(e)}. \]  
(A.8)

The width (height) of the two product terms is the node count of the element. Similarly, the gradient of the test function is given by

\[ \nabla r(x) = \nabla^{(e)}(x)r^{(e)}. \]  
(A.9)

The matrix \( B^{(e)} \) is given by, Smith and Griffiths (2004),

\[ B^{(e)} = \text{AS}^{(e)}. \]  
(A.10)

Matrix \( A \) is a derivative operator and matrix \( S^{(e)} \) is the shape function stretched in the test/trial function dimension. For 3D Cartesian case, these two matrices are

\[ \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 \\ 0 & \frac{\partial}{\partial y} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} \end{bmatrix} \begin{bmatrix} N \end{bmatrix} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \ldots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & 0 & \ldots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & 0 & \ldots \end{bmatrix} \]  
(A.11)

The width of \( S^{(e)} \) is the element node count multiplied by dimension, 3. The shape function and its spatial derivatives differ for different choice of element. If \( B^{(e)} \) has the layout determined by Eq. (A.10), (A.11), the \( r^{(e)} \) should be flattened, i.e. \( \{r_f r_i \ldots \} \), to keep the product in Eq. (A.9), conforming.

For a flattened test function, \( N^{(e)} \) should be substituted by \( S^{(e)} \), to keep the product of Eq. (A.8) conforming. As shown later however, the shape function itself only appears in \( N^{(e)}N^{(e)} \) when assembling the element mass (and/or damping) matrix. Instead of forming \( S^{(e)} \), we first take this inner product, and stretch it to be conforming with the flattened nodal vector space, \( r^{(e)} \). In Defmod, this is done by function FormELM in the file m_local.F90. Without causing ambiguity, we keep using the notation \( N^{(e)} \) together with the presumably flattened \( r^{(e)} \).

The element integrals of the inner products in Eq. (A.7) become

\[ \langle r (\cdot) \rangle_e = r^{(e)T} \int_{\Omega_e} N^{(e)T} \cdot r^{(e)} d\Omega_e, \]  
\[ \langle \nabla r (\cdot) \rangle_e = r^{(e)T} \int_{\Omega_e} B^{(e)T} (\cdot) d\Omega_e. \]  
(A.12)

The equation Eq. (A.8) also applies to the trial functions, i.e. \( u(x) \) and its time derivatives, by substituting \( r^{(e)} \) with \( u^{(e)} \) and its time derivatives. Likewise, the element (nodal) vectors are presumably flattened.

The first row of Eq. (A.12), after writing the trial functions in form of Eq. (A.8), and substitute \( \cdot \), becomes

\[ \langle (c^{(1)} u^{(e)} \rangle_e = c^{(1)} \int_{\Omega_e} N^{(e)T} N^{(e)} u^{(e)} d\Omega_e, \]  
(A.13)

where \( c^{(1)} \) is one of the coefficients listed in Eq. (A.7) that contain \( \rho \). Eq. (A.12) suggests that, a left multiplication term \( r^{(e)T} \) would be resulted by all the integrals on both LHS and RHS of Eq. (A.7). Due to the arbitrariness of \( r \), the equation should hold before the left multiplication. What needs to be integrated by element is
\[ \int_{\Omega} \rho \mathbf{N}^{(e)} \mathbf{N}^{(e)} d\Omega = \mathbf{M}^{(e)}. \]  
(A.14)

When \( \mathbf{U} \) is available, we use the vector \( \mathbf{M} \mathbf{U} \) to assemble the RHS, if not, we use the matrix \( \mathbf{M} \) to assemble the factor matrix of the LHS. Here, we use capital vector symbols to denote the global vectors, and matrix symbols without superscript \( e \) to denote the global matrices. The global matrices and vectors are formed by adding the contribution from each element via an element-to-global degree of freedom map.

The product of \( \mathbf{B}^{(e)} \) and \( \mathbf{u}^{(e)} \) yields the (vector shaped) strain tensor.
\[ \epsilon(x) = \mathbf{B}^{(e)}(x) \mathbf{u}^{(e)}. \]  
(A.15)

The formulation of \( \mathbf{B}^{(e)} \) insures a symmetric tensor \( \epsilon \), known as symmetric gradient. Substituting the strain in Eq. (A.1) by Eq. (A.15), the stress within the element \( \epsilon \) is then
\[ \sigma(x) = \mathbf{D}^{(e)} \mathbf{B}^{(e)}(x) \mathbf{u}^{(e)}. \]  
(A.16)

The second row of Eq. (A.12), after substituting \((-\omega)\) with Eq. (A.16), becomes
\[ (\nabla \sigma)_{\Omega} = \epsilon^{(T)} T \int_{\Omega} \mathbf{B}^{(e)} \mathbf{u}^{(e)} d\Omega. \]  
(A.17)

Again, because of the arbitrariness of the test function, what needs to be integrated is
\[ \int_{\Omega} \mathbf{B}^{(e)} \mathbf{u}^{(e)} d\Omega = \mathbf{K}^{(e)}. \]  
(A.18)

When \( \mathbf{U} \) is available, we use the vector \( \mathbf{K} \mathbf{U} \) to assemble the RHS, if not, we use the matrix \( \mathbf{K} \) to assemble the factor matrix of the LHS.

In principal, the damping matrix should be integrated like
\[ \int_{\Omega} \mathbf{N}^{(e)} \eta \mathbf{N}^{(e)} d\Omega = \mathbf{C}^{(e)}. \]  
(A.19)

The damping parameter \( \eta \) can be scalar or tensor (e.g. absorbing boundary) valued. In Defmod, instead of doing this integration, we use the relation \( \mathbf{C} = e \mathbf{M} + \beta \mathbf{K} \) to form the global damping matrix. The scalar parameters \( e \) and \( \beta \), known as ‘Rayleigh’ damping coefficients, are discussed by Timoshenko et al., 1974.

In Defmod we implement a first order time scheme, i.e. neglect the last terms in Eq. (A.3), and set the parameters \( r = 1, \beta = 1, a_m = 0 \) and \( a_c = 1 \). This leads to, after writing Eq. (A.7) in the global integral form,
\[ \mathbf{M} \ddot{\mathbf{U}}_{n+1} = -e \mathbf{C} \mathbf{U}_n - \mathbf{K} \mathbf{U}_n + \int_{\Omega} \mathbf{N}^{(e)} \mathbf{N}^{(e)} d\Omega + \int_{\partial\Omega} \mathbf{N}^{(e)} \mathbf{u}^{(e)} d\partial\Omega. \]

Eq. (A.14) is then obtained by adjusting the time indices, and combining the traction and body force terms into a global nodal vector \( \mathbf{f} \).

In Defmod, the boundary traction is given facet-wise. Instead of making the boundary integral as in Eq. (A.20), we multiply the traction with the facet area, and evenly distribute the force to the nodes associated with the facet. Likewise, we evenly distribute the element body force (weight) to the associated nodes, instead of making the volume integral as in Eq. (A.20). In addition to boundary traction and body force, one can also specify nodal force directly.

This first order approach is attractive because the mass matrix can be diagonalized, Zienkiewicz (2000), without significantly affecting the accuracy. This diagonalization, also known as lumping, makes the updates Eqs. (15), (16) explicit. In Defmod, mass matrix lumping is done element-wise, following a simple row sum method, see function FormElM. The Courant-Friedrichs-Lewy criterion states that, the maximum time step and maximum wave frequency in this explicit scheme is restricted by
\[ \Delta t \leq L / \sqrt{\mu / \rho}, \quad \Delta t \leq 0.1 / \sqrt{\mu / \rho}, \]  
(A.21)

where \( L \) is the representative length of the elements, \( E \) and \( \mu \) are the elastic moduli. Note that, \( \sqrt{\mu / \rho} \) gives the shear wave velocity, which means one wavelength should be described by at least ten elements. From the aiming frequency, density and elastic moduli, one can estimate the appropriate \( L \) and then \( \Delta t \). The problem size is determined by the domain scale and \( L \).

In the context of ground motion, the uncertainties from the earth model would certainly overwhelm the artifact due to the low order time scheme and mass matrix lumping, which makes this explicit approach adequate enough. With all the necessary parts, \( \mathbf{K}, \mathbf{M} \) and \( \mathbf{f} \), in hand, one should be able to formulate a second order approach without too much work. In that case, \( \mathbf{M} \) does not have to be lumped, because a non-trivial matrix inversion cannot be avoided in the first place, i.e. requiring implicit solving. This only makes sense when accuracy rather than efficiency becomes priority.

An elastostatic problem has the integration governing equation, by simplifying Eq. (A.20),
\[ \mathbf{K} \mathbf{U}_n = \mathbf{f}_n, \]  
(A.22)

where \( \mathbf{K} \) may need to be updated depending on the viscoelasticity order.

The slightly compressible pore fluid pressure \( p \) has the governing equation:
\[ c_f \phi \phi - \nabla \cdot k \nabla p - a_{pf} \epsilon = q, \]  
(A.23)

where \( a_{pf} = a_{bf} [1 \ 1 \ 1 \ 0 \ 0 \ 0]^T, \epsilon = [\epsilon_{xx} \ \epsilon_{yy} \ \epsilon_{zz} \ \epsilon_{yz} \ \epsilon_{xz} \ \epsilon_{xy}]^T \) in 3D, \( a_{bf} \in [0, \ 1] \) being the Biot’s coefficient, \( c_f \) is the total compressibility, \( k \) is the fluid mobility (permeability divided by fluid viscosity) and \( q \) is the fluid body source (injection positive).

We multiply both sides of Eq. (A.23) by a scalar valued test function \( r, \forall r \in L^2(\mathcal{O}) \), and integrate over \( \mathcal{O} \). The term with pressure gradient, after integrated by part, becomes
\[ (r, \nabla \cdot k \nabla p)_{\mathcal{O}} = -(\nabla r, k \nabla p)_{\mathcal{O}} + (r, n \cdot k \nabla p)_{\partial \mathcal{O}}. \]  
(A.24)
We use \( f \) to denote \( \mathbf{n} \cdot \mathbf{k} \mathbf{p} \), the fluid flux in the boundary normal direction \( \mathbf{n} \), The weak form governing equation becomes
\[
\mathbf{a} = \mathbf{L} \text{ where, } \mathbf{a} = (r, c \phi) + (\nabla r, \mathbf{k} \mathbf{V} \mathbf{p})_{\Omega} - (r, \mathbf{a}_{\mathbf{q}} \mathbf{q})_{\Omega} = (r, q)_{\Omega} + (r, f)_{\partial \Omega}.
\] (A.25)

Note that, we use the bold font to differentiate the vector/tensor values from scalar values. The fluid mobility \( k \) is sometimes tensor valued,
\[
\mathbf{k} = \begin{bmatrix}
k_x & k_y & k_z
\end{bmatrix}
\]
in 3D. Eq. (A.8) stays the same after substituting the vector test function with the scalar one. Eq. (A.9) becomes
\[
\mathbf{V}_f (x) = \mathbf{dN}^{(i)}(x) \mathbf{r}^{(i)},
\] (A.26)
where in 3D,
\[
\mathbf{dN}^{(i)} = \begin{bmatrix}
\partial N_i/\partial x & \partial N_j/\partial x & \cdots \\
\partial N_i/\partial y & \partial N_j/\partial y & \cdots \\
\partial N_i/\partial z & \partial N_j/\partial z & \cdots
\end{bmatrix}.
\] (A.27)
The width of \( \mathbf{dN}^{(i)} \) is the node count of the element.

Similar to Eq. (A.12), the element integrations in Eq. (A.25) are of the forms,
\[
(r, \cdot)_{e} = r^{(e)T} \int_{\Omega_e} N^{(e)T} (\cdot) d\Omega_e, (\mathbf{V}_f, \cdot)_{e} = r^{(e)T} \int_{\Omega_e} \mathbf{dN}^{(e)T} (\cdot) d\Omega_e.
\] (A.28)
The time derivative \( \mathbf{p} (x) \) can be written in the form of Eq. (A.8), by substituting \( r(x) \) and \( r^{(e)} \) with \( \mathbf{p} (x) \) and \( \mathbf{p}^{(e)} \). The gradient \( \mathbf{V}_f (x) \) can be written in the form of Eq. (A.26) by substituting \( r(x) \) and \( r^{(e)} \) with \( \mathbf{p} (x) \) and \( \mathbf{p}^{(e)} \). The strain rate \( \dot{e} \) is given by Eq. (A.15), by substituting \( \mathbf{u} \) with \( \mathbf{u} \). Very integral in Eq. (A.25), shown by Eq. (A.28), would result a left multiplication term \( r^{(e)T} \). Again, because of the arbitrariness of the test function, the equation should hold before the left multiplication. What need to be integrated are
\[
\int_{\Omega_e} \mathbf{dN}^{(e)T} \mathbf{k} (\mathbf{dN}^{(e)}) d\Omega_e = \mathbf{K}^{(e)} = \int_{\Omega_e} N^{(e)T} \mathbf{a}_{\mathbf{q}}^{(e)} B^{(e)} \mathbf{dN}^{(e)} = \mathbf{H}^{(e)}, \int_{\Omega_e} N^{(e)T} \mathbf{N}^{(e)} d\Omega_e = \mathbf{S}^{(e)},
\] (A.29)
where, \( a_{\mathbf{q}}^{(e)} \) is formed by stacking \( a_{\mathbf{q}} \) (identical within one element) for each node of \( e \) horizontally, the coupling matrix \( \mathbf{H}^{(e)} \) is defined this way to state consistent with Eq. (2) and the code. The total compressibility is given by
\[
c_i = (1 - a_{\mathbf{q}})(a_{\mathbf{q}} - \phi)/k_x + \phi/k_f,
\] (A.30)
where, \( k_x \) and \( k_f \) are bulk moduli for the solid and fluid respectively.

The element integral equation becomes
\[
-\mathbf{H}^{(e)} \mathbf{u}_e^{(e)} + \mathbf{S}^{(e)} \mathbf{p}_e^{(e)} + \mathbf{K}^{(e)} \dot{\mathbf{p}}_e^{(e)} = (\mathbf{N}^{(T,e)} q_{,e}^{(T,e)}, f_{,e}^{(T,e)})_{\Omega_e} + (\mathbf{N}^{(T,e)} f_{,e}^{(T,e)}, f_{,e}^{(T,e)})_{\partial \Omega_e}.
\] (A.31)
Applying the first order time scheme, i.e. \( \mathbf{p}_e = (\mathbf{p}_e - \mathbf{p}_{e-1})/\Delta t \) and \( \mathbf{u}_e = (\mathbf{u}_e - \mathbf{u}_{e-1})/\Delta t \), Eq. (A.31) becomes
\[
-\mathbf{H}^{(e)} (\mathbf{u}_e^{(e)} - \mathbf{u}_{e-1}^{(e)})/\Delta t + \mathbf{S}^{(e)}((\mathbf{p}_e^{(e)} - \mathbf{p}_{e-1}^{(e)})/\Delta t + \mathbf{K}^{(e)} \dot{\mathbf{p}}_e^{(e)} = (\mathbf{N}^{(T,e)} q_{,e}^{(T,e)}, f_{,e}^{(T,e)})_{\Omega_e} + (\mathbf{N}^{(T,e)} f_{,e}^{(T,e)}, f_{,e}^{(T,e)})_{\partial \Omega_e}.
\] (A.32)
Note that, a higher order time scheme for \( p \) is impossible for Eq. (A.23), becomes the second order time derivative \( \dot{p} \) has been already neglected. Again, we combine the source/flux terms as a global vector, denoted by \( \mathbf{q} \), in Eq. (2), instead of doing the integrations. This is done by multiplying the facet-wise (element-wise) flux (source) by the facet area (element volume) and evenly distribute the value to the associated nodes. In addition, one can specify the nodal flow rate directly.

The global incremental equation, after denoting \( \Delta \mathbf{u}_e = \mathbf{u}_e - \mathbf{u}_{e-1} \), and \( \Delta \mathbf{p}_e = \mathbf{p}_e - \mathbf{p}_{e-1} \), is given as:
\[
-\mathbf{H}^{(T)} \Delta \mathbf{u}_e + (\mathbf{S}^{(T)} + \Delta \mathbf{K}^{(T)}) \Delta \mathbf{p}_e = -\Delta \mathbf{K}^{(T)} \Delta \mathbf{p}_{e-1} + q_e.
\] (A.33)
Note that, the block structured matrix in Eq. (2) only exits temporarily and element-wise, but we hide the superscript to be concise.

When the pore pressure acts on the solid, it will result an additional stress term,
\[
\sigma_{\text{total}} = \sigma - a_{\mathbf{q}} \mathbf{p}
\] (A.34)
where, the Biot’s coefficient should be tensor shaped, \( a_{\mathbf{q}} = a_{\mathbf{q}}^{(T)} I_{3,3} \) in 3D, to make the product conforming. Shown later however, the Biot’s coefficients will all appear as \( a_{\mathbf{q}}^{(T)} \), e.g. in Eq. (A.29). Therefore, without causing ambiguity, we keep the notation unchanged.

The contribution of this stress to the RHS of Eq. (A.1) is \( -\nabla \cdot \mathbf{a}_{\mathbf{q}} \mathbf{q} \). Multiplied by the test function \( r \) and integrated by part, this term becomes
\[
(r, \nabla \cdot \mathbf{a}_{\mathbf{q}} \mathbf{q})_{\Omega} = (\mathbf{V}_f, \mathbf{a}_{\mathbf{q}} \mathbf{q})_{\Omega} - (r, (\mathbf{a}_{\mathbf{q}} \mathbf{q})_{\Omega} - (r, f)_{\partial \Omega}.
\] (A.35)
The second term in Eq. (A.35) is taken care of by the pressure boundary condition, a part of \( f \), in Eq. (A.22), instead of doing the integration. \( \mathbf{V}_f \) is given by Eq. (A.9), and \( p \) is given by Eq. (A.8), after substituting \( r(x) \) and \( r^{(e)} \) with \( \mathbf{p}(x) \) and \( \mathbf{p}^{(e)} \). The first term in Eq. (A.35), element-wise, becomes
\[
(\mathbf{V}_f, \mathbf{a}_{\mathbf{q}} \mathbf{q})_{\Omega} = r^{(e)T} \int_{\Omega_e} \mathbf{B}^{(e)T} a_{\mathbf{q}}^{(e)} \mathbf{N}^{(e)} d\Omega_e \mathbf{p}^{(e)}.
\] (A.36)
What needs to be integrated, comparing Eq. (A.36) to Eq. (A.29), is actually \( \mathbf{H} \). The global integral equation for the solid becomes
\[
\mathbf{K} \Delta \mathbf{u}_e + \mathbf{H} \Delta \mathbf{p}_e = \mathbf{f}_e,
\] (A.37)
where, \( K_r \) denotes the elastic stiffness matrix. After substituting \( \mathbf{u}, \mathbf{p} \) and \( f \) with \( \Delta \mathbf{u}, \Delta \mathbf{p} \) and \( \Delta f \), Eq. (A.37) becomes a incremental equation,
\[ \text{K}_e \Delta u_e + \text{H} \Delta p_e = \Delta f_e. \quad (A.38) \]

Writing Eq. (A.38) on top of Eqs. (A.33), (2) is then obtained. Note that, the stress resolved by this formulation, is the solid stress instead of the total stress.

The pressure stabilization scheme in Eq. (3), Bochev and Dohrmann, 2006, is equivalent to adding an additional term to the RHS of Eq. (A.25).

\[ q(x)_{n+1} = (N^o(x) - (1/n_e) I)q_e(\Delta p_{n+1}^{(e)} + p_e)^e h(2G). \]

where, \( N^o(x) - (1/n_e) I \) is a zero-sum shape function which ensures the mass balance. Implementing the same treatment as to the other terms in Eq. (A.25), substituting the shape function with the zero-sum one, Eq. (3) is then obtained.

**Appendix B. Prepare and run defmod model**

To download the unmodified Defmod, execute the Mercurial command `hg clone https://bitbucket.org/stali/defmod` To download the modified Defmod, Meng (2015), execute the Mercurial command `hg clone https://bitbucket.org/chunfangmeng/defmod-dev` Compilation of Defmod requires a PETSc installation, detail described in the file INSTALL.

A single input (.inp) file contains both the mesh data and model parameters. To run a model `foo.inp`, execute command:

```
(path/to/defmod)/defmod -f foo.inp
```

Defmod takes in PETSc arguments. To use the direct solver MUMPS for example, run the model with the PETSc option

```
--pc_type lu --pc_factorizationsolverpackage mumps
```

The python code `Mandel3D_prepro.py` in example/Mandel/Mandel3D provides an example of preparing the .inp file, for an unconstrained model. This code requires installation of the meshing tool Cubit/Trelis Sandia National Lab, 2012. To generate the input file `Mandel3D.inp`, run:

```
./Mandel3Dpreproc.py
```

This code generates the mesh, specifies the model parameters and write out the .inp file.

The python codes `Fault3dhex.py` and `Fault2dqua.py` in `example/Fault3d/fault3dhyb` and `example/Fault2d/fault2dhyb` provide examples of preparing the .inp files, for fault models in 3D and 2D. These two codes do not require Cubit/Trelis. Instead, they require netCDF4-python installation, [http://unidata.github.io/netcdf4-python/](http://unidata.github.io/netcdf4-python/), and mesh (.exo) files in Exodus II format. To generate the input files `3D_flt_cur.inp` and `2D_flt_cur.inp`, run:

```
./Fault3dhex.py 3D_flt_cur.exo
```

```
./Fault2dqua.py 2D_flt_cur.exo
```

The resulting .inp files can be viewed by any text editor. The meaning of each data block is described in the python codes that generate the .inp files. Users are encouraged to modify the python and Cubit journals to build their own specific models. Note that, in order to run a model generated by the python code, one needs to use the modified Defmod.

**Appendix C. Duplicate the benchmark results**

Assuming one has completed the compiling process following `INSTALL`, all the benchmark results presented here can then be duplicated. Go to `(path/to/defmod)/example`, and execute command lines as follows.

**C.1. Mandel benchmark**

Go to Mandel, execute the commands

```
[MPI options] ./defmod -f Mandel2D/Mandel2D.inp [PETSc options] [MPI options] ./defmod -f Mandel3D/Mandel3D.inp [PETSc options]
```

and run the python code

```
./Mandelbench.py
```

to plot the pressure comparisons, Figs. 3 and 4. In addition to the pressure, the displacement also has analytical form, Kurashige et al. (2005). One can modify the code `Mandel_bench.py` to benchmark the displacement as well.

**C.2. Mohr-Coulomb benchmark**

Go to `Fault3d_hyb`, make sure the preprocessing python code `Fault3Dhex.py` has the variables set as follow.

```python
...fault = True
poro = False
visc = False...t = 1500000. ;dt = 100000. ;nviz = 1...fc = .6*np.ones((len(vecf), 1))
```

Run the preprocessing code

```
./Fault3Dhex.py 3D_flt_cur.exo
```

Input file `3D_flt_cur.inp` is then produced. Run the model, view the .vtk files using Paraview Henderson, 2007, and run the python code

```
./defmodCoulomb.py 3D_flt_cur
```
to make the Mohr-Coulomb circle plot Fig. 8.

C.3. SCEC TPV205 benchmark

Go to example/tpv5, and run the meshing script with Cubit/Trelis
trelis – nographics – nojournalTPV205.jou
to generate the mesh file TPV205.exo. Run preprocessing command
/TPV205.pyTPV205.exo

to generate the input file TPV205.inp. Check the screen output for the problem size, and run the model TPV205.inp. Note that, the result presented here was produced on a cluster computer using 128 processors. The runtime was within 4 min. For different mesh size, set by TPV205.jou, and different number of processors, the runtime would vary.

Sort the output data by command.
/ defsort.pyTPV205slip

A sorted file TPV205.mat is then produced. Run the waveform plotting script
/TPV205plot.pyTPV205slip
to produce Figs. 14 and 17. The temporal slip rate on the fault can be animated by running the script
/ defplot.pyTPV205slip

C.4. Additional (quasi-)static benchmarks

Here, we present a viscoelastic benchmark by modeling deformation due to 1.0 m slip on a 100 km long strike-slip fault that extends from the surface to a depth of 25 km. The elastic crust is 25 km thick and overlies a 225 km thick viscoelastic mantle that has a Maxwell viscosity of $\eta = 10^{18}$ Pa s. The model domain is 500 km by 500 km by 250 km, Fig. C.1.

Fig. C.2 compares the displacement in strike direction against Okada (1985), Relax, Barbot and Fialko (2010), and Abaqus solutions. Okada and Relax are for infinite half space, whereas Defmod and Abaqus are for finite space. Due to this difference, the figure shows disagreement. Note that, Okada does not consider viscoelasticity, so only time zero solution is available.

We present a non-conformal approach for the cylindrical Mandel benchmark to see how the localized mesh refinement would improve the agreement. Fig. C.3 shows the mesh of a cylindrical domain with two refined regions near the perimeter of the cylinder, leaving two non-conformal interfaces. If we treat the non-conformal interfaces as locked and permeable faults, we can effectively constrain the hanging nodes.

Since the singularity of the t=0 solution, we only compare the pressure starting from non-zero time. Fig. C.4 compares the model pressures, along the radius at the domain bottom, against the analytical Mandel solution. This comparison suggests that the refined (non-conformal) model produces the fluid pressure closer to the analytical solution, not only at the perimeter but across the whole radius. Note, due to the limitation of the FE tool, we can only improve the agreement at cost of the problem size.

C.5. SCEC benchmark TPV102

Following the same procedure, one can solve TPV102 for the rate and state friction law. Here we compare the results against an FE code Pylith (Bochev and Dohrmann, 2006) and a FD code DFM Dalguer and Day (2006).

Fig. C.1. Part of the finite element mesh used for validation of quasistatic viscoelastic relaxation. Colors represent fault parallel displacements, at t=0 years, provided by Ali (2014).
Fig. C.5 compares the waveforms recorded at the station 1, 6 km off the fault plane, 12 km off the epicenter along the strike (y) direction. Due to symmetry, the normal (x) and vertical (z) components are almost zero Fig. C.6.

Fig. C.7 compares the rupture front contours. Again, both waveforms and rupture contours suggest that Defmod agree with the FD code DFM better than with the FE code Pylith.
References


