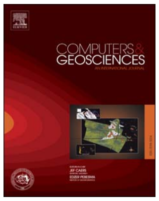
Computers & Geosciences 100 (2017) 10–26

Contents lists available at ScienceDirect

Computers & Geosciences

journal homepage: www.elsevier.com/locate/cageo



Research paper

Benchmarking Defmod, an open source FEM code for modeling episodic fault rupture

Chunfang Meng

Earth Resources Laboratory, Massachusetts Institute of Technology, Cambridge, MA, USA

ARTICLE INFO

Keywords:
Induced earthquake
Finite element
Fault
Poroelasticity
Dynamic rupture
Defmod

ABSTRACT

We present Defmod, an open source (linear) finite element code that enables us to efficiently model the crustal deformation due to (quasi-)static and dynamic loadings, poroelastic flow, viscoelastic flow and frictional fault slip. Ali (2015) provides the original code introducing an implicit solver for (quasi-)static problem, and an explicit solver for dynamic problem. The fault constraint is implemented via Lagrange Multiplier. Meng (2015) combines these two solvers into a hybrid solver that uses failure criteria and friction laws to adaptively switch between the (quasi-)static state and dynamic state. The code is capable of modeling episodic fault rupture driven by quasi-static loadings, e.g. due to reservoir fluid withdraw or injection. Here, we focus on benchmarking the Defmod results against some establish results.

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 finite element method, Zienkiewicz (2000), provides a system of linear equations describing the (quasi-)static state. Eq. (1) lists the absolute and incremental versions of the linear equation, Smith and Griffiths, 2004.

$$\mathbf{K}_n\mathbf{U}_n = \mathbf{F}_n,\ \text{absolute},\ \mathbf{K}_n\Delta\mathbf{U}_n = \Delta\mathbf{F}_n,\ \text{incremental}. \tag{1}$$

where, $\mathbf{K}$ is the system stiffness matrix, $\mathbf{U}$ is the solution vector and $\mathbf{F}$ 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 $\Delta\mathbf{U}_n$ of a poroelastic problem contains the nodal displacement and pressure, $\Delta\mathbf{U}_n = \begin{bmatrix}\Delta\mathbf{u}_n\\ \Delta\mathbf{p}_n\end{bmatrix}$, whereas the solution for an elastic problem is only the nodal displacement.

The stiffness matrix $\mathbf{K_n}$ and RHS function $\Delta\mathbf{F}_n$ are also different for the elastic and poroelastic problems, Eq. (2).

$$(\mathbf{K}_n,\ \Delta\mathbf{F}_n) = \begin{cases} (\mathbf{K}_e,\ \Delta\mathbf{f}_n) & \text{, elastic.}\\ \left(\begin{bmatrix}\mathbf{K}_e & \mathbf{H}\\ -\mathbf{H}^T & \Delta t\mathbf{K}_c + \mathbf{S}_p\end{bmatrix}, \begin{bmatrix}\Delta\mathbf{f}_n\\ \mathbf{q}_n - \Delta t\mathbf{K}_c\mathbf{p}_{n-1}\end{bmatrix}\right) & \text{, poroelastic.}\end{cases} \tag{2}$$

where, $\mathbf{K}_e$ is the elastic stiffness matrix, depending on the elastic constants of the solid. $\mathbf{K}_c$ is the fluid stiffness matrix, depending on the fluid flow conductivity. $\mathbf{H}$ is the coupling matrix, depending on the Biot's coefficient. $\mathbf{S}_p$ 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 $\mathbf{K}_n$ is constant for evenly spaced time step Δt. In a later section, we show that for Newtonian viscoelasticity, $\mathbf{K}_n$ is, although modified, still independent of time. $\mathbf{f}_n$ and $\mathbf{q}_n$ 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,

$$\mathbf{F}_{n+1} = \mathbf{F}_{n+1} - \begin{bmatrix}\mathbf{0}\\ \mathbf{H}_s\mathbf{p}_n\end{bmatrix}, \mathbf{K}_{n+1} = \mathbf{K}_{n+1} + \begin{bmatrix}\mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{H}_s\end{bmatrix}, \text{where}$$
$$, \mathbf{H}_s = \int_\Omega (\mathbf{N} - (1/n_e)I_\mathbf{N})^\mathrm{T}(\mathbf{N} - (1/n_e)I_\mathbf{N})/(2G)d\Omega, \tag{3}$$

E-mail address: cmeng@mit.edu.

http://dx.doi.org/10.1016/j.cageo.2016.11.014
Received 7 July 2016; Received in revised form 27 November 2016; Accepted 29 November 2016
Available online 02 December 2016

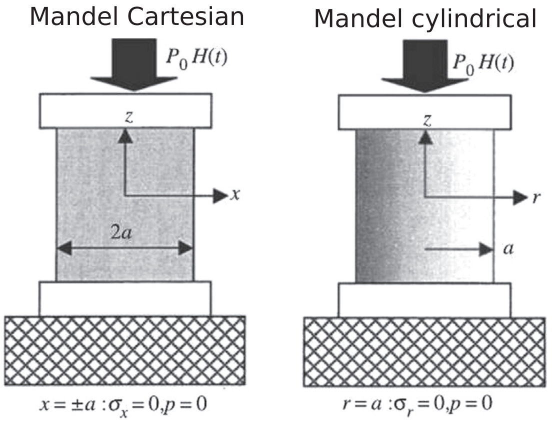


Fig. 1. Schematics of the Mandel benchmark problem by Kurashige et al. (2005).

N is the shape function of an element e, n_e is the node count of the element, and G is the shear modulus.

To benchmark the poroelastic model, we compare the Defmod 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 matrix suddenly has the 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 values 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 pressure 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 mentioned, the models only show some spatial oscillation near the side boundaries at time zero. The rigid loading plate can be replaced by uniform loading to avoid the contact problem. This requires the matrix aspect ratio height /a to be large enough such that the bottom, at z=0, will not *feel* the tilted deformation at the top.

The poroelastic model parameters are listed in Table 1

Fig. 2 plots the initial pressure of a 2D and a 3D poroelastic models approximating the Cartesian and cylindrical Mandel problems respectively. Because of the symmetry, we only need to consider half of the Cartesian domain and a quarter of the cylindrical domain in Fig. 1.

The numerical pressure should be normalized by p_0 to be comparable with the Mandel pressure, Kurashige et al. (2005). A quasi-static model always produces solutions of some stress equilibrium states. Therefore, the model, being continuum, cannot produce the theoretical initial pressure which has a sharp pressure drop, infinite gradient, at x=a. To resolve this, we sync the model pressure, at r=0, z=0, $t = t_1$, $p_{\text{model}}(0, 0, t_1)$ with the Mandel pressure, at r =0, $t = t_1$, $p_{\text{Mandel}}(0, t_1)$, i.e. multiply all the model pressures by $\frac{p_{\text{Mandel}}(0,t_1)}{p_{\text{model}}(0,0,t_1)}$, t_1 being the first/smallest nonzero model time.

Fig. 3 plots the normalized pressure as a function of r/a by the 2D and 3D models, at z=0, against the Cartesian and cylindrical Mandel solutions respectively. Where the normalized time τ_t is given by $\tau_t = \frac{tK}{Sa^2}$, 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 $\tau_t = 0$ than at later times.

Table 1
Poroelastic model parameters: Young's modulus E, Poison's ratio ν, fluid mobility k, Biot's coefficient α_B, fluid bulk modulus k_f.

E [Pa]	ν	k [m^2/Pa/s]	α_B	k_f [Pa]
3.0E10	0.25	1.0E-12	0.96	2.2E9

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 (Mendel 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 $\mathbf{K}_n$ and the RHS function $\Delta\mathbf{F}_n$, Eqs. (4) and (5) by Melosh and Raefsky, 1980.

$$\mathbf{K}_e^{(n+1)} = \int_\Omega \mathbf{B}^T(\mathbf{D}^{-1} + \alpha\Delta t\boldsymbol{\beta}'_n)^{-1}\mathbf{B}d\Omega\mathbf{f}_{n+1} = \int_\Omega \mathbf{B}^T(\mathbf{D}^{-1} + \alpha\Delta t\boldsymbol{\beta}'_n)^{-1}(\Delta t\boldsymbol{\beta}_n) d\Omega + \mathbf{F}_{n+1}. \tag{4}$$

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.

$$\boldsymbol{\beta} = \frac{\sigma^{e-1}}{4\eta}\mathbf{C}_c: \boldsymbol{\sigma},\ \ e \geq 1,\ \sigma = \sqrt{\sum_{i\neq j}(\boldsymbol{\sigma}_{ii} - \boldsymbol{\sigma}_{jj})^2/(2d) + \boldsymbol{\sigma}_{ij}^2},\quad d = 2 \text{ or } 3,$$

$$\mathbf{C}_c = \begin{cases} \begin{bmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 4 \end{bmatrix}, & \text{2D,} \\ \begin{bmatrix} 4/3 & -2/3 & -2/3 & & \\ -2/3 & 4/3 & -2/3 & 0 & \\ 2/3 & -2/3 & 4/3 & & \\ & & & 4 & \\ & 0 & & & 4 \end{bmatrix}, & \text{3D.} \end{cases} \tag{5}$$

$$\boldsymbol{\beta}' = \begin{cases} \frac{\sigma^{e-1}}{4\eta}\begin{bmatrix} c_1 & -c_1 & c_3 \\ -c_1 & c_1 & -c_3 \\ c_3 & -c_3 & 4c_2 \end{bmatrix}, & \text{2D,} \\ \frac{\sigma^{e-1}}{4\eta}\begin{bmatrix} 4/3 + S_x^2 & -2/3 + S_xS_y & -2/3 + S_xS_z & S_xT_1 & S_xT_2 & S_xT_3 \\ -2/3 + S_xS_y & 4/3 + S_y^2 & -2/3 + S_yS_z & S_yT_1 & S_yT_2 & S_yT_3 \\ -2/3 + S_xS_z & -2/3 + S_yS_z & 4/3 + S_z^2 & S_zT_1 & S_zT_2 & S_zT_3 \\ S_xT_1 & S_yT_1 & S_zT_1 & 4 + T_1^2 & T_1T_2 & T_1T_3 \\ S_xT_2 & S_yT_2 & S_zT_2 & T_2T_1 & 4 + T_2^2 & T_2T_3 \\ S_xT_3 & S_yT_3 & S_zT_3 & T_3T_1 & T_3T_2 & 4 + T_3^2 \end{bmatrix}, & \text{3D.} \end{cases}$$

where, e is the viscosity power law parameter. When e=1, η is the linear

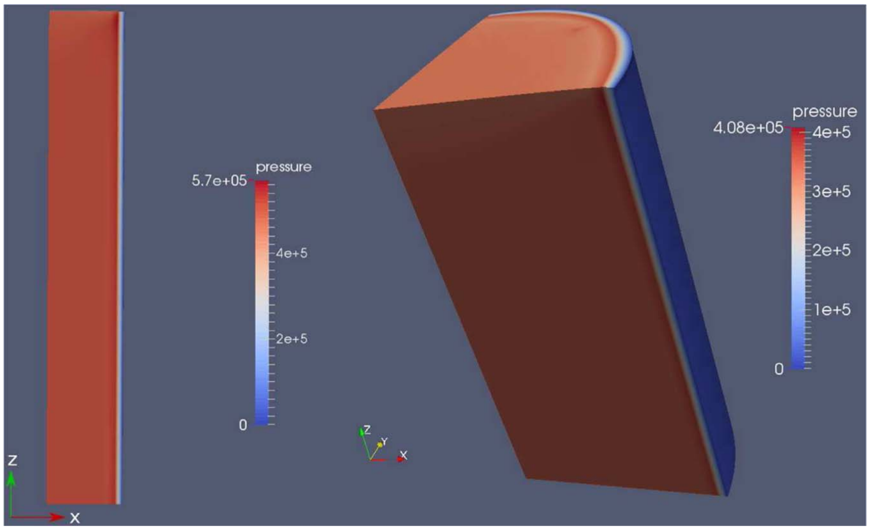


Fig. 2. 2D numerical model for Cartesian Mandel problem.

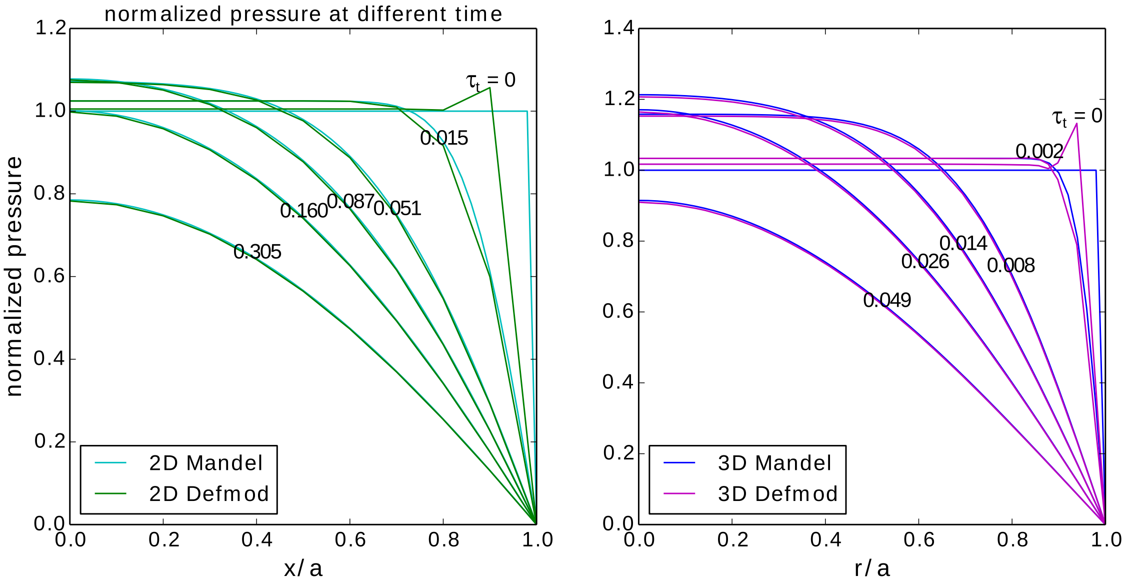


Fig. 3. Normalized pressure comparison between 2D model and Cartesian Mandel solution, left, and between 3D model and cylindrical Mandel solution, right.

viscosity. The stress dependent matrix entries are formulated by,

$$c_1 = 1 + (e - 1)((\boldsymbol{\sigma}_{xx} - \boldsymbol{\sigma}_{yy})/(2\sigma))^2,\ c_2 = 1 + (e - 1)(\boldsymbol{\sigma}_{xy}/\sigma)^2,\ c_3 = (e - 1)(\boldsymbol{\sigma}_{xx}\boldsymbol{\sigma}_{yy} - \boldsymbol{\sigma}_{yy}\boldsymbol{\sigma}_{xy})/\sigma^2,\ S_x = c\,(2\sigma_{xx} - \sigma_{yy} - \sigma_{zz})/(3\sigma),$$
$$\text{where } c = \sqrt{e - 1}\,.S_y = c\,(2\sigma_{yy} - \sigma_{zz} - \sigma_{xx})/(3\sigma),\ S_z = c\,(2\sigma_{zz} - \sigma_{xx} - \sigma_{yy})/(3\sigma),\ T_1 = 2c\sigma_{xy}/\sigma,\ T_2 = 2c\sigma_{yz}/\sigma,\ T_3 = 2c\sigma_{xz}/\sigma, \tag{6}$$

To model visco-poroelasticity, only the submatrix $\mathbf{K}_e$ and the subvector $\mathbf{f}_n$ 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 e=1, i.e. the medium is Newtonian viscoelastic, we have a constant $\boldsymbol{\beta}' = \frac{1}{4\eta}\mathbf{C}_c$. 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

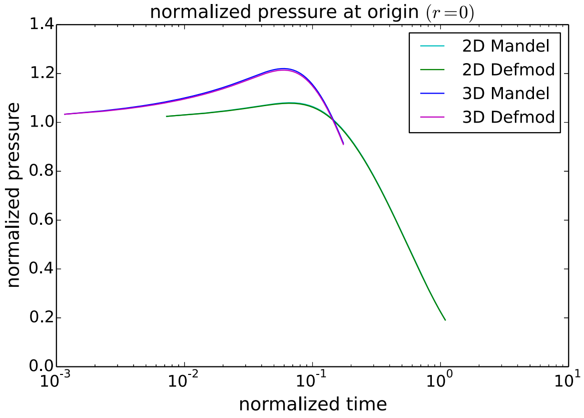


Fig. 4. Normalized pressure at $z = 0$, $r = 0$ versus normalized time (τ_t) by 2D model, Cartesian Mandel solution, 3D model and cylindrical Mandel solution.

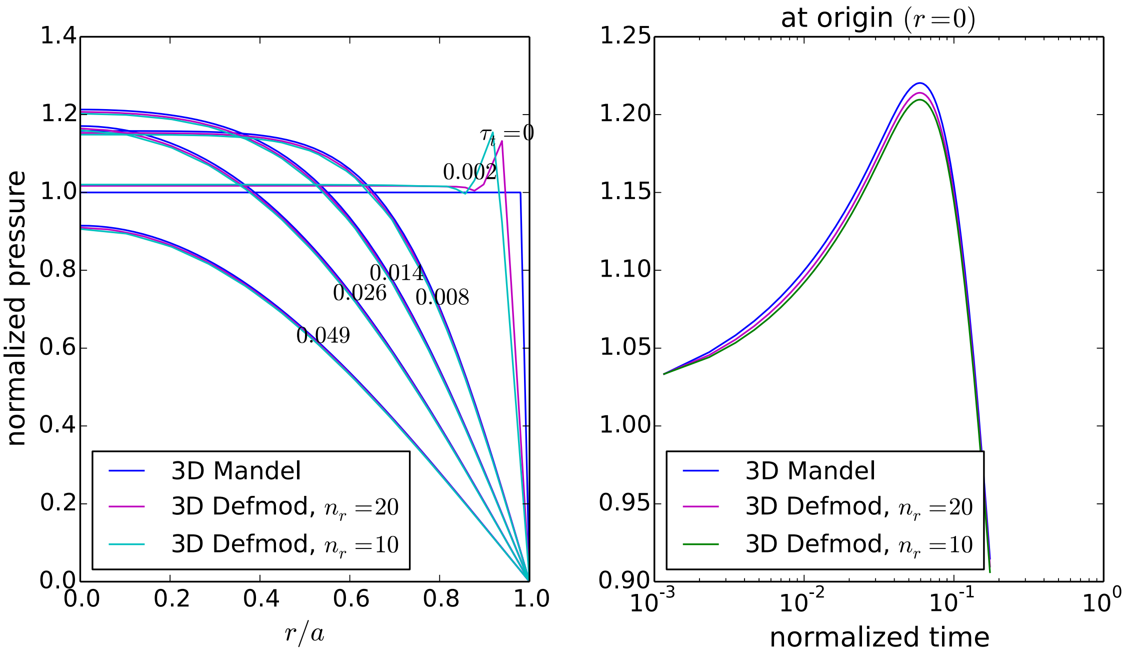


Fig. 5. Normalized pressure from a same 3D model of different mesh resolutions, 10 cells along radius n_r=10 and 20 cells along radius n_r=20, and analytical result. Left, pressure profile along radius; right, pressure at origin as a time function.

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 $(n_1, \ n_3)$ in Fig. 6 is given by Eq. (7)

$$\begin{bmatrix} n_x & n_z & -n_x & -n_z \\ t_x & t_z & -t_x & -t_z \end{bmatrix}\begin{bmatrix} u_x^{(1)} \\ u_z^{(1)} \\ u_x^{(3)} \\ u_z^{(3)} \end{bmatrix} = \mathbf{0}, \text{ for locked fault;} [n_x \quad n_z \quad -n_x \quad -n_z]\begin{bmatrix} u_x^{(1)} \\ u_z^{(1)} \\ u_x^{(3)} \\ u_z^{(3)} \end{bmatrix} = \mathbf{0}, \text{ for slipping fault.} \tag{7}$$

where, $\mathbf{u}^{(\cdot)}$ 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.

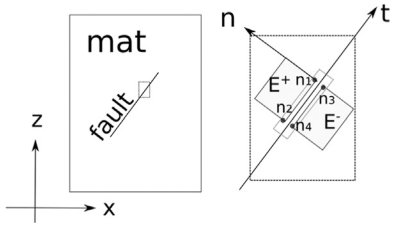


Fig. 6. Fault constraint: node n_1 coincides with node n_3; node n_2 coincides with node n_4; the vector $\mathbf{n}$ and $\mathbf{t}$ are the fault's normal and tangent vectors respectively.

$$\begin{bmatrix} n_x & n_z & 0 & -n_x & -n_z & 0 \\ t_x & t_z & 0 & -t_x & -t_z & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} u_x^{(1)} \\ u_z^{(1)} \\ p^{(1)} \\ u_x^{(3)} \\ u_z^{(3)} \\ p^{(3)} \end{bmatrix} = \mathbf{0}, \text{ locked, permeable}$$

$$\begin{bmatrix} n_x & n_z & 0 & -n_x & -n_z & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} u_x^{(1)} \\ u_z^{(1)} \\ p^{(1)} \\ u_x^{(3)} \\ u_z^{(3)} \\ p^{(3)} \end{bmatrix} = \mathbf{0}, \text{ slipping, permeable} \tag{8}$$

where, $p^{(\cdot)}$ 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.

$$\begin{bmatrix} \mathbf{K} & \mathbf{G}^T \\ \mathbf{G} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \Delta \mathbf{U}_n \\ \lambda_n \end{bmatrix} = \begin{bmatrix} \Delta \mathbf{f}_n \\ \mathbf{I}_n \end{bmatrix}. \tag{9}$$

where the Lagrange Multiplier, λ_n, is the nodal forces/sources needed to honor the constraint equation. Each row of the constraint matrix $\mathbf{G}$ has two exact opposite nonzero entries, Eqs. (7), (8), so λ_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)

$$\mathbf{f}^{(1)} = -\mathbf{f}^{(3)} = \begin{bmatrix} \lambda_{\mathbf{n}}^{(1,3)} \\ \lambda_{\mathbf{t}}^{(1,3)} \end{bmatrix} = \begin{bmatrix} \lambda_{\mathbf{n,x}}^{(1,3)} \\ \lambda_{\mathbf{n,z}}^{(1,3)} \\ \lambda_{\mathbf{t,x}}^{(1,3)} \\ \lambda_{\mathbf{t,z}}^{(1,3)} \end{bmatrix}. \tag{10}$$

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).

$$\frac{\tau^{(1,3)}}{\sigma_{\boldsymbol{n}}^{(1,3)}} = \frac{\| \boldsymbol{\lambda}_{\mathbf{t}}^{(1,3)} \|}{\| \boldsymbol{\lambda}_{\mathbf{n}}^{(1,3)} \|}. \tag{11}$$

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.

$$\begin{aligned}\boldsymbol{\sigma}^{(E^+)} = \boldsymbol{D}^{(E^+)}\boldsymbol{B}^{(E^+)}\colon \boldsymbol{u}^{(E^+)}\, \frac{\tau^{(1)}}{\sigma_{\boldsymbol{n}}^{(1)}} &= \frac{(\boldsymbol{T}\boldsymbol{\sigma}^{(1)}\boldsymbol{T}^T)_{12}}{(\boldsymbol{T}\boldsymbol{\sigma}^{(1)}\boldsymbol{T}^T)_{22}}, \text{ at node 1}\\ \frac{\tau^{(2)}}{\sigma_{\boldsymbol{n}}^{(2)}} &= \frac{(\boldsymbol{T}\boldsymbol{\sigma}^{(2)}\boldsymbol{T}^T)_{12}}{(\boldsymbol{T}\boldsymbol{\sigma}^{(2)}\boldsymbol{T}^T)_{22}}, \text{ at node 2}\end{aligned} \tag{12}$$

where, the matrices $\boldsymbol{B}$ and $\boldsymbol{D}$ are the same as in Eq. (4) and Appendix A Matrix $\boldsymbol{T}$ translates the stress tensor from the (x,z) coordinates to $(\boldsymbol{t}, \boldsymbol{n})$ coordinates in Fig. 6. If the fault is curved, the translation matrix $\boldsymbol{T}$ varies with position. Stress tensor $\boldsymbol{\sigma}^{(1)}$ and $\boldsymbol{\sigma}^{(2)}$ are subsets of the element stress $\boldsymbol{\sigma}^{(E^+)}$, corresponding to nodes 1 and 2 respectively.

The coinciding node pairs belong to different elements. For instance, calculation for the stress at the nodes 1, 2 in Fig. 6 does not involve the nodal displacement of the element E^-. Likewise, the stress at nodes 3 and 4 are calculated without considering element E^+. The stress, by Eq. (12), is more of an element value than nodal value, although they are equivalent when an element collapses to a dot. The fault stress would not be very accurate, if either the element is large, or the stress gradient is high.

Fig. 7 illustrates the anisotropic loading model. At the start, we incrementally load the sample at the rate of $\Delta\sigma_H = \Delta\sigma_V = 0.2$ 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 α, and associated stress τ and $\sigma_{\mathbf{n}}$. 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 τ and $\sigma_{\mathbf{n}}$ 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.

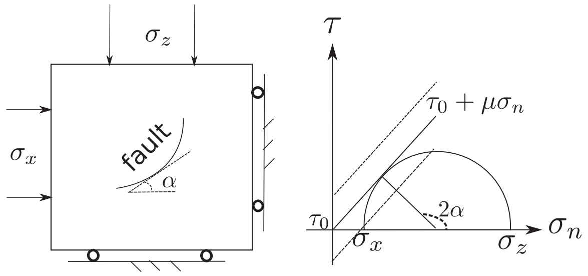


Fig. 7. Schematics of a curved fault embedded in an elastic domain under anisotropic confining stress, left; the Mohr-Coulomb circle varying with the dip angle, right. A positive τ-truncation ($\tau_0 > 0$) is usually associated with the cohesion, and the negative truncation is usually associated with the pressurized pore space.

Fig. 9 compares the stress ratio $\tau/\sigma_{\mathbf{n}}$ 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 $\beta = 60°$. We apply gravity at start and gradually add compressional traction on the top, $\dot{\sigma}_{zz} = 0.2$ MPa/yr. The bottom and side walls have zero normal displacement. Fig. 10 plots the stress $(\tau, \sigma_{\mathbf{n}})$ and associated Mohr-Coulomb circles. In this example, analytical stress are $\sigma_{zz} = \rho gz + \dot{\sigma}_{zz}t$ and $\sigma_{xx} = \sigma_{yy} = \frac{\nu}{1-\nu}\sigma_{zz}$. For $\nu = 0.25$, all the stress evaluations $(\tau, \sigma_{\mathbf{n}})$ should appear on a straight line that passes the origin, and has $\sqrt{3}/3$ slope, $\theta = 30°$. 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).

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f}, \tag{13}$$

where, $\mathbf{M}$ is the mass matrix; $\mathbf{C} = \alpha_\eta\mathbf{M} + \beta_\eta\mathbf{K}$ is the damping matrix; α_η and β_η are the Rayleigh damping coefficients. The detailed matrices and RHS vector formulations are given in Appendix A.

Newmark explicit scheme has Eq. (14).

$$\mathbf{u}_n = \mathbf{M}^{-1}(\Delta t^2(\mathbf{f}_n - \mathbf{K}\mathbf{u}_{n-1}) - \Delta t\mathbf{C}(\mathbf{u}_{n-1} - \mathbf{u}_{n-2})) + 2\mathbf{u}_{n-1} - \mathbf{u}_{n-2} \tag{14}$$

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

$$\begin{aligned}\Delta\mathbf{u}_n = \mathbf{M}^{-1}(\Delta t^2(\Delta\mathbf{f}_n - \mathbf{K}\Delta\mathbf{u}_{n-1}) - \Delta t\mathbf{C}(\Delta\mathbf{u}_{n-1} - \Delta\mathbf{u}_{n-2})) + 2\Delta\mathbf{u}_{n-1}\\ - \Delta\mathbf{u}_{n-2}\end{aligned} \tag{15}$$

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

$$\boldsymbol{\lambda}_n = (\Delta t^2\mathbf{G}\mathbf{M}^{-1}\mathbf{G}^T)^{-1}(\mathbf{G}\Delta\mathbf{u}_n - \mathbf{I}_n)\Delta\mathbf{u}_n = \Delta\mathbf{u}_n - \Delta t^2\mathbf{M}^{-1}\mathbf{G}^T\boldsymbol{\lambda}_n, \tag{16}$$

where the Lagrange Multiplier $\boldsymbol{\lambda}_n$ is the nodal force needed to constrain the solution. λ 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. λ component tangent to the fault face, with the product of friction coefficient and normal stress, i.e. λ 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.

Lysmer 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.

$$c_{ii} = \begin{cases} c_{ii} + \int_\Gamma \rho V_p d\Gamma, & \text{p wave}\|i^{th}\ \textbf{axis}, \\ c_{ii} + \int_\Gamma \rho V_s d\Gamma, & \text{p wave}\perp i^{th}\text{axis}, \end{cases} \text{for all} i - \text{axes}. \tag{17}$$

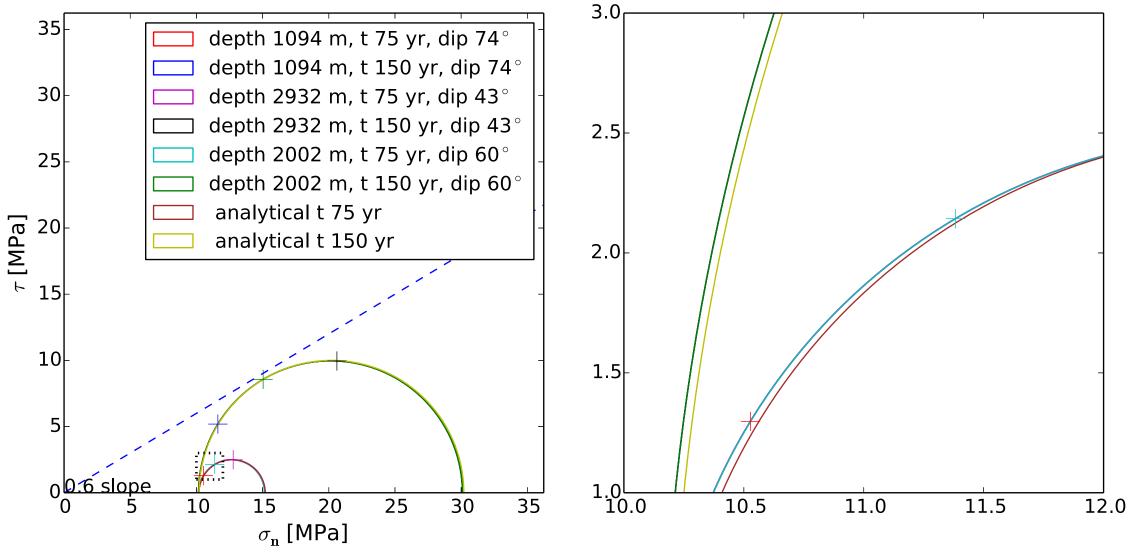


Fig. 8. Mohr-Coulomb circles by Defmod against the analytical results. The right is an amplified plot of the left enclosed by the square.

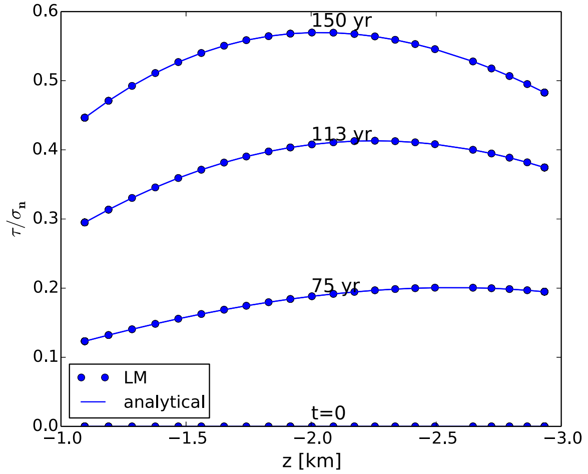


Fig. 9. Stress ratio τ/σ_n calculated by Defmod at different depth and time against the analytical values.

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, τ and σ_n satisfies some failure criterion, the dynamic solver will takeover 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 μ 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 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

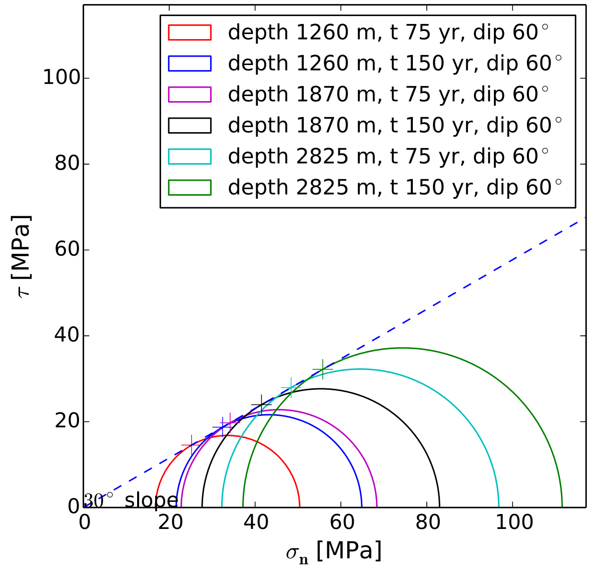


Fig. 10. Mohr-Coulomb circles for a 60° dip fault.

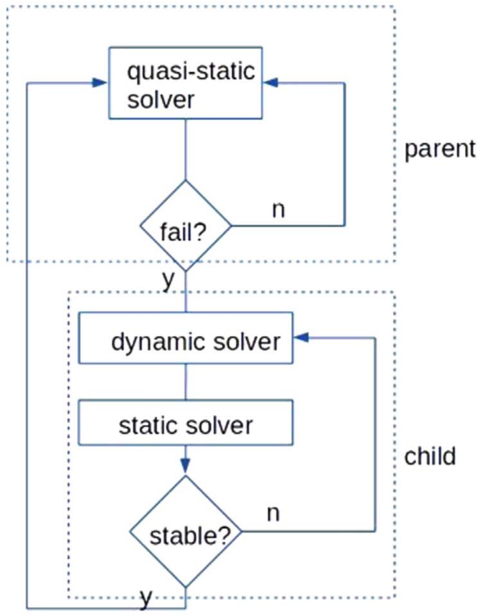


Fig. 11. Flow chart of the hybrid solver.

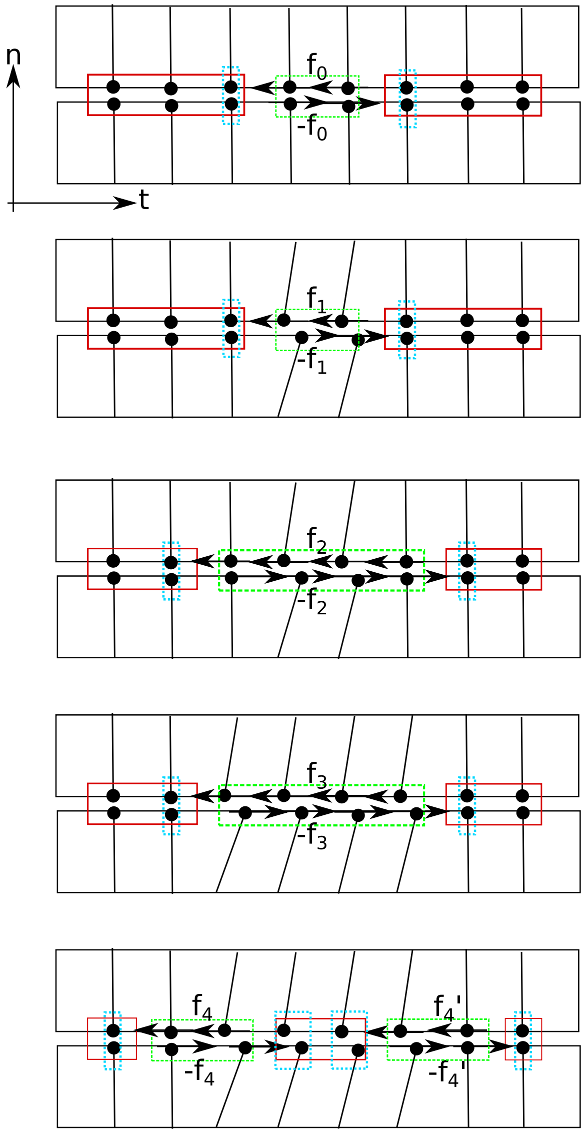


Fig. 12. Schematics of the rupture process: $f_{(\cdot)}$ denotes the net nodal force at different steps, the green frames enclose the slipping patches, the red frames enclose the locked patches, and the cyan frames enclose the rupture/arrest fronts. (For interpretation of the references to color in 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.

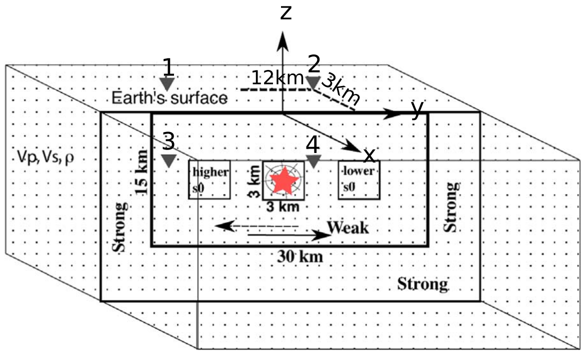


Fig. 13. SCEC TPV205 theme.

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 τ_{mid} overshooting the yielding stress, i.e. static friction $\mu_{\mathrm{sta}}\sigma_n$. The background shear stress τ_{ref} is between the dynamic friction $\mu_{\mathrm{dyn}}\sigma_n$ and static friction. A square patch to the left of the nucleation zone has shear stress τ_{left} higher than the background, and a square patch to right has shear stress τ_{right} lower than the background. The temporal friction coefficient follows the slip weakening law, Eq. (18),

$$\mu = \begin{cases} (\mu_{\mathrm{sta}} - \mu_{\mathrm{dyn}})(1 - D/D_c) + \mu_{\mathrm{dyn}}, & D \le D_c \\ \mu_{\mathrm{dyn}}, & D > D_c{}' \end{cases} \tag{18}$$

where, μ_{sta} and μ_{dyn} are the static and dynamic friction coefficients respectively, D is the slip distance, $\| \boldsymbol{u}^{(+)} - \boldsymbol{u}^{(+)} \|$, and $\boldsymbol{D_c}$ 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 2
SCEC benchmark TPV205 parameter list.

v_s	v_p	ρ	σ_n	τ_{ref}	τ_{left}	τ_{mid}	τ_{right}	D_c	μ_{sta}	μ_{dyn}
m/s	m/s	kg/m^3	MPa	MPa	MPa	MPa	MPa	m		
3464	6000	2670	120	70	78	81	62	0.4	0.667	0.525

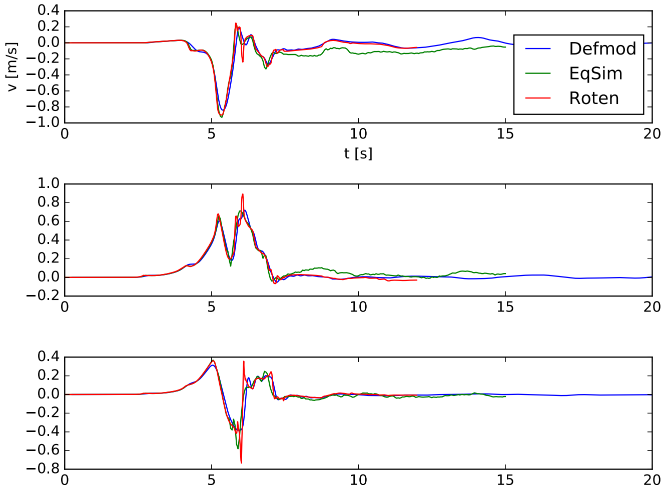


Fig. 14. SCEC TPV205 waveform comparison, station 1, velocity in x, y,z directions.

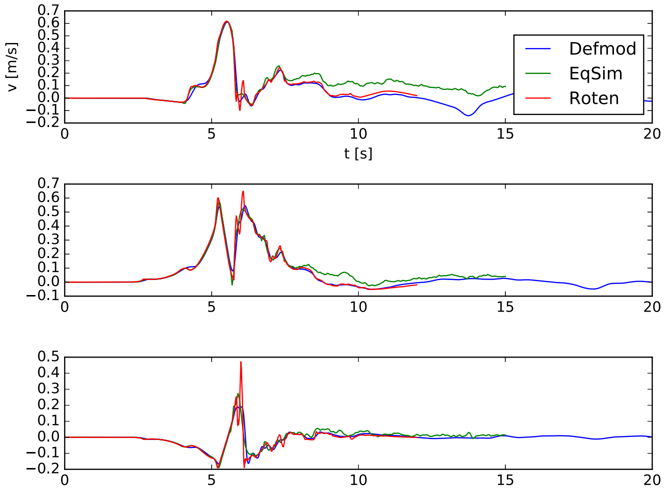


Fig. 15. SCEC TPV205 waveform comparison, station 2, velocity in x, y, z directions.

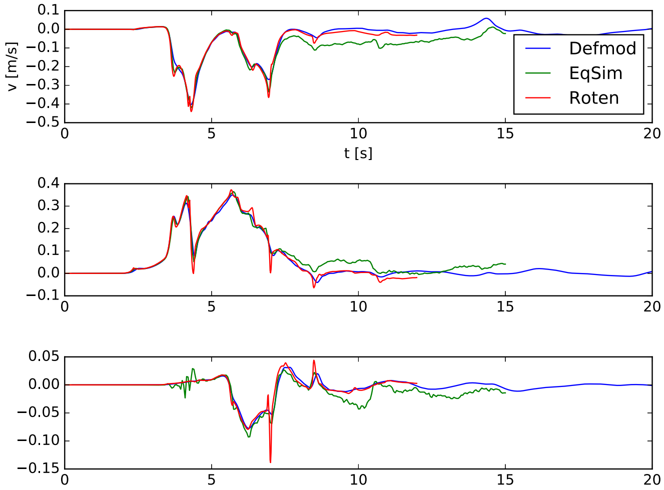


Fig. 16. SCEC TPV205 waveform comparison, station 3, velocity in x, y, z directions.

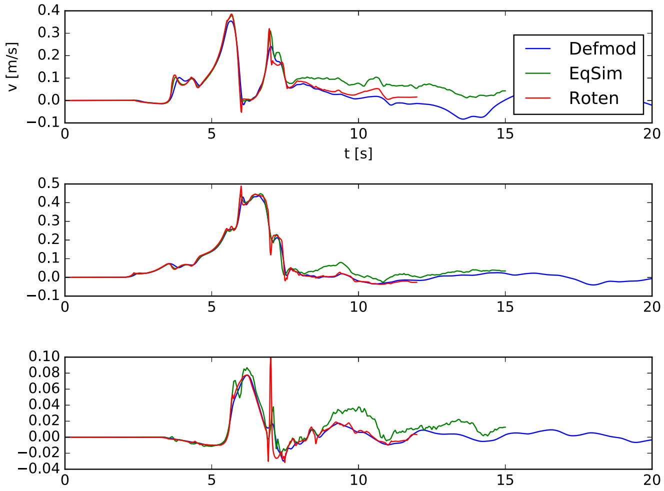


Fig. 17. SCEC TPV205 waveform comparison, station 4, velocity in x, y, z directions.

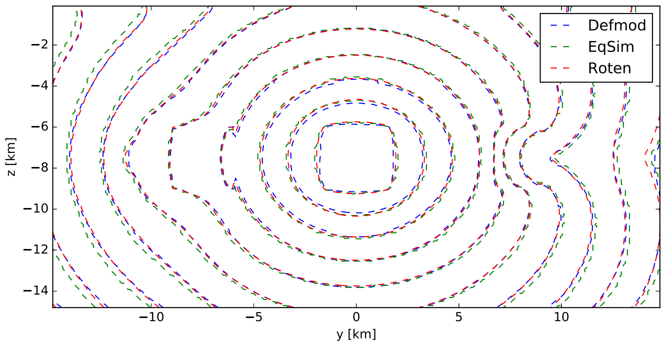


Fig. 18. SCEC TPV205 fault rupture front comparison.

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 $\mu_{\text{sta}} = 0.55$ 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.

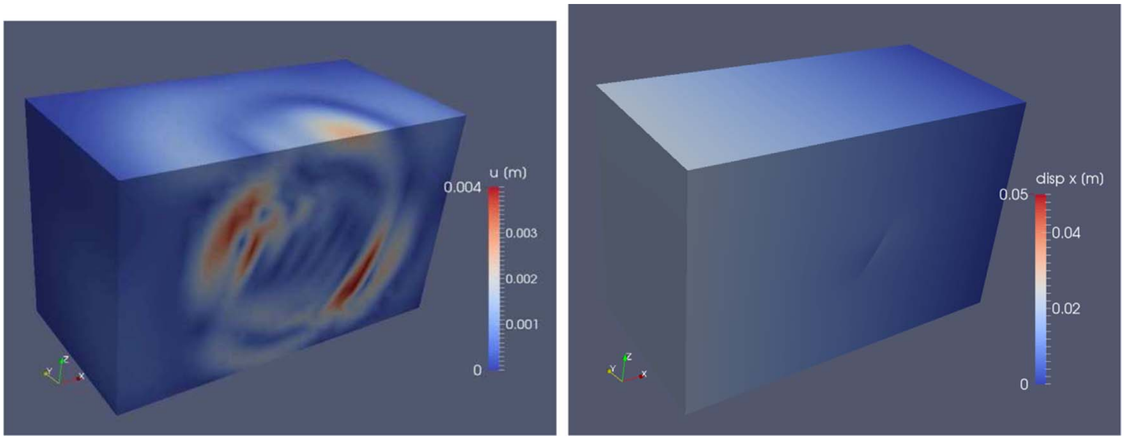


Fig. 19. Left: dynamic output showing displacement increment (delta function) magnitude within one time step due to the fault rupture. Right: quasi-static output showing the absolute x-displacement that is discontinuous due to the previous dynamic rupture.

Table 3
Summary of Defmod functionalities, numerical methods, benchmarks and references.

Functionality	Method	Benchmark	Reference
Poroelasticity	Implicit, pore pressure stabilization	Mandel	Smith and Griffiths (2004), Bochev and Dohrmann (2006), Kurashige et al. (2005)
Viscoelastic	Implicit	Abaqus	Melosh and Raefsky (1980)
power law			Ali (2014)
(quasi)static constraint	Lagrange Multiplier	Mohr-Coulomb	Zienkiewicz (2000)
Elastodynamic	Forward increment		Wong et al. (1989)
constraint	Lagrange Multiplier	SCEC	Carpenter et al. (1991)
absorbing	viscous damping		Lysmer and Kuhlemeyer (1969)
Isostasy	Winkler Foundation		Fry`ba (1995)
Fault/faulting	implicit/explicit	SCEC	Bartolomeo et al. (2010)

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:

$$\rho\frac{\partial^2\mathbf{u}}{\partial t^2}+\eta\frac{\partial\mathbf{u}}{\partial t}-\nabla\cdot\boldsymbol{\sigma}-\mathbf{f}=0,\quad \text{where,}\quad \boldsymbol{\sigma}=\mathbf{D}\boldsymbol{\epsilon},\quad \boldsymbol{\epsilon}=\nabla\mathbf{u}, \tag{A.1}$$

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

$$\begin{aligned}&\rho\ddot{\mathbf{u}}_{n+1-\alpha_\mathrm{m}}+\eta\dot{\mathbf{u}}_{n+1-\alpha_\mathrm{f}}-\nabla\cdot\boldsymbol{\sigma}_{n+1-\alpha_\mathrm{f}}-\mathbf{f}_{n+1-\alpha_\mathrm{f}}=0,\\ &\text{where, }\ddot{\mathbf{u}}_{n+1-\alpha_\mathrm{m}}=(1-\alpha_\mathrm{m})\ddot{\mathbf{u}}_{n+1}+\alpha_\mathrm{m}\ddot{\mathbf{u}}_n,\ \dot{\mathbf{u}}_{n+1-\alpha_\mathrm{f}}=(1-\alpha_\mathrm{f})\dot{\mathbf{u}}_{n+1}+\alpha_\mathrm{f}\dot{\mathbf{u}}_n,\ \nabla\cdot\boldsymbol{\sigma}_{n+1-\alpha_\mathrm{f}}=(1-\alpha_\mathrm{f})\nabla\cdot\boldsymbol{\sigma}_{n+1}+\alpha_\mathrm{f}\nabla\cdot\boldsymbol{\sigma}_n,\ \mathbf{f}_{n+1-\alpha_\mathrm{f}}=(1-\alpha_\mathrm{f})\mathbf{f}_{n+1}+\alpha_\mathrm{f}\mathbf{f}_n,\\ &\boldsymbol{\sigma}_k=\boldsymbol{\sigma}(\mathbf{u}_k),\end{aligned} \tag{A.2}$$

Newmark time stepping scheme for a choice of β and γ:

$$\dot{\mathbf{u}}_{n+1} = \frac{\gamma}{\beta\Delta t}(\mathbf{u}_{n+1} - \mathbf{u}_n) - \left(\frac{\gamma}{\beta} - 1\right)\dot{\mathbf{u}}_n - \Delta t\left(\frac{\gamma}{2\beta} - 1\right)\ddot{\mathbf{u}}_n\ddot{\mathbf{u}}_{n+1} = \frac{1}{\beta\Delta t^2}(\mathbf{u}_{n+1} - \mathbf{u}_n - \Delta t\dot{\mathbf{u}}_n) - \left(\frac{1}{2\beta} - 1\right)\ddot{\mathbf{u}}_n \tag{A.3}$$

After substitutions, the only variable that has subscript $n + 1$, i.e. needs to be sought for time t_{n+1}, is the displacement $\mathbf{u}_{n+1}$. 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 others to the right hand side:

$$\text{LHS} = c_{\text{m}}^{(1)}\mathbf{u}_{n+1} + c_{\text{d}}^{(1)}\mathbf{u}_{n+1} - (1 - \alpha_{\text{f}})\Delta\cdot\boldsymbol{\sigma}_{n+1}\text{RHS} = c_{\text{m}}^{(1)}\mathbf{u}_n + c_{\text{m}}^{(2)}\dot{\mathbf{u}}_n + c_{\text{m}}^{(3)}\ddot{\mathbf{u}}_n + c_{\text{d}}^{(1)}\mathbf{u}_n \quad + c_{\text{d}}^{(2)}\dot{\mathbf{u}}_n + c_{\text{d}}^{(3)}\ddot{\mathbf{u}}_n + \alpha_{\text{f}}\Delta\cdot\boldsymbol{\sigma}_n \quad + (1 - \alpha_{\text{f}})\mathbf{f}_{n+1} + \alpha_{\text{f}}\mathbf{f}_n, \text{ where,}$$
$$c_{\text{m}}^{(1)} = \frac{\rho(1 - \alpha_{\text{m}})}{\beta\Delta t^2},\ c_{\text{m}}^{(2)} = \frac{\rho(1 - \alpha_{\text{m}})}{\beta\Delta t},\ c_{\text{m}}^{(3)} = \frac{\rho(1 - \alpha_{\text{m}} - 2\beta)}{2\beta},\ c_{\text{d}}^{(1)} = \frac{\eta\gamma(1 - \alpha_{\text{f}})}{\beta\Delta t},\ c_{d}^{(2)} = \frac{\eta((1 - \alpha_{\text{f}})\gamma + \beta)}{\beta},\ c_{d}^{(3)} = \frac{\eta\Delta t(1 - \alpha_{\text{f}})\gamma(\gamma - 2\beta)}{2\beta} \tag{A.4}$$

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 $\mathbf{r}(\mathbf{x})$, and take volume integrals over the model domain.

$$(\mathbf{r}, \text{LHS})_{\Omega} = (\mathbf{r}, \text{RHS})_{\Omega}, \quad \forall\ \mathbf{r} \in L^2(H^1(\Omega)), \tag{A.5}$$

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

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

$$(\mathbf{r}, (\nabla\cdot\boldsymbol{\sigma}_k))_{\Omega} = -(\nabla\mathbf{r}, \boldsymbol{\sigma}_k)_{\Omega} + (\mathbf{r}, (\boldsymbol{\sigma}_k\cdot\mathbf{n}))_{\partial\Omega}, \tag{A.6}$$

where, $\mathbf{n}$ is normal vector on the domain boundary denoted by $\partial\Omega$.

For given boundary traction $\boldsymbol{\tau}_k = \boldsymbol{\sigma}_k\cdot\mathbf{n}$, the governing equation is written as:

$$\mathbf{a} = \mathbf{L}, \text{ where}\mathbf{a} = (\mathbf{r}, c_{\text{m}}^{(1)}\mathbf{u}_{n+1})_{\Omega} + (\mathbf{r}, c_{\text{d}}^{(1)}\mathbf{u}_{n+1})_{\Omega} + (1 - \alpha_{\text{f}})(\nabla\mathbf{r}, \boldsymbol{\sigma}_{n+1})_{\Omega},\ \mathbf{L} = (\mathbf{r}, c_{\text{m}}^{(1)}\mathbf{u}_n)_{\Omega} + (\mathbf{r}, c_{\text{m}}^{(2)}\dot{\mathbf{u}}_n)_{\Omega} + (\mathbf{r}, c_{\text{m}}^{(3)}\ddot{\mathbf{u}}_n)_{\Omega} + (\mathbf{r}, c_{\text{d}}^{(1)}\mathbf{u}_n)_{\Omega} + (\mathbf{r}, c_{\text{d}}^{(2)}\dot{\mathbf{u}}_n)_{\Omega}$$
$$+ (\mathbf{r}, c_{\text{d}}^{(3)}\ddot{\mathbf{u}}_n)_{\Omega} - \alpha_{\text{f}}(\nabla\mathbf{r}, \boldsymbol{\sigma}_n)_{\Omega} + (\mathbf{r}, (1 - \alpha_{\text{f}})\boldsymbol{\tau}_{n+1} + \alpha_{\text{f}}\boldsymbol{\tau}_n))_{\partial\Omega} + (\mathbf{r}, (1 - \alpha_{\text{f}})\mathbf{f}_{n+1} + \alpha_{\text{f}}\mathbf{f}_n)_{\Omega}. \tag{A.7}$$

This equation can be solved for given initial conditions $(\mathbf{u}_0,\ \dot{\mathbf{u}}_0,\ \ddot{\mathbf{u}}_0)$, boundary conditions $\boldsymbol{\tau}_n$ and body forces $\mathbf{f}_n$. Usually the body force is a known constant, e.g. $\mathbf{f}_{n+1} = \mathbf{f}_n = \rho\mathbf{g}$. A second order time scheme, is an implementation of the generalized-α method with specific β, γ, α_{m} and α_{f}. A first oder time scheme is obtained by neglecting the last terms in Eq. (A.3).

The test function, $\mathbf{r}(\mathbf{x})$, within an element e is given by the products of the shape function, $\mathbf{N}^{(e)}(\mathbf{x})$, $([N_1N_2\ldots])$, and the nodal values, $\mathbf{r}^{(e)}$ $([\mathbf{r}_1\mathbf{r}_2\ldots]^{\text{T}})$,

$$\mathbf{r}(\mathbf{x}) = \mathbf{N}^{(e)}(\mathbf{x})\mathbf{r}^{(e)}. \tag{A.8}$$

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

$$\nabla\mathbf{r}(\mathbf{x}) = \mathbf{B}^{(e)}(\mathbf{x})\mathbf{r}^{(e)}. \tag{A.9}$$

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

$$\mathbf{B}^{(e)} = \mathbf{A}\mathbf{S}^{(e)}. \tag{A.10}$$

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

$$\mathbf{A} = \begin{bmatrix} \partial/\partial x & 0 & 0 \\ 0 & \partial/\partial y & 0 \\ 0 & 0 & \partial/\partial z \\ \partial/\partial y & \partial/\partial x & 0 \\ 0 & \partial/\partial z & \partial/\partial y \\ \partial/\partial z & 0 & \partial/\partial x \end{bmatrix},\ \mathbf{S}^{(e)} = \begin{bmatrix} N_1 & 0 & 0 & N_2 & 0 & 0 & \ldots \\ 0 & N_1 & 0 & 0 & N_2 & 0 & \ldots \\ 0 & 0 & N_1 & 0 & 0 & N_2 & \ldots \end{bmatrix}. \tag{A.11}$$

The width of $\mathbf{S}^{(e)}$ is the element node count multiplied by dimension, 3. The shape function and its spatial derivatives differ for different choice of element. If $\mathbf{B}^{(e)}$ has the layout determined by Eq. (A.10), (A.11), the $\mathbf{r}^{(e)}$ should be *flattened*, i.e. $[\mathbf{r}_1^{\text{T}}\mathbf{r}_2^{\text{T}}\ldots]^{\text{T}}$, to keep the product in Eq. (A.9), conforming.

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

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

$$(\mathbf{r}, (\cdot))_e = \mathbf{r}^{(e)\text{T}}\int_{\Omega_e}\mathbf{N}^{(e)\text{T}}(\cdot)d\Omega_e,\ (\nabla\mathbf{r}, (\cdot))_e = \mathbf{r}^{(e)\text{T}}\int_{\Omega_e}\mathbf{B}^{(e)\text{T}}(\cdot)d\Omega_e, \tag{A.12}$$

The equation Eq. (A.8) also applies to the trial functions, i.e. $\mathbf{u}(\mathbf{x})$ and its time derivatives, by substituting $\mathbf{r}^{(e)}$ with $\mathbf{u}^{(e)}$ 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

$$(\mathbf{r}, c_m^{(\cdot)}\ddot{\mathbf{u}})_e = \mathbf{r}^{(e)\text{T}}\int_{\Omega_e} c_m^{(\cdot)}\mathbf{N}^{(e)\text{T}}\mathbf{N}^{(e)}\ddot{\mathbf{u}}^{(e)}d\Omega_e, \tag{A.13}$$

where $c_m^{(\cdot)}$ is one of the coefficients listed in Eq. (A.7) that contain ρ. Eq. (A.12) suggests that, a left multiplication term $\mathbf{r}^{(e)\text{T}}$ would be resulted by all the integrals on both LHS and RHS of Eq. (A.7). Due to the arbitrariness of $\mathbf{r}$, the equation should hold before the left multiplication. What needs to be integrated by element is

$$\int_{\Omega_e} \rho \mathbf{N}^{(e)\mathrm{T}} \mathbf{N}^{(e)} d\Omega_e = \mathbf{M}^{(e)}. \tag{A.14}$$

When $\ddot{\mathbf{U}}$ is available, we use the vector $\mathbf{M}\ddot{\mathbf{U}}$ 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.

$$\boldsymbol{\epsilon}(\mathbf{x}) = \mathbf{B}^{(e)}(\mathbf{x})\mathbf{u}^{(e)}. \tag{A.15}$$

The formulation of $\mathbf{B}^{(e)}$ insures a symmetric tensor $\boldsymbol{\epsilon}$, known as symmetric gradient. Substituting the strain in Eq. (A.1) by Eq. (A.15), the stress within the element e is then

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{D}^{(e)}\mathbf{B}^{(e)}(\mathbf{x})\mathbf{u}^{(e)}. \tag{A.16}$$

The second row of Eq. (A.12), after substituting $(\cdot)$ with Eq. (A.16), becomes

$$(\nabla \mathbf{r}, \boldsymbol{\sigma})_{\Omega_e} = \mathbf{r}^{(e)\mathrm{T}} \int_{\Omega_e} \mathbf{B}^{(e)\mathrm{T}} \mathbf{D}^{(e)} \mathbf{B}^{(e)} \mathbf{u}^{(e)} d\Omega_e. \tag{A.17}$$

Again, because of the arbitrariness of the test function, what needs to be integrated is

$$\int_{\Omega_e} \mathbf{B}^{(e)\mathrm{T}} \mathbf{D}^{(e)} \mathbf{B}^{(e)} d\Omega_e = \mathbf{K}^{(e)}. \tag{A.18}$$

When $\mathbf{U}$ is available, we use the vector $\mathbf{KU}$ 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

$$\int_{\Omega_e} \mathbf{N}^{(e)\mathrm{T}} \boldsymbol{\eta} \mathbf{N}^{(e)} d\Omega_e = \mathbf{C}^{(e)}, \tag{A.19}$$

The damping parameter $\boldsymbol{\eta}$ can be scaler or tensor (e.g. absorbing boundary) valued. In Defmod, instead of doing this integration, we use the relation $\mathbf{C} = \alpha_\eta \mathbf{M} + \beta_\eta \mathbf{K}$ to form the global damping matrix. The scaler parameters α_η and β_η, 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 $\gamma = 1$, $\beta = 1$, $\alpha_{\mathbf{m}} = 0$ and $\alpha_{\mathrm{f}} = 1$. This leads to, after writing Eq. (A.7) in the global integral form,

$$\mathbf{M}\ddot{\mathbf{U}}_{n+1} = -\mathbf{C}\dot{\mathbf{U}}_n - \mathbf{K}\mathbf{U}_n + \int_{\partial\Omega} \mathbf{N}^{\mathrm{T}} \boldsymbol{\tau}_n d\partial\Omega + \int_{\Omega} \mathbf{N}^{\mathrm{T}} \mathbf{f} d\Omega,\ \dot{\mathbf{U}}_n = (\mathbf{U}_n - \mathbf{U}_{n-1})/\Delta t,\ \ddot{\mathbf{U}}_{n+1} = (\mathbf{U}_{n+1} - 2\mathbf{U}_n + \mathbf{U}_{n-1})/\Delta t^2, \tag{A.20}$$

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

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

$$\Delta t \le L/\sqrt{E/\rho},\ f \le 0.1\sqrt{\mu/\rho}/L, \tag{A.21}$$

where L is the representative length of the elements, E and μ 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 Δt. 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 $\mathbf{f}_n$, 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),

$$\mathbf{K}\mathbf{U}_n = \mathbf{f}_n, \tag{A.22}$$

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

The slightly compressible pore fluid pressure p has the governing equation:

$$c_t \phi \dot{p} - \nabla\cdot k \nabla p - \boldsymbol{\alpha}_{\mathrm{B}}^{\mathrm{T}} \dot{\boldsymbol{\epsilon}} = q, \tag{A.23}$$

where $\boldsymbol{\alpha}_{\mathrm{B}} = \alpha_{\mathrm{B}}[1\ \ 1\ \ 1\ \ 0\ \ 0\ \ 0]^{\mathrm{T}}$, $\boldsymbol{\epsilon} = [\epsilon_{xx}\ \ \epsilon_{yy}\ \ \epsilon_{zz}\ \ \epsilon_{xy}\ \ \epsilon_{yz}\ \ \epsilon_{xz}]^{\mathrm{T}}$ in 3D, $\alpha_{\mathrm{B}} \in [0,\ \ 1]$ being the Biot's coefficient, c_t 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 scaler valued test function r, $\forall\ r \in L^2(H^1(\Omega))$, and integrate over Ω. The term with pressure gradient, after integrated by part, becomes

$$(r, \nabla\cdot k\nabla p)_{\Omega} = -(\nabla r, k\nabla p)_{\Omega} + (r, \mathbf{n}\cdot k\nabla \mathbf{p})_{\partial\Omega}. \tag{A.24}$$

We use f to denote $\mathbf{n}\cdot k\nabla\mathbf{p}$, the fluid flux in the boundary normal direction $\mathbf{n}$, The weak form governing equation becomes

$$\mathbf{a} = \mathbf{L}\text{where, } \mathbf{a} = (r,\ c_t\phi\dot{p})_\Omega + (\nabla r,\ k\nabla p)_\Omega - (r,\ \boldsymbol{\alpha}_\mathrm{B}^\mathrm{T}\dot{\boldsymbol{\epsilon}})_\Omega \mathbf{L} = (r,\ q)_\Omega + (r,\ f)_{\partial\Omega}, \tag{A.25}$$

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

$$\nabla r\,(\mathbf{x}) = \mathbf{dN}^{(e)}(\mathbf{x})\mathbf{r}^{(e)}, \tag{A.26}$$

where in 3D,

$$\mathbf{dN}^{(e)} = \begin{bmatrix} \partial N_1/\partial x & \partial N_2/\partial x & \dots \\ \partial N_1/\partial y & \partial N_2/\partial y & \dots \\ \partial N_1/\partial z & \partial N_2/\partial z & \dots \end{bmatrix}. \tag{A.27}$$

The width of $\mathbf{dN}^{(e)}$ is the node count of the element.

Similar to Eq. (A.12), the element integrations in Eq. (A.25) are of the forms,

$$(r,\ (\cdot))_e = \mathbf{r}^{(e)\mathrm{T}} \int_{\Omega_e} \mathbf{N}^{(e)\mathrm{T}}(\cdot)d\Omega_e,\ (\nabla r,\ (\cdot))_e = \mathbf{r}^{(e)\mathrm{T}} \int_{\Omega_e} \mathbf{dN}^{(e)\mathrm{T}}(\cdot)d\Omega_e, \tag{A.28}$$

The time derivative $\dot{p}\,(\mathbf{x})$ can be written in the form of Eq. (A.8), by substituting $\mathbf{r}(\mathbf{x})$ and $\mathbf{r}^{(e)}$ with $\dot{p}\,(\mathbf{x})$ and $\dot{\mathbf{p}}^{(e)}$. The gradient $\nabla p\,(\mathbf{x})$ can be written in the form of Eq. (A.26) by substituting $r(x)$ and $\mathbf{r}^{(e)}$ with $p(x)$ and $\mathbf{p}^{(e)}$. The strain rate $\dot{\boldsymbol{\epsilon}}$ is given by Eq. (A.15), by substituting $\mathbf{u}$ with $\dot{\mathbf{u}}$. Very integral in Eq. (A.25), shown by Eq. (A.28), would result a left multiplication term $\mathbf{r}^{(e)\mathrm{T}}$. Again, because of the arbitrariness of the test function, the equation should hold before the left multiplication. What need to be integrated are

$$\int_{\Omega_e} \mathbf{dN}^{(e)\mathrm{T}}\mathbf{k}(\mathbf{dN}^{(e)})d\Omega_e = \mathbf{K}_c^{(e)},\ \int_{\Omega_e} \mathbf{N}^{(e)\mathrm{T}}\boldsymbol{\alpha}_\mathrm{B}^{\mathrm{T}(e)}\mathbf{B}^{(e)}d\Omega_e = \mathbf{H}^{\mathrm{T}(e)},\ \int_{\Omega_e} \mathbf{N}^{(e)\mathrm{T}}c_t\mathbf{N}^{(e)}d\Omega_e = \mathbf{S}_p^{(e)}, \tag{A.29}$$

where, $\boldsymbol{\alpha}_\mathrm{B}^{(e)}$ is formed by stacking $\boldsymbol{\alpha}_\mathrm{B}$ (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

$$c_t = (1-\alpha_\mathrm{B})(\alpha_\mathrm{B}-\phi)/k_s + \phi/k_f, \tag{A.30}$$

where, k_s and k_f are bulk moduli for the solid and fluid respectively.

The element integral equation becomes

$$-\mathbf{H}^{\mathrm{T}(e)}\dot{\mathbf{u}}_n^{(e)} + \mathbf{S}_p^{(e)}\dot{\mathbf{p}}_n^{(e)} + \mathbf{K}_c^{(e)}\mathbf{p}_n^{(e)} = (\mathbf{N}^{\mathrm{T}(e)},\ q_n)_{\Omega_e} + (\mathbf{N}^{\mathrm{T}(e)},\ f_n)_{\partial\Omega_e}. \tag{A.31}$$

Applying the first order time scheme, i.e. $\dot{\mathbf{p}}_n = (\mathbf{p}_n - \mathbf{p}_{n-1})/\Delta t$, and $\dot{\mathbf{u}}_n = (\mathbf{u}_n - \mathbf{u}_{n-1})/\Delta t$, Eq. (A.31) becomes

$$-\mathbf{H}^{\mathrm{T}(e)}(\mathbf{u}_n^{(e)} - \mathbf{u}_{n-1}^{(e)})/\Delta t + \mathbf{S}_p^{(e)}(\mathbf{p}_n^{(e)} - \mathbf{p}_{n-1}^{(e)})/\Delta t + \mathbf{K}_c^{(e)}\mathbf{p}_n^{(e)} = (\mathbf{N}^{\mathrm{T}(e)},\ q_n)_{\Omega_e} + (\mathbf{N}^{\mathrm{T}(e)},\ f_n)_{\partial\Omega_e}. \tag{A.32}$$

Note that, a higher order time scheme for p is impossible for Eq. (A.23), becomes the second order time derivative $\ddot{p}$ has been already neglected. Again, we combine the source/flux terms as a global vector, denoted by $\mathbf{q}_n$ 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 $\Delta\mathbf{u}_n = \mathbf{u}_n - \mathbf{u}_{n-1}$, and $\Delta\mathbf{p}_n = \mathbf{p}_n - \mathbf{p}_{n-1}$, is given as:

$$-\mathbf{H}^\mathrm{T}\Delta\mathbf{u}_n + (\mathbf{S}_p + \Delta t\mathbf{K}_c)\Delta\mathbf{p}_n = -\Delta t\mathbf{K}_c\mathbf{p}_{n-1} + \mathbf{q}_n. \tag{A.33}$$

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,

$$\boldsymbol{\sigma}_\mathrm{total} = \boldsymbol{\sigma} - \boldsymbol{\alpha}_\mathrm{B}p, \tag{A.34}$$

where, the Biot's coefficient should be tensor shaped, $\boldsymbol{\alpha}_\mathrm{B} = \alpha_\mathrm{B}I_{3\times3}$ in 3D, to make the product conforming. Shown later however, the Biot's coefficients will all appear as $\boldsymbol{\alpha}_\mathrm{B}^{(e)}$, 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 $-\nabla\cdot\boldsymbol{\alpha}_\mathrm{B}p$. Multiplied by the test function $\mathbf{r}$, and integrated by part, this term becomes

$$-(\mathbf{r},\ \nabla\cdot\boldsymbol{\alpha}_\mathrm{B}p)_\Omega = (\nabla\mathbf{r},\ \boldsymbol{\alpha}_\mathrm{B}p)_\Omega - (\mathbf{r},\ (\boldsymbol{\alpha}_\mathrm{B}p,\ \mathbf{n}))_{\partial\Omega}. \tag{A.35}$$

The second term in Eq. (A.35) is taken care of by the pressure boundary condition, a part of $\mathbf{f}_n$ in Eq. (A.22), instead of doing the integration. $\nabla\mathbf{r}$ is given by Eq. (A.9), and p is given by Eq. (A.8), after substituting $\mathbf{r}(\mathbf{x})$ and $\mathbf{r}^{(e)}$ with $p\,(\mathbf{x})$ and $\mathbf{p}^{(e)}$. The first term in Eq. (A.35), element-wise, becomes

$$(\nabla\mathbf{r},\ \boldsymbol{\alpha}_\mathrm{B}p)_{\Omega_e} = \mathbf{r}^{(e)\mathrm{T}} \int_{\Omega_e} \mathbf{B}^{(e)\mathrm{T}}\boldsymbol{\alpha}_\mathrm{B}^{(e)}\mathbf{N}^{(e)}d\Omega_e\mathbf{p}^{(e)}. \tag{A.36}$$

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

$$\mathbf{K}_e\mathbf{u}_n + \mathbf{H}\mathbf{p}_n = \mathbf{f}_n, \tag{A.37}$$

where, $\mathbf{K}_e$ denotes the elastic stiffness matrix. After substituting $\mathbf{u}$, $\mathbf{p}$ and $\mathbf{f}$ with $\Delta\mathbf{u}$, $\Delta\mathbf{p}$ and $\Delta\mathbf{f}$, Eq. (A.37) becomes a incremental equation,

$$\mathbf{K}_e\Delta\mathbf{u}_n + \mathbf{H}\Delta\mathbf{p}_n = \Delta\mathbf{f}_n. \tag{A.38}$$

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).

$$q\,(\mathbf{x})_{n+1} = (\mathbf{N}^{(e)}(\mathbf{x}) - (1/n_e)I_{\mathbf{N}})(\Delta\mathbf{p}_{n+1}^{(e)} + \mathbf{p}_n^{(e)})/(2G), \tag{A.39}$$

where, $\mathbf{N}^{(e)}(\mathbf{x}) - (1/n_e)I_{\mathbf{N}}$ 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

```
(pathtodefmod)/defmod − ffoo. inp[PETScoptions]
```

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

```
−pctypelu − pcfactormatsolverpackagemumps
```

The python code `Mandel3D_preproc.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 `Fault3d_hyb` and `Fault2d_hyb` 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/, and mesh (`.exo`) files in Exodus II format. To generate the input files `3D_flt_cur.inp` and `2D_flt_cur.inp`, run

```
./ Fault3dhex. py3Dfltcur. exo
```

and

```
./ Fault2dqua. py2Dfltcur. 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.

```
…fault = Trueporo = Falsevisc = False…t = 1500000. ;dt = 100000. ;nviz = 1…fc = .6*np. ones((len(vecf), 1))…
```

Run the preprocessing code

```
./ Fault3Dhex. py3Dfltcur. 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. py3Dfltcur
```

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 wave form 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 constraint 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).

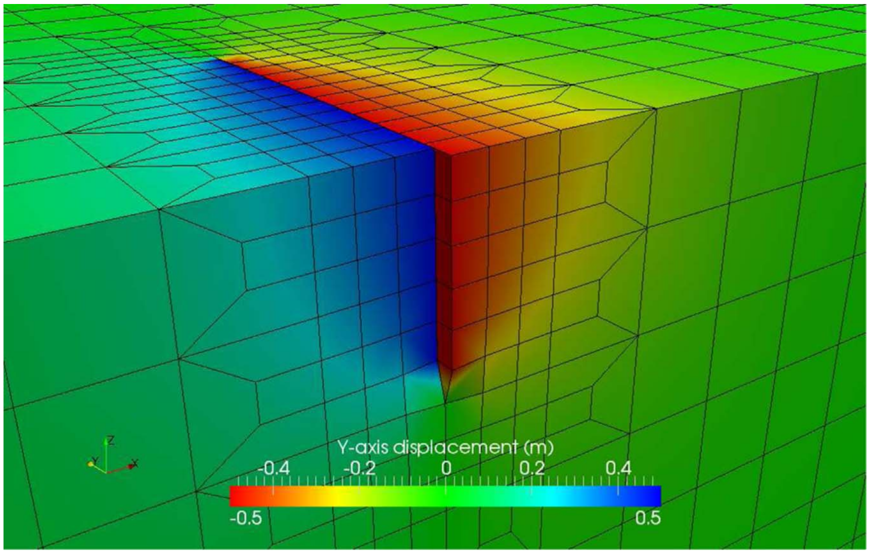


Fig. C.1. Part of the finite element mesh used for validation of quasistatic viscoelastic relaxation. Colors represent fault parallel displacements, at t=0 years, provided by Ali (2014).

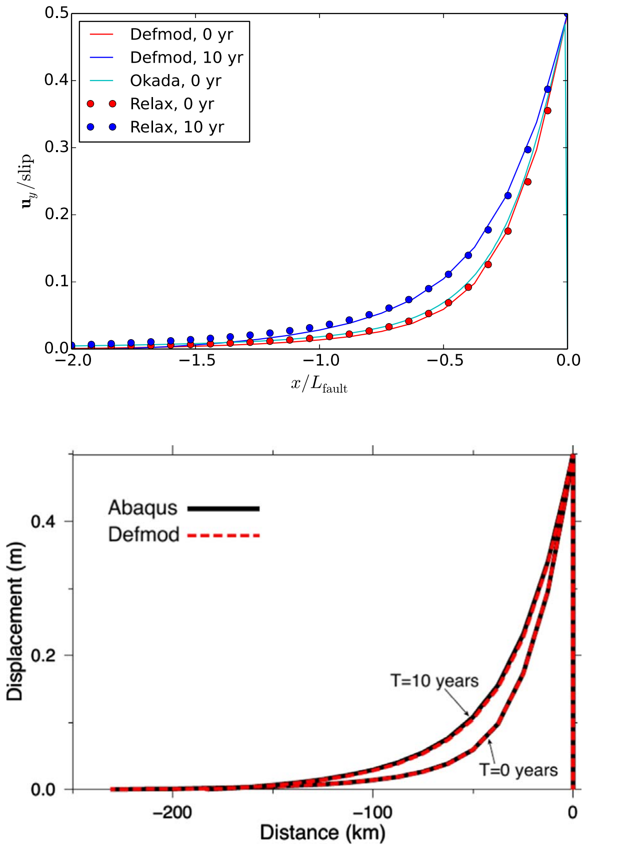


Fig. C.2. Deformation in the strike direction at the surface, along the rout perpendicular to the fault, upper: comparison between Defmod, Okada and Relax; lower: comparison between Defmod and Abaqus provided by Ali (2014).

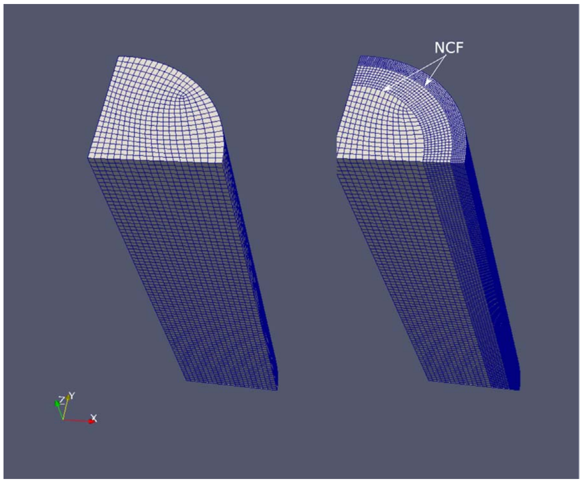


Fig. C.3. Mesh with two non-conformal interfaces used for duplicating the cylindrical Mandel solution.

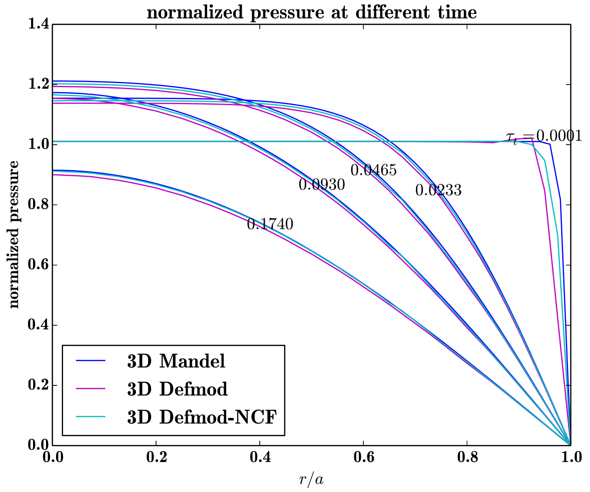


Fig. C.4. Pressure along the radius of the bottom by conformal and non-conformal

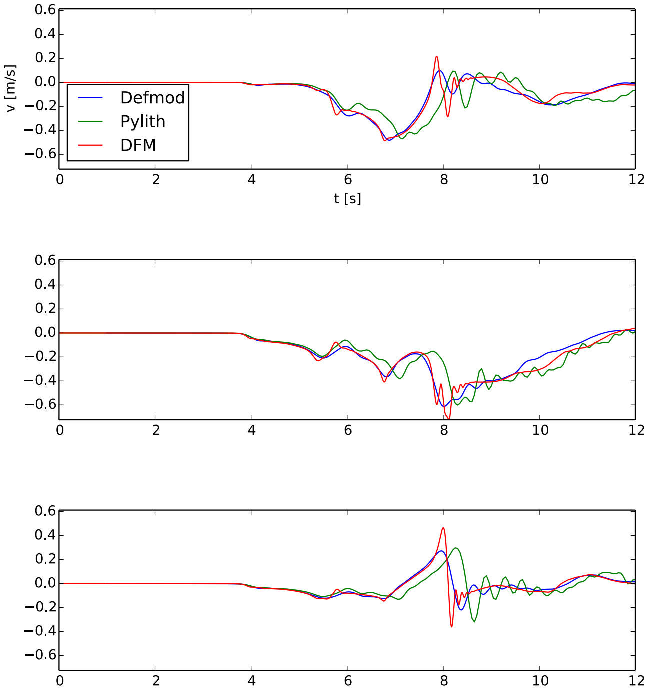


Fig. C.5. SCEC TPV102 waveform comparison, station 1, velocity in x, y, z directions.

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.

Fig. C.5 compares the waveforms recorded at the station 2, 9 km off the fault plane, aligned with the epicenter along the normal (x) 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.

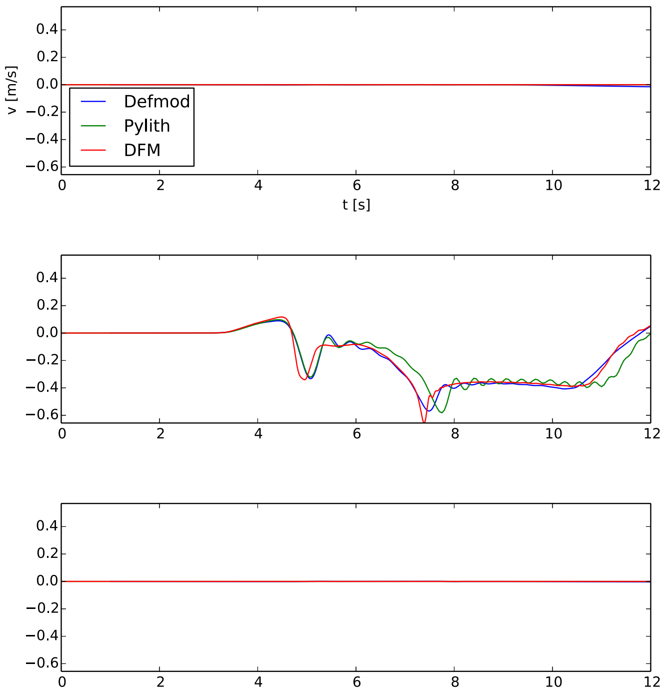


Fig. C.6. SCEC TPV102 waveform comparison, station 2, velocity in x, y, z directions.

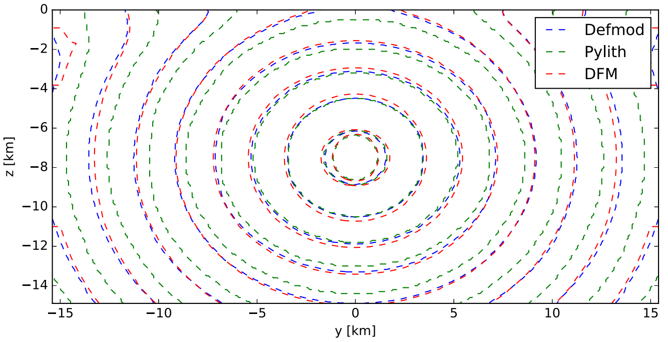


Fig. C.7. SCEC TPV102 fault rupture front comparison.

References

Aagaard, B., Kientz, S., Knepley, M., Strand, L., Williams, C., 2015. Pylith – an finite-element code for dynamic and quasistatic simulations of crustal deformation, primarily earthquakes and volcanoes., ⟨https://github.com/geodynamics/pylith⟩.

Ali, S.T., 2015. Defmod – Finite element code for modeling crustal deformation, Bitbucket, Bitbucket repository ⟨https://bitbucket.org/stali/defmod⟩.

Ali, S.T., 2014. Defmod - Parallel Multiphysics Finite Element Code for Modeling Crustal Deformation during the Earthquake/rifting Cycle, arXiv:1402.0429 ⟨http://arxiv.org/abs/1402.0429⟩.

Ampuero, J.P., Rubin, A.M., 2008. Earthquake nucleation on rate and state faults – aging and slip laws. J. Geophys. Res.: Solid Earth 113, B01302, (n/a-n/a).

Balay, S., Abhyankar, S., Adams, M.F., Brown, J., Brune, P., Buschelman, K., Dalcin, L., Eijkhout, V., Gropp, W.D., Kaushik, D., Knepley, M.G., McInnes, L.C., Rupp, K., Smith, B.F., Zampini, S., Zhang, H., 2015. PETSc Web page, ⟨http://www.mcs.anl.gov/petsc⟩.

Barbot, S., Fialko, Y., 2010. A unified continuum representation of post-seismic relaxation mechanisms: semi-analytic models of afterslip, poroelastic rebound and viscoelastic flow. Geophys. J. Int. 182, 1124–1140.

Bartolomeo, M.D., Meziane, A., Massi, F., Baillet, L., Fregolent, A., 2010. Dynamic rupture at a frictional interface between dissimilar materials with asperities. Tribol. Int. 43, 1620–1630.

Bochev, P., Dohrmann, C., 2006. A computational study of stabilized, low-order c 0 finite element approximations of darcy equations. Comput. Mech. 38, 323–333.

Carpenter, N.J., Taylor, R.L., Katona, M.G., 1991. Lagrange constraints for transient finite element surface contact. Int. J. Numer. Methods Eng. 32, 103–128.

Cui, Y., Olsen, K.B., Jordan, T.H., Lee, K., Zhou, J., Small, P., Roten, D., Ely, G., Panda, D.K., Chourasia, A., Levesque, J., Day, S.M., Maechling, P., 2010. Scalable earthquake simulation on petascale supercomputers. In: Proceedings of the 2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, SC '10, IEEE Computer Society, Washington, DC, USA, pp. 1–20.

Dalguer, L.A., Day, S.M., 2006. Comparison of fault representation methods in finite difference simulations of dynamic rupture. Bull. Seismol. Soc. Am. 96, 1764–1778.

Dalguer, L.A., Day, S.M., 2007. Staggered-grid split-node method for spontaneous rupture simulation. J. Geophys. Res.: Solid Earth 112, B02302.

Fry`ba, L., 1995. History of winkler foundation. Veh. Syst. Dyn. 24, 7–12.

Harris, R. Barall, M. Archuleta, R.J. Aagaard, B. Ampuero, J.P. Andrews, D.J., Cruz-Atienza, V.M. Dalguer Gudiel, L.A., Day, S.M., Duan, B., Dunham, E.M., Ely, G.P., Gabriel, A.A., Kaneko, Y., Kase, Y., Lapusta, N., Ma, S., Noda, H., Oglesby, D.D., Olsen, K.B., Roten, D., Song, S., 2010. The SCEC-USGS Dynamic Earthquake Rupture Code Verification Exercise: Regular and Extreme Ground Motion, AGU Fall Meeting Abstracts.

Henderson, A., 2007. ParaView Guide, A Parallel Visualization Application. Kitware Inc., ⟨http://www.paraview.org/⟩.

Jha, B., Juanes, R., 2014. Coupled multiphase flow and poromechanics: a computational model of pore pressure effects on fault slip and earthquake triggering. Water Resour.

Res. 50, 3776–3808.

Kurashige, M., Sato, K., Imai, K., 2005. Mandel and cryer problems of fluid-saturated foams with negative poisson's ratio. Acta Mech. 175, 25–43.

Lysmer, J., Kuhlemeyer, R.L., 1969. Finite dynamic model for infinite media. J. Eng. Mech. – ASCE 859, 877.

Melosh, H.J., Raefsky, A., 1980. The dynamical origin of subduction zone topography. Geophys. J. R. Astron. Soc. 60, 333–354.

Meng, C., 2015. A Developer Version of Defmod Focusing on Fault/faulting Modeling. ⟨https://bitbucket.org/chunfangmeng/defmod-dev⟩.

Okada, Y., 1985. Surface deformation due to shear and tensile faults in a half-space. Bull. Seismol. Soc. Am. 75, 1135–1154.

Sandia National Lab, 2012. Cubit–Geometry and Mesh Generation Toolkit. ⟨http://cubit.sandia.gov/⟩

Savage, J.C., Prescott, W.H., 1978. Asthenosphere readjustment and the earthquake cycle. J. Geophys. Res.: Solid Earth 83, 3369–3376.

Simulia, 2008. Dassault Systemes, Abaqus User's Manual, Version 6.8.

Smith, I., Griffiths, D., 2004. Programming the Finite Element Method. John Wiley & Sons.

Timoshenko, S., Young, D., Weaver, W., 1974. Vibration Problems in Engineering. Wiley.

Wong, S.W., Smith, I.M., Gladwell, I., 1989. Pcg methods in transient fe analysis. Part ii: second order problems. Int. J. Numer. Methods Eng. 28, 1567–1576.

Zienkiewicz, T.R., O.C., 2000. Finite Element Method (5th Edition) Volume 1 – The Basis, Elsevier.