Appendix: Computational Method

The finite element method is a well-known computation technique used to solve problems in many areas, including fluid flow, electrostatics/dynamics, and solid mechanics/dynamics. In the case of an explicit dynamic finite element code such as Dyna2D (Whirley et al., 1992) and Dyna3D (Whirley and Engelmann, 1993), the general purpose of the code is to calculate how a body will respond to applied forces. The general process is first to discretize the medium and assign material properties to every element in it. Then, under the desired applied forces, the program simultaneously solves Newton's second law and some sort of material constituitive law (Hooke's law in the case of a linearly elastic solid) for every node (element corner) in the medium. This process provides as a result the acceleration of each node. This acceleration can then be numerically integrated to produce velocity and displacement. The resulting displacements cause the elastic forces on the elements to be modified, and the process is repeated for the next time step. In this manner elastic waves (such as seismic waves in the Earth) can be propagated.

The finite element method has been explained in detail in numerous sources (e.g., Zienkiewicz and Cheung, 1967), and the current work makes use of the default finite element solution method in both Dyna2D and Dyna3D. However, as a preliminary to studying the seismic source and its interaction with radiated seismic waves, it is useful to verify that elastic waves are propagated correctly using these codes. To test the three-dimensional code, I calculated the ground motion due to a

vertical strike-slip point source fault. The point source is shown schematically in Figure 1. Nine pairs of co-located nodes are used, and each node is displaced 0.09 m in the direction of shear with an impulsive velocity time history. Each node carries with it an area equal to the side of one element (500 m in this model), so this situation corresponds to a fault plane with area equal to nine element sides. With density 3000 kg/m^3, shear modulus 3 X 10^{10} N/m^2, poisson ratio 0.25, and assuming 10 grids per wavelength, the maximum frequency of this simulation is around 0.6 Hz. The point source is at a depth of 25 km, and the ground motion is sampled at a surface point 11.2 km away at an azimuth of 26.6°. I compare the resulting ground motion to the ground motion calculated by the well-tested reflectivity code Axitra (Coutant, 1994) (Figure 2). The velocity synthetics are lowpass filtered to 0.6 Hz. The fit between the different methods is quite good, implying that the three-dimensional finite-element code correctly models the propagation of seismic waves. Similar tests with the two-dimensional codes produced similarly accurate results.

While Dyna2D and Dyna3D are well-suited to solve a wide range of complicated problems (such as exploding missiles and cars crashing into walls), neither program includes a suitable fault/frictional contact algorithm. The problem is not a simple one. The desired behavior of a point on the fault is outlined in the introduction. To match this behavior our model must have the following characteristics: Before the fault ruptures, the fault should not have any physical manifestation—it should behave elastically, so seismic waves propagate through itself as though it were not there (assuming no physical material discontinuity

across the fault). When the shear stress on the fault exceeds the static frictional (yield) level, the fault should start to move in the fault-parallel direction under the competing forces of elasticity and friction. Finally, at some point the fault should stop slipping (heal) and return to elastic behavior.

One approach to implementing a fault in a finite element mesh is to model the two sides of the fault as different objects sharing a common boundary. The task of modeling a fault can be thought of as finding a boundary condition by which the two objects are coupled by frictional and normal forces. These forces are added to the elastic forces already operating on the nodes on the fault, and the finite element code solves for the particle motions via Newton's second law. In principle one may specify the force coupling between the two sides of the fault to be whatever is desired, but it is important to make sure that one's boundary condition is rooted in the physics of crack propagation and friction. The fault boundary condition implementation used in the present work is based on a suggestion by Edward Ziwicz (1997, personal communication). It is quite similar to the "traction at split nodes" method of Andrews (1999), although with the exception of the healing method it was implemented prior to the author's knowledge of the Andrews (1999) work.

A schematic diagram of a 2-D finite element mesh in the vicinity of the fault is shown in Figure 3. For clarity the nodes in a pair on the fault plane are shown to be a small distance apart, but in reality are co-located. Prior to rupture, the nodes on opposite sides of the fault should be locked together. However, simply

constraining them not to move would incorrectly preclude their moving in unison as the fault undergoes an overall displacement (perhaps due to seismic waves radiated elsewhere on the fault). Therefore the boundary condition (prior to rupture) is that the nodes on either side of the fault must have the same acceleration. Assuming an initial velocity of zero, this condition is sufficient to ensure their continued co-location. An advantage of this boundary condition is that it directly leads to the concept of a constraint force that is applied to both nodes to insure the same acceleration. Figure 4 shows a close-up view of a nodal pair on either side of the fault, with a finite distance between them introduced for clarity. Assume that the node on top has an elastic (or applied) force $\mathbf{F}_{e1}$ on it, and the bottom node has a force $\mathbf{F}_{e2}$ (Figure 4a). In general these forces are not equal, and neither are the masses of the two nodes (due to possibly different element shapes on either side of the fault). Thus in the absence of a correction force on each node, the two nodes will have different accelerations, and will move apart—in violation of our physical fault boundary condition. However, it is possible to solve analytically for a constraint force that can be subtracted from one node's force, and added to the other, to cause them to have the same acceleration. This condition is formulated mathematically by

$$\frac{\mathbf{F}_{e1} + \mathbf{F}_c}{m_1} = \frac{\mathbf{F}_{e2} - \mathbf{F}_c}{m_2} = a \qquad (1)$$

where α is the common acceleration of both nodes and $\mathbf{F}_c$ is the constraint force that serves as the coupling between the two sides of the fault. Solving equation 1 for $\mathbf{F}_c$ yields

$$\mathbf{F}_c = \frac{m_1\mathbf{F}_{e2} - m_2\mathbf{F}_{e1}}{m_1 + m_2}. \tag{2}$$

Note that in the case of $m_1 = m_2$, this expression reduces to $\mathbf{F}_c = (\mathbf{F}_{e2} - \mathbf{F}_{e1})/2$, or half the difference between the original nodal forces. This constraint force is shown added to the top node and subtracted from the bottom node in Figure 4b. With this constraint force applied, the final force on each node is

$$\begin{aligned} \mathbf{F}_1 &= \frac{m_1}{m_1 + m_2}(\mathbf{F}_{e1} + \mathbf{F}_{e2}) \\ \mathbf{F}_2 &= \frac{m_2}{m_1 + m_2}(\mathbf{F}_{e1} + \mathbf{F}_{e2}) \end{aligned}, \tag{3}$$

as shown in Figure 4c. Note again that if $m_1 = m_2$, the final forces are equal, and are simply the average of the two initial elastic or applied forces. Now the acceleration of both nodes is equal to

$$\mathbf{a} = \frac{\mathbf{F}_{e1} + \mathbf{F}_{e2}}{m_1 + m_2}, \tag{4}$$

which is equal to the acceleration of a single node of mass $m_1 + m_2$ under a force $\mathbf{F}_{e1} + \mathbf{F}_{e2}$. Thus, since $m_1 + m_2$ is equal to the mass of a single node elsewhere in the medium, the fault behaves as if it is a single line of nodes, subject to forces from both sides—identical to the rest of the medium.

The situation becomes more complicated for the cases of fault rupture and slip. However, the way to introduce this behavior into the model becomes clear

when one recognizes that the above expression (2) for the constraint force can be broken into the fault-parallel (shear) and fault-normal components. We identify the fault-normal component as the normal force across the fault, and the fault-parallel component as the frictional force. Thus, we may introduce a static frictional yield strength proportional to the normal force:

$$F_y = \left|\mu_s F_n\right|, \tag{5}$$

where μ_s is the static coefficient of friction on the fault and F_{n} is the constraint force component normal to the fault. When the fault-parallel frictional constraint force exceeds this value, it is then dropped to the sliding frictional level

$$F_s = \mu_d F_n \tag{6}$$

where μ_d is the dynamic (sliding) coefficient of friction, and it is implicit that this force is applied in the direction opposite that of the fault's slip. Because this shear force is now smaller than the fault-parallel constraint force required to keep the opposite nodes locked together, the nodes will move apart along the fault, corresponding to slip. An additional issue is the manner by which the stress drops. To keep singular behavior out of the model, it is helpful as well as physically plausible to introduce a gradual stress drop, either by a slip weakening distance (Ida, 1972; Andrews, 1976a+b) or by a slip weakening time (Nielsen, 1998). Introducing such a parameter allows the crack tip to absorb energy, keeping its speed down to physically reasonable (sub-elastic-wave speed) values, and reducing noise in the calculation.

Finally, after the fault has slipped, a method is needed by which to heal the fault, under the physically-observed constraint that backwards slip is to be avoided. In two dimensions, the easiest process is to monitor the slip velocity of the fault, and when it turns negative, to arrest the fault slip and return to the initial locked fault boundary condition. In the case of the current work, this was accomplished by returning to the "tied" boundary condition included in the codes Dyna2D and Dyna3D. However, in three dimensions such a healing criterion is difficult to formulate, because slip can be in any direction in the plane of the fault. Thus, there is no straightforward way to define backwards slip. We may arbitrarily define a preferred direction of slip based on the initial applied stresses (e.g., up-dip for a thrust fault) and stop the fault when that component of the slip velocity goes negative, but this formulation ignores the fact that the slip rate in the perpendicular direction may be far from zero, leading to an instantaneous fault deceleration that could radiate energy. In spite of these difficulties, this slip-rate reversal criterion was the method used in the initial three-dimensional work with planar faults (Chapter 4). However, for the final work on the non-planar faults, we have implemented a method similar to that suggested by Day (1998, personal communication), whereby the healing criterion is based on the slip velocity being small enough for the fault to heal during the next time step. This method is described in detail in Andrews (1999). The procedure is to look at each nodal pair and calculate at each time step the frictional force that would be required to heal these fault nodes (i.e., bring the slip velocity down to zero) in the next time step. If

that force is smaller than the frictional force (from Eq. 6), then the stopping force is applied, and the fault is healed. For most places on the fault this healing method is functionally identical to the previous method, except where there is a large rake rotation. In this case healing is later than with the previous method. In addition to being more physically justifiable, another advantage of this method is that there is no *a priori* preferred direction for slip imposed in the problem, leading to a more flexible code.

It is important to note that in the preceding development, only the nodes on the fault itself were discussed. The reason that the nodes away from the fault may safely be ignored is that for the propagation of seismic waves in a linearly elastic medium, only perturbations on the background stress level matter. The absolute level of stress in the surrounding medium cancels out of the equations of motion. Therefore, the forces need only be applied directly on the fault, with the rest of the medium stress-free. This pertubative approach is standard and used in almost all computational approaches to fault dynamics. It is possible that in cases where the assumptions of linear elasticity break down, such as when large-scale deformation or material failure occur, such an approach may be invalid, but for the problems in this thesis the perturbative approach is considered quite adequate. However, the flexibility of the finite element method in general and Dyna3D in particular make a non-perturbative, absolute-force reference frame possible for future work.

There are no analytical solutions for a completely spontaneous rupture with a slip-weakening friction law, so it is difficult to validate the above fault boundary

condition. However, Andrews (1999) describes a method by which to validate a similar case—the case of an expanding circular crack with a prescribed (constant) rupture velocity. In the case where the rupture velocity is less than the Rayleigh wave velocity, the analytical solution for the slip as a function of time and radius is (Kostrov, 1964)

$$\Delta u_1(r,t) = C(v_R)\left(\frac{\beta}{v_R}\right)\left(\frac{\Delta\tau}{\mu}\right)\left[(v_R t)^2 - r^2\right]^{1/2}, \tag{7}$$

where β is the shear wave velocity, $\Delta\tau$ is the stress drop, v_R is the rupture velocity, and μ is the shear modulus of the medium. Assuming that $v_R/\beta = 0.8$, the constant $C(v_R)$ is about 0.75. Since the slip velocity corresponding to (7) is singular when $r = v_R t$, Andrews approximates the slip velocity by

$$\Delta v_1(r,t) = \frac{\Delta u_1(r, t + D/v_R) - \Delta u_1(r, t - D/v_R)}{(2D/v_r)}. \tag{8}$$

This expression is an approximate derivative of (7) over a finite distance equal to $2D$. For comparison with the approximate analytical slip rate function above, we may use the stress following stress drop function in the finite element calculations (Andrews, 1999):

$$\begin{array}{ll} \tau = \tau_e & (r > v_R t + D) \\ \tau = \tau_y + (\tau_y - \tau_k)(r - v_R t - D)/2D & (v_R t + D > r > v_R t - D) \\ \tau = \tau_k & (r < v_R t - D) \end{array} \tag{9}$$

where τ_y is the shear stress at the time of rupture (when $r = v_R t + D$), and τ_k is the sliding frictional stress.

Analytical and computed slip rate functions for two points (corresponding to Mode II and Mode III rupture, respectively) are shown in Figure 5 for the case of grid spacing equal to 1.0, $v_R/\beta = 0.8$, $\mu = 1.0$, $\Delta\tau = 0.2 \text{ X } 10^{-5}$, and $r = 20$. Both slip rate functions show very good agreement with the approximate analytical solution, except for some slip rate oscillations after rupture, a slight undershoot of peak velocity at the mode 2 location, and a slight overshoot at the mode 3 location.

It is important to remember that the above slip rate function comparison is for a situation different from the spontaneous rupture used in the fault simulations for this thesis. Nonetheless, it does imply that the codes correctly calculate fault slip from a frictional stress drop. Similarly, the earlier point-source seismogram comparisons show that the codes correctly propagate the seismic waves caused by slip on a fault. For these reasons we believe that the computational method used in this thesis is valid within the bounds of the physical approximations of the model.

References

Andrews, D. J., Rupture propagation with finite stress in antiplane strain, *J. Geophys. Res., 81,* no. 20, 3575-3582, 1976a.

Andrews, D. J., Rupture velocity of plane strain shear cracks, *J. Geophys. Res., 81,* no. 32, 5679-5687, 1976b.

Andrews, D. J., Test of two methods for faulting in finite difference calculations, *Bull. Seismol. Soc. Am.*, in press.

Coutant, O., *Axitra* (computer program), 1994.

Ida, Y., Cohesive force across the tip of a longitudinal-shear crack and Griffith's specific surface energy, *J. Geophys. Res., 77,* 3796-3805, 1997.

Kostrov, B. V., Self-similar problems of propagation of shear cracks*, J. Appl. Math. Mech.*, *28*, 1077-1087, 1964.

Nielsen, S. B., Free surface effects on the propagation of dynamic rupture, *Geophys. Res. Lett.*, *25*, 125-128, 1998.

Whirley, R. G., B. E. Engelmann, and J. O. Hallquist, *DYNA2D: A Nonlinear, Explicit, Two-Dimensional Finite Element Code for Solid Mechanics - User Manual*. University of California, Lawrence Livermore National Laboratory, UCRL-MA-110630, 1992.

Whirley, R. G., B. E. Engelmann, and J. O. Hallquist, *DYNA2D: A Nonlinear, Explicit, Two-Dimensional Finite Element Code for Solid Mechanics - User Manual*. University of California, Lawrence Livermore National Laboratory, UCRL-MA-110630, 1992.

Zienkiewicz, O.C., and Y.K. Cheung, The Finite Element Method in Structural and Continuum Mechanics, McGraw-Hill, New York, 1967.

Figure Captions

Figure 1. Schematic diagram of a shear point source in the finite element code (shown in two dimensions for clarity). Each source node is given an impulsive slip rate time history. Source nodes are actually co-located, but are shown slightly displaced for clarity.

Figure 2. Comparison of synthetic seismograms from a point shear source. Solid = reflectivity (Coutant, 1994), dashed = 3-D finite element (Whirley and Engelmann, 1993). Both time histories are lowpass filtered to 0.6 Hz, which is the maximum resolvable frequency of the finite element mesh. The fit is quite good.

Figure 3. Schematic diagram of the boundary condition used to model fault rupture. The top and bottom media are separate finite element meshes, but the boundary nodes are coupled to each other via shear (frictional) and normal forces. Fault rupture is accomplished through manipulation of these coupling forces. Boundary nodes are actually co-located, but are shown slightly displaced for clarity.

Figure 4. Cartoon showing the application of the nodal constraint force that ensures that nodes on opposite sides of the fault have the same acceleration prior to rupture. Nodes are actually co-located, but are shown slightly displaced for clarity

a) Initially the finite element code calculates different nodal forces for the two nodes, which would lead to differential motion.

b) A constraint force $\mathbf{F}_c$ is calculated from the forces and masses of the two nodes, and then applied to each node.

c) The final nodal force configuration. The fault is now locked, but it may translate or move as a unit.

Figure 5. Comparison between the computed slip-rate functions and the analytical solution for points on the fault experiencing a) Mode II and b) Mode III rupture.

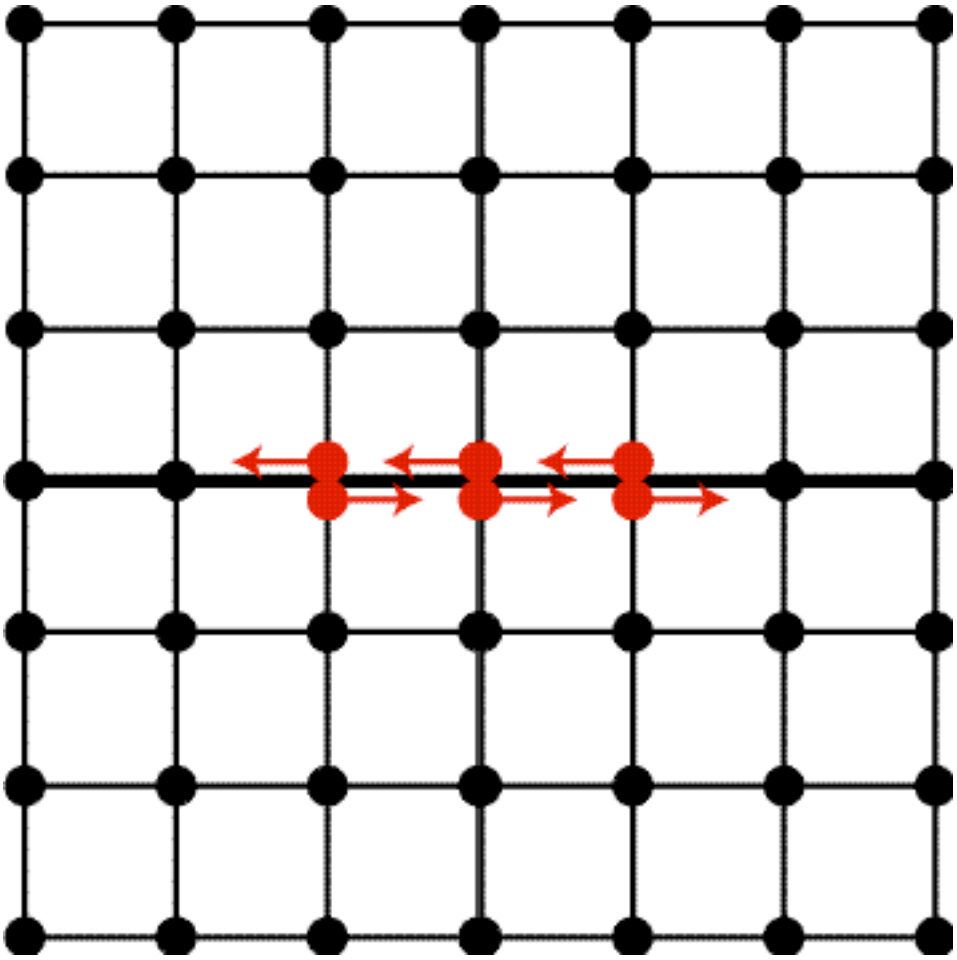

Figure 1.

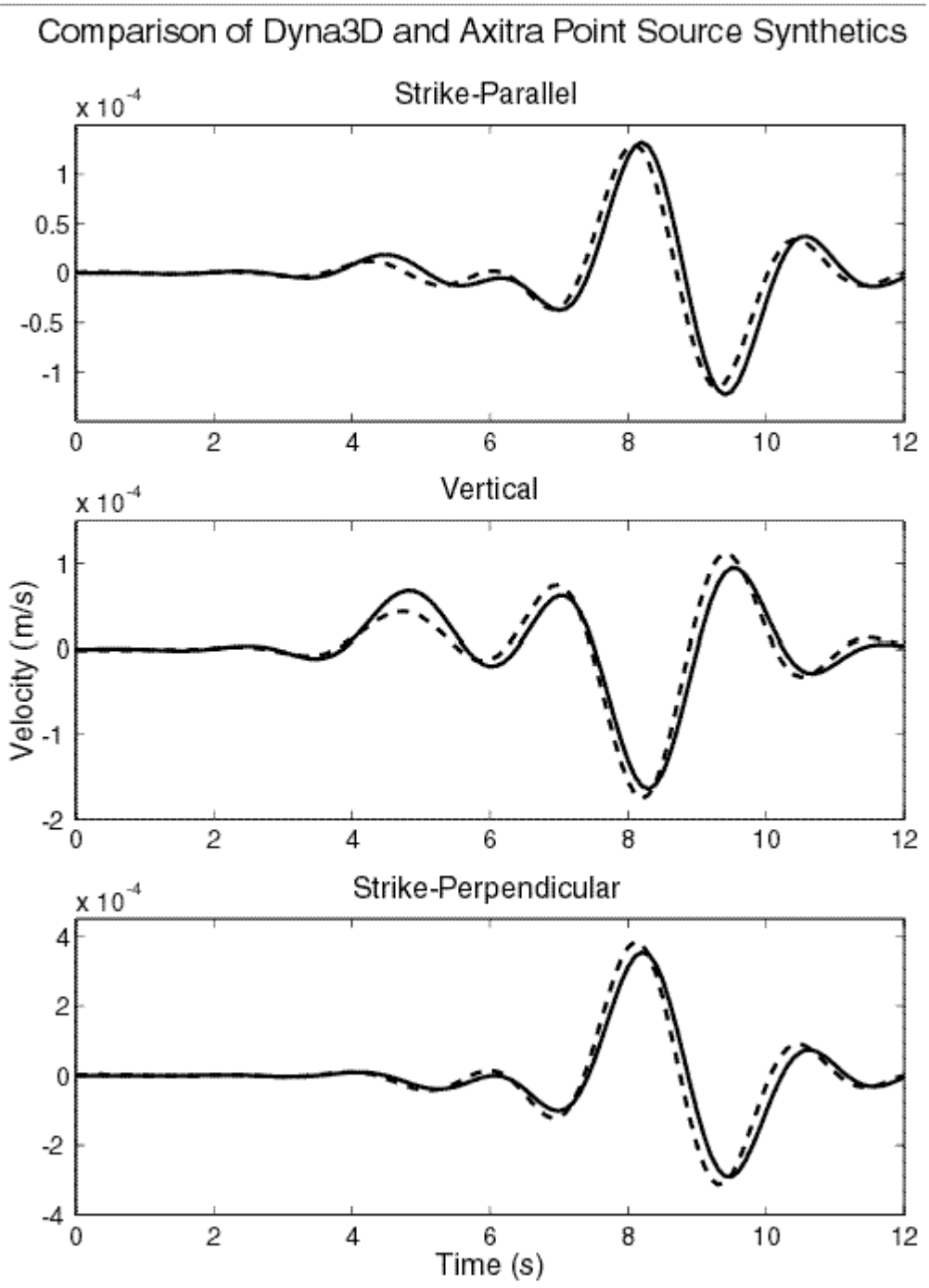
Comparison of Dyna3D and Axitra Point Source Synthetics
Strike-Parallel
x 10^-4
1
0.5
0
-0.5
-1
0
2
4
6
8
10
12
Vertical
x 10^-4
Velocity (m/s)
1
0
-1
-2
Strike-Perpendicular
x 10^-4
4
2
0
-2
-4
Time (s)

Figure 2.

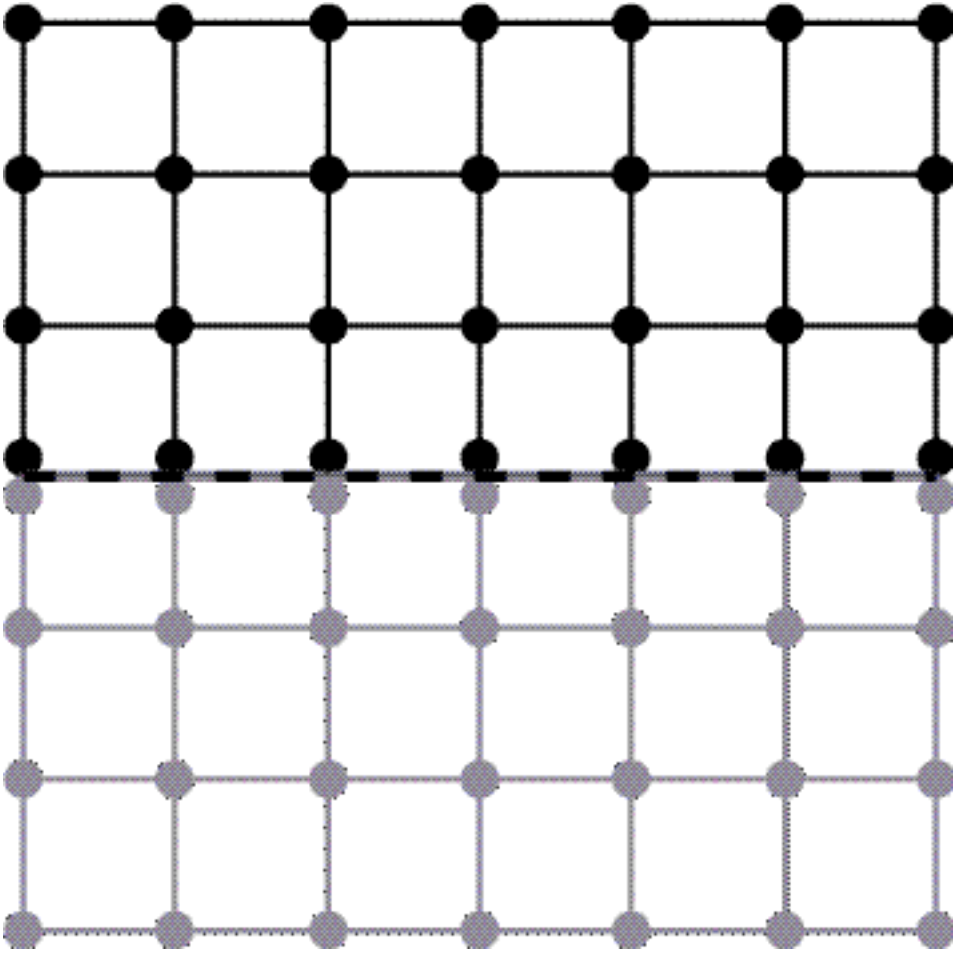

Figure 3.

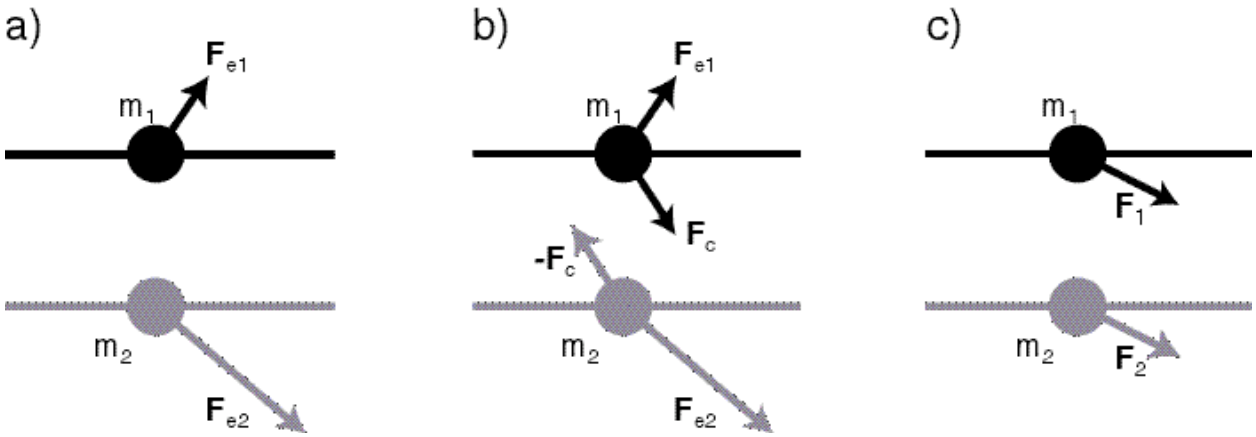

Figure 4.

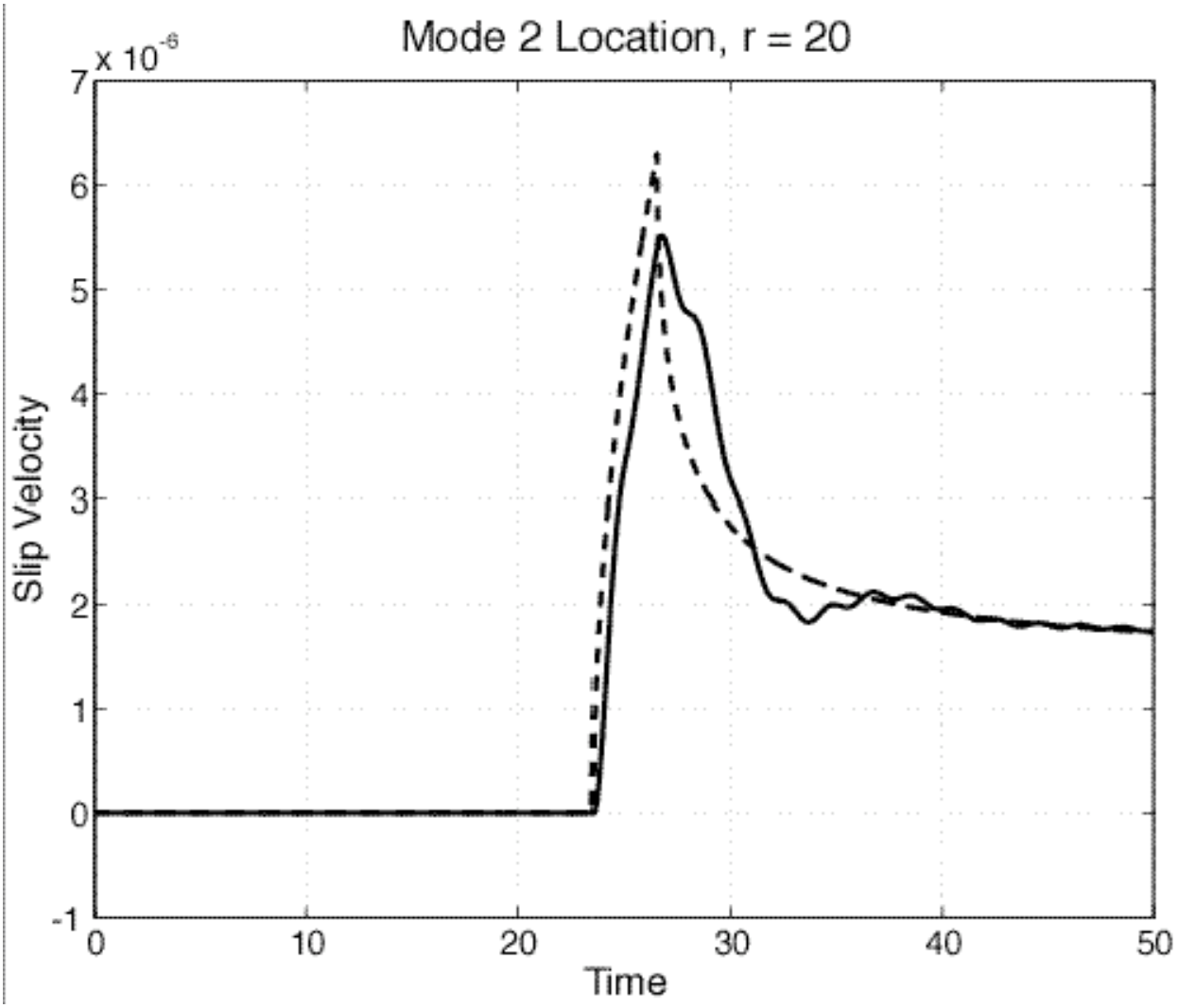
Mode 2 Location, r = 20
x 10^-6
7
6
5
4
3
2
1
0
-1
Slip Velocity
0
10
20
30
40
50
Time

Figure 5. a)

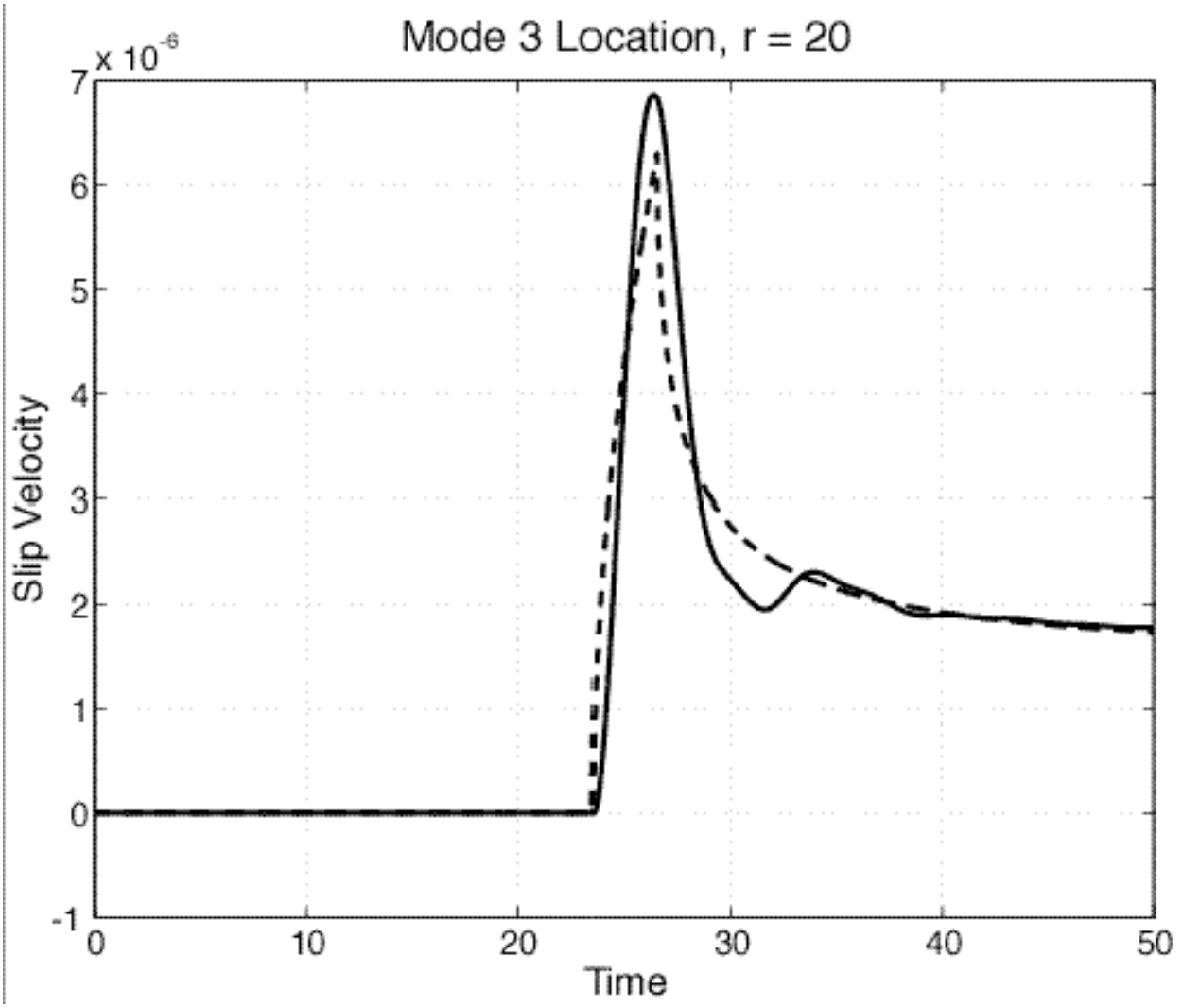
Mode 3 Location, r = 20
x 10^-6
7
6
5
4
3
2
1
0
-1
Slip Velocity
0
10
20
30
40
50
Time

Figure 5. b)