

SIMULATING THE EFFECT OF A SHALLOW WEAK ZONE
ON NEAR-SOURCE GROUND MOTION

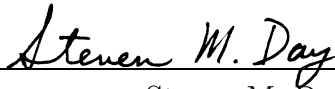
A Thesis
Presented to the
Faculty of
San Diego State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
in
Geological Sciences

by
Geoffrey P. Ely
Spring 2001

THE UNDERSIGNED FACULTY COMMITTEE APPROVES

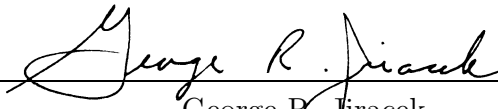
THE THESIS OF GEOFFREY P. ELY



Steven M. Day, Chair
Department of Geological Sciences

April 12, 2001

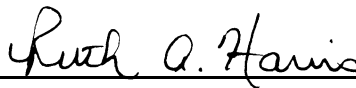
Date



George R. Jiracek
Department of Geological Sciences



José E. Castillo
Department of Mathematical Sciences



Ruth A. Harris
Department of Geological Sciences

SAN DIEGO STATE UNIVERSITY

Spring 2001

Copyright 2001

by

Geoffrey P. Ely

DEDICATION

To Mom and Dad.

ACKNOWLEDGEMENTS

I am very grateful to Professor Steve Day for being a most patient and supporting mentor. Thanks to Abdolrasool Anooshehpour and James Brune for providing original data from their experiments and assistance in setting up my model. This work was funded by the Southern California Earthquake Center, and the Institute of Geophysics and Planetary Physics, Los Alamos National Laboratory.

TABLE OF CONTENTS

	PAGE
ACKNOWLEDGEMENTS	v
LIST OF TABLES	vii
LIST OF FIGURES	viii
CHAPTER	
I. INTRODUCTION	1
II. DESCRIPTION OF THE MODELS	5
Physical Model	6
Numerical Model	10
III. CODE PARALLELIZATION	15
Parallel Algorithm	17
Benchmarks	20
IV. NUMERICAL MODELING RESULTS	23
Validation	23
Further Analysis of the Numerical Model	28
Sensitivity Study	35
V. CONCLUSIONS	41
APPENDIX	
CODE USAGE AND INPUT FILES	43
REFERENCES	47
ABSTRACT	50

LIST OF TABLES

TABLE		PAGE
1	Model Parameters	14
2	Normalized Surface PA	38

LIST OF FIGURES

FIGURE	PAGE
1 Model diagram	7
2 Friction in foam	9
3 Slip-weakening friction in the numerical model	11
4 Friction comparison	13
5 Weak zone friction comparison	14
6 Diagram of MPI communications	19
7 Parallel performance of the modeling code.	22
8 Comparison of acceleration records	25
9 Comparison of peak acceleration along the fault	28
10 Comparison of displacement records	29
11 Comparison of acceleration records near the fault with records at the fault	31
12 Snapshots of particle velocity, no weak zone	32
13 Snapshots of particle velocity, 20 cm weak zone	33
14 Peak acceleration and peak velocity at the surface	34
15 Peak acceleration and peak velocity ratios at the surface	35
16 Comparison of acceleration records, modified μ_s	36
17 Comparison of displacement records, modified μ_s	37
18 Comparison of displacement records, modified μ_w	38
19 Comparison of acceleration records, modified μ_w	39
20 Snapshots of particle velocity, 20 cm weak zone, modified μ_w	40

CHAPTER I

INTRODUCTION

Currently, scientists have a rather incomplete understanding of the physics of earthquake sources. This is, in part, due to the difficulty in measuring key dynamic properties of faults, such as tectonic forces acting on the fault plane as well as the frictional properties of fault wall rocks. Additionally, capturing earthquakes in action is hindered by their unpredictable nature and the inaccessibility of the subsurface source region for placing instrumentation. Very few ground motion records exist for the near-fault region of large earthquakes. In fact, no records at all are available within 10 km for very large, strike-slip earthquakes comparable to, for example, the $M_w = 7.9$ 1906 San Francisco, California earthquake (Hanks and Kanamori, 1979). Yet, the need for realistic ground motion predictions is critical for seismic hazard analysis and structural design. In lieu of real data, numerical earthquake modeling provides a tool for quantifying near-fault ground motion effects. This study performs dynamic three-dimensional finite-difference simulations of strike-slip earthquake rupture. The computer code is adapted for multiprocessor computers, thus enabling bigger and more realistic simulations. Model validation is performed using data from physical model experiments in which foam rubber scale models were used to simulate earthquakes. Physical models can be valuable validation tools because they have relatively few unknown properties, and can provide significantly more thorough recordings of ground motion than are available for real earthquakes. The physical model data set used is from an experiment performed by Brune and Anooshehpour (1998). Both the physical model and the numerical model include a special treatment of the near-surface part of the fault (upper two to three kilometers in real faults). The reduced stresses and distinct frictional properties of the near-surface section are termed a "weak zone" in this study.

Ground motion from shallow earthquakes can be strongly enhanced at sites very close to the surface trace of the fault rupture, relative to more distant sites. The high

intensity and damage potential of near-fault ground motion is due both to the proximity of the source and to the occurrence of pronounced directivity effects (e.g., Somerville et al., 1997). The strong motion near the surface trace of large earthquake ruptures is frequently pulse-like in waveform. That is, most of the ground displacement takes place in a coherent, high-velocity (often exceeding 1 m/s) pulse of short duration (typically 2-4 seconds), with the strongest motion usually polarized in the direction perpendicular to the fault (e.g., Archuleta and Hartzell, 1981; Anderson and Bertero, 1987; Hall et al., 1995). Such pulses can be highly damaging to structures, and (because of the nonlinearity of structural response to high-amplitude ground motion) reliable modeling of the performance of a particular structure may require constraints on the pulse waveform, as well as estimates of its amplitude and duration (Hall et al., 1995).

Most large (moment magnitude greater than roughly 7), shallow earthquake ruptures extend to the earth's surface, and the amount of co-seismic slip at the surface is often comparable to that occurring at depth. The appropriate representation of this surficial slip in earthquake models is problematical. Near-fault ground motion predictions for surface-rupturing earthquakes are highly sensitive to the rate of slip on the shallowest several kilometers of the fault. If high slip-rate faulting persists to shallow depths (less than 2 or 3 km), earthquake models predict unrealistically large ground accelerations at short distances (of order 2 or 3 km and less) from the fault trace (e.g., Anderson and Luco, 1983). Therefore, most computational modeling of ground motion, if it has considered the near-fault region at all, has simply excluded the upper several kilometers of the model fault from slipping. However, artificially excluding shallow slip from models of large, surface-rupturing earthquakes must lead to some degree of underestimation of the longer-period components of near-fault ground motion, and may undermine efforts to accurately predict the amplitude, duration, and waveform of near-source velocity pulses.

The underlying problem is that current earthquake models have limited facility to account for the distinct physical properties and stress state in the upper several kilometers of faults. Brune and Anooshehpour (1998) give a good summary of these physical considerations. Firstly, there are good reasons to expect the shallow zone to be mechanically weak, and therefore unable to store and release significant elastic strain energy. Sec-

ondly, the presence of thick, unconsolidated gouge layers, as is likely in the near-surface of well-developed faults, may lead to velocity-strengthening frictional behavior (Marone and Scholz, 1988). In that case, shallow slip would occur without a local dynamic stress drop. Alternatively, if a dynamic stress drop does occur in the shallow zone, it may develop gradually, i.e., with a low slip-weakening slope, as suggested by Ide and Takeo (1997) on the basis of their space-time slip image of the 1995 Kobe, Japan earthquake. Either way, the shallow slip would be expected to occur at reduced slip velocity (longer rise time).

In kinematic earthquake models, the fault slip-rate distribution is prescribed *a priori*, and kinematic models can accommodate the above physical considerations only through *ad hoc*, and very uncertain, modifications to the slip model. Simulations based on dynamic (spontaneous rupture) earthquake models have the potential to reduce this uncertainty, since the rupture growth and slip distribution in dynamic models evolve in response to the conservation laws of continuum mechanics, rather than being prescribed *a priori*. There are still highly uncertain model inputs, such as the distributions of fault strength and shear and normal prestresses, but at least these are physically meaningful quantities for which a range of estimates can be constructed on the basis of laboratory measurements and independent geophysical studies.

Even with dynamic models, establishing the validity of the mathematical formulation behind a numerical earthquake model poses a significant challenge. Successful modeling of earthquake recordings may be a necessary, but not sufficient, component of model validation. Strong motion record sets for earthquakes are sparse and highly aliased spatially, observations are almost exclusively confined to the earth's surface, and the recording sites are usually far removed from the fault plane. The physical scale model experiments of Brune and Anooshehpour (1998), in contrast, permit acceleration time histories to be sampled in the interior of the model, directly adjacent to the simulated fault, and can therefore provide recordings of rupture propagation, slip, and wave motion of a type unavailable for real earthquakes. Furthermore, model properties such as the wavespeeds and friction coefficients are well known through laboratory testing, limiting the free parameters available to the numerical model. These controlled experiments

therefore provide a valuable test of numerical modeling methodology.

Brune and Anooshehpour's (1998) experiment uses a foam rubber scale model to assess the effects on ground motion when there is a low frictional-strength surficial layer (weak zone) present on the fault. This study simulates the physical model of Brune and Anooshehpour with 3D finite-difference numerical calculations. The mathematical formulation and numerical methodology (elastodynamic solver, boundary conditions, fault formulation, spontaneous rupture criterion) are identical to those used in earlier work by Day (1982a, b) and in more recent studies by Harris and Day (1999) and Magistrale and Day (1999). The primary objective is to validate the numerical model through comparison with the physical model. A secondary objective is to use the flexibility and detail provided by numerical modeling to gain additional insight into the mechanical behavior of the physical model, and the effects of a weak zone in general.

This thesis is organized into four chapters. Chapter I is this introduction which discusses the scientific relevance of the research and important background information. Chapter II reviews the methods used by Brune and Anooshehpour for their foam rubber scale modeling, and details the numerical modeling methods. Chapter III covers the parallel computer code development beginning with the motivation for parallelizing the code, followed by a description of the parallel algorithm, and finally performance benchmarks for the code. Chapter IV presents the results of the model validation and weak zone study. The appendix lists the input files used for the numerical modeling simulations.

CHAPTER II

DESCRIPTION OF THE MODELS

Brune and colleagues have used foam rubber blocks to model a wide range of earthquake processes including, nucleation, predictability, and rupture mechanism (Brune et al., 1990), interface separation as a possible solution to the heat flow paradox and the paradox of large overthrusts (Brune et al., 1993), and frictional heat generation and seismic radiation (Anooshehpour and Brune, 1994). Physical models of this type provide an opportunity to validate numerical models under well-controlled conditions. The foam rubber model of Brune and Anooshehpour (1998) is particularly relevant to the problem of near-fault ground motion excitation. It incorporates a weak near-surface layer in the fault zone. The weak layer represents the effects on fault friction caused by the thick gouge that builds up along well-developed natural faults (e.g., Marone, 1998) and which may significantly affect near-fault ground motion. The weak layer can also be caused by thick surficial deposits of sediments.

Used together, physical and numerical modeling are complementary approaches to studying earthquake source physics. The physical processes of the foam rubber earthquake model are recorded more completely and in greater detail than is possible for natural events in the earth. The foam model allows independent measurement and control of physical parameters such as normal stress and tectonic loading rate as well as direct measurement of other parameters such as prestress, static stress-drop, and friction. In contrast, in numerical simulations of natural earthquakes, these parameters usually must be inferred indirectly from geophysical observations (e.g., geodetic or seismic observations), or simply left as free parameters to be estimated by trial and error fitting of the simulation results to the observations. An additional advantage is that the physical model provides improved observations of the fault slip and ground motion. Sensors can be placed adjacent to the fault plane close to the rupture process, and experiments can be run many times to build large statistical samples. This type of data is valuable for

validating the numerical modeling method, and comparable data sets are not available for natural earthquakes. Foam models have the further advantage of being real physical systems (rather than mathematical approximations as in numerical modeling). They, therefore, provide an independent check not only of numerical code logic, but also of the inherent adequacy of the discrete representation of the continuum required in numerical modeling. Such validation is particularly important in highly nonlinear problems such as those involving rupture and friction.

However, physical models have certain limitations that a well-validated numerical model is better able to contend with. Physical models can be affected by artifacts due to (1) artificial boundaries, (2) the finite stiffness of the loading apparatus, and (3) disturbances to the medium such as those caused by the mass and rigidity of embedded sensors. The stiffness of the loading apparatus is not a major issue in the case of the foam rubber model since the loading apparatus can be approximated as infinitely stiff in comparison with the highly compliant foam rubber (Brune and Anooshehpour, 1998), but the issue is relevant in the case of most rock mechanics experiments. Of the three problems, only artificial boundary effects apply to numerical models. Even then, they can be controlled, for example, either through boundary conditions that absorb most incident energy to mimic an unbounded medium, or simply by greatly extending the computational domain to minimize boundary effects. Additionally, numerical models also allow unlimited flexibility for varying parameters. This allows sensitivity studies to be performed to determine how different parameters influence the rupture process. Numerical models also provide detailed spatial resolution of particle motion which gives a more complete picture of the rupture process than is evident from isolated sensors.

Physical Model

The foam rubber model of Brune and Anooshehpour consists of two stacked blocks of foam rubber, $1 \times 2 \times 2.5 \text{ m}^3$ each (Figure 1). The bottom of the lower block is fixed to the floor and the system is loaded by slowly shearing the top of the upper block using a piston mounted to the wall. The shearing induces episodic unstable sliding (stick-slip) events along the interface (fault plane) between the blocks. The individual stick-slip

events are the model earthquakes that are the focus of this study.

The initial normal stress (σ_{n0}) on the fault can be controlled by adjusting the supporting jacks of the upper block and was set to 3.3 mbar (330 Pa) for the set of experiments studied here. The upper block was driven at 1 mm/s by the piston. Characteristic slip events average about 1 cm of slip across the entire fault plane. The objective is to model strike-slip motion on a vertical fault; therefore one of the vertical sides parallel to the drive direction is designated as the ground surface (the ground surface is indicated in Figure 1 by arrows showing relative motion).

An array of five accelerometers is embedded in the fault wall of the lower block at the locations shown in Figure 1. The accelerometers have a single component oriented

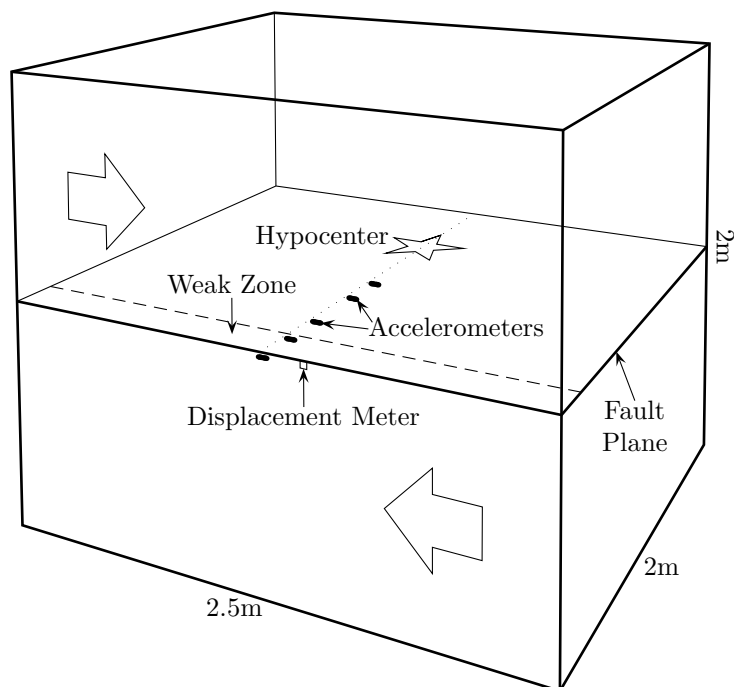


Figure 1: Diagram of foam rubber model model for simulating strike-slip earthquakes. The upper block is forced horizontally past the fixed lower block, building up stress and causing rupture along the horizontal fault plane. The arrows indicate the relative motion of the two blocks and are drawn on the surface corresponding to the ground surface of the earth. A weak zone on the fault is present near the ground surface. The accelerometers have a single component oriented in the strike direction, and are embedded in the fault wall three centimeters from the fault plane.

parallel to the strike of the fault and they are offset about 3 cm from the fault plane. Also, fault slip is measured by a sensor located adjacent to the fault trace at the ground surface. The sensor records the displacement of one wall of the fault relative to the fixed external reference frame. Since the stick-slip events are short in duration compared with the time for reloading by the piston motion, shearing displacements of the fault walls are equal in magnitude and opposite in direction during the events. The displacement can therefore be interpreted as a record of one half of the total fault slip.

A force meter was used to record stress on the fault. The average shear prestress (τ_0) measured while the fault was locked up just prior to rupture was 7.3 mbar (730 Pa). The average final stress (τ_f) was 6.1 mbar which translates to a static stress drop ($\Delta\tau$) of 1.2 mbar. These values reported by Brune and Anooshehpour (1998) are averages over many events. Some of the events are double events, so the stress drop measurement may be slightly higher than the value for a single event. As an independent check, stress drop can be estimated from the static displacement. For small, uniform slip over the plane in an elastic material, $\Delta\tau$ is directly related to static slip ($s_f = |\mathbf{s}(t = \infty)|$) by the formula $\Delta\tau = \rho V_s^2 s_f / L$ where L is the thickness of the model, ρ is density, and V_s is S-wave speed. Properties of foam rubber include a density of 16 kg/m³, S-wave speed of 30 m/s, and a Poisson's ratio of 0.3. The slip for the particular events that will be looked at is 16 mm, which implies a 1.2 mbar static stress drop, consistent with the above estimate obtained from force meter measurements.

An important feature of foam rubber events is that during the slip pulse, the walls of the fault separate (Brune et al., 1993). As the fault opens, frictional resistance quickly goes to zero. The bulk of the sliding takes place while the fault is open. Then the fault closes and frictional resistance abruptly rises back up again. Figure 2 shows a hypothetical curve relating how frictional resistance may evolve as the fault slips. Although dynamic stress drop is quite high (perhaps as high as the prestress), static stress drop is only a fraction of the dynamic stress drop. There is no strong evidence for widespread occurrence of opening in real earthquakes, although it cannot be ruled out on the basis of current knowledge, and there is some evidence for its plausibility in shallow thrust events (Allen et al., 1998).

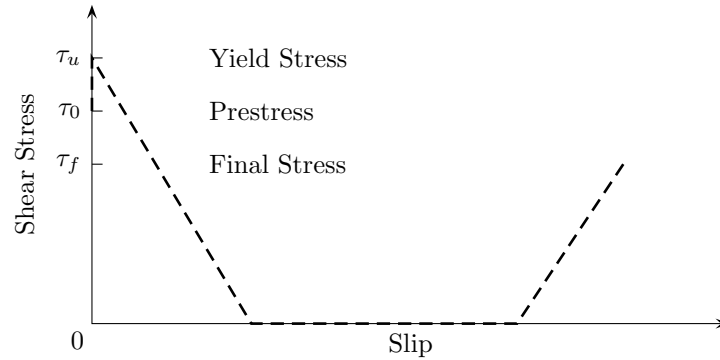


Figure 2: A plausible graph of friction in foam related to slip on the fault. The shape of the curve is speculative, but it is known that friction goes to zero for some time (due to fault opening) before rising back up again.

To model the weak zone, strips of plastic are inserted into the fault interface along the ground surface. With the plastic material present, the foam fault has two important properties that are intended to mimic weak zones in earth faults: the coefficient of friction is significantly lower than on the part of the fault with foam surfaces in contact, and the coefficient of friction increases with sliding velocity (velocity strengthening). Rock mechanics experiments have shown that the presence of a thick gouge layer produces similar effects in rock and that the velocity strengthening behavior leads to stable slip rather than stick-slip sliding (Marone and Scholz, 1988). The absence of seismicity in the upper few kilometers of well-developed faults may be the result of stable sliding due to thick accumulations of gouge. In contrast, the foam by itself is strongly velocity weakening. In fact, due to opening in the rupture pulse the highest slip velocities occur at zero sliding friction. To determine the frictional properties of the plastic material, Brune and Anooshehpour (1998) did separate tests with the entire fault covered with plastic. Under the same 3.3 mbar of normal stress, shear stress was measured at different sliding velocities ranging from less than 1 mm/s up to 130 mm/s. Sliding stress at pre-event creep velocity of 1 mm/s was less than 1 mbar and increased monotonically to 1.7 mbar at 130 mm/s sliding velocity. Measurement was not possible of the sliding stress at the typical sliding velocity of rupture (1 m/s). By extrapolation from the values in the measured range, Brune and Anooshehpour estimated this value to be 2.3 mbar.

Numerical Model

The numerical simulations employ the method used by Day (1982a, b). The linearized continuum equations of motion for an isotropic viscoelastic solid (Kelvin-Voigt model) are

$$\sigma_{ij} = \lambda(u_{k,k} + \gamma \dot{u}_{k,k})\delta_{ij} + \mu(u_{i,j} + \gamma \dot{u}_{i,j} + u_{j,i} + \gamma \dot{u}_{j,i}) \quad \dot{v}_i = \frac{1}{\rho}\sigma_{ij,j} \quad \dot{u}_i = v_i$$

where $\boldsymbol{\sigma}$ is a stress tensor; \mathbf{u} and \mathbf{v} are displacement and velocity vectors; λ and μ are elastic moduli; ρ is density; $\gamma = dt\beta$, the product of the time step dt and a viscosity damping factor β ; and δ_{ij} is the Kronecker delta function. $\boldsymbol{\sigma}$, \mathbf{u} and \mathbf{v} are functions of 3D space and time. A dot over a variable indicates a time derivative. λ , μ , ρ , and β are function of 3D space. These equations are solved by finite difference approximations that are second order in space and time. Using a staggered grid approach, \mathbf{u} is defined on the grid nodes, spatial derivatives of \mathbf{u} (e.g., $\boldsymbol{\sigma}$) are defined on the cell centers between the nodes, and time derivatives of \mathbf{u} (e.g., \mathbf{v}) are defined at times which leapfrog the times where \mathbf{u} is defined. Time is stepped explicitly, which means there is no prescribed relationship among the new nodal values; they are computed independent of one another directly from the state of the model at the previous time step. Additionally, borrowing from finite-element analysis, bending modes of the grid cells are accounted for and included in the calculations. A reflecting boundary condition is imposed by requiring that $\boldsymbol{\sigma}\hat{\mathbf{n}}$ be zero on the boundary, where $\hat{\mathbf{n}}$ is the unit normal to the surface.

The region is discretized to $220 \times 220 \times 140$ nodal points at a spacing of 2 cm and a time step of 0.15 ms. The dimensions ($4.4 \times 4.4 \times 2.8 \text{ m}^3$) are made much larger than the foam model to prevent unwanted artificial reflections from the model boundary from affecting the near-fault solution. Due to the low Q (high attenuation) of foam (on the order of 10), reflections appear to have minimal affect on fault behavior in the foam model despite the smaller dimensions. The larger dimensions of the numerical model provide the added advantage that the evolution of the wavefield can be observed as it propagates away from the fault without the complication of side-boundary reflections.

Faulting is formulated as a special boundary condition over a planar surface within the model, using a dynamic, slip-weakening friction law (Ida, 1972). The frictional

strength of the fault is the product of normal stress σ_n and the coefficient of friction $\mu(l)$ which depends on the on the slip path length l . ($\mu(l)$ should not be confused with the elastic modulus μ). When the magnitude of shear stress (τ) is less than the frictional strength, the fault behaves elastically. Slip occurs when necessary to ensure that τ is bounded by the frictional strength. The amount of relative displacement between the fault walls (slip) is denoted \mathbf{s} , and l is given by $l = \int_0^t |\dot{\mathbf{s}}| dt$. For the slip-weakening model the coefficient of friction (shown in Figure 3) is

$$\mu(l) = \begin{cases} \mu_s - (\mu_s - \mu_d)l/d_0 & l \leq d_0 \\ \mu_d & l > d_0 \end{cases}$$

where μ_s and μ_d are coefficients of static and dynamic friction and d_0 is the slip-weakening distance. This friction law is a simplification of rate-and-state friction models (e.g. Dieterich, 1979). The simplification entails neglect of the rate dependent effect, and has been shown to provide good approximation to the full friction law under the high slip-rate conditions of dynamic rupture propagation (Okubo, 1989). Although the rate dependent effects are important to earthquake nucleation, they have minimal influence on the rupture once it has begun, and so can be safely ignored here. Slip-weakening friction laws of this type have been successfully applied to model earthquakes (e.g., Ida, 1972; Andrews, 1976; Day, 1982b), and have been shown to be consistent with seismic recordings from past earthquakes (Ide and Takeo, 1997; Olsen et al., 1997; Day et al., 1998).

Since a primary goal of this study is to validate the existing numerical mod-

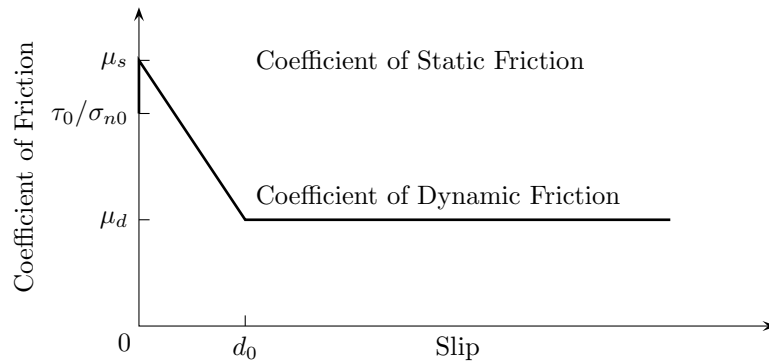


Figure 3: Plot of the slip-weakening friction model. The curve represents the coefficient of friction as a function of the amount of slip on the fault.

eling method in preparation for further application simulating real earthquakes, this slip-weakening model is retained without change rather than formulating a new fault constitutive equation that mimics the opening seen in foam rubber.

A result of the different friction behavior in the two models is that for a given static stress drop, the foam model has a much greater dynamic stress drop than the numerical model, and thus is capable of generating stronger accelerations. Since the numerical modeling effort primarily aims to reproduce the foam rubber acceleration records, some means of compensating for the greater dynamic stress drop is needed. To achieve this compensation, the coefficient of static friction μ_s was treated as a free parameter. This is a natural choice since no measurement of μ_s was reported for the foam rubber used in the experiment. Its value was adjusted by trial and error to produce good agreement between the measure and simulated acceleration records on the fault. The result is a preferred value for μ_s of 2.4.

This value may not necessarily be a good estimate for μ_s of foam however. Since this is the only free parameter in the model, it must accommodate any discrepancies in the dynamics of the two models, and it is clear that the dynamics of the two models are quite different. Lower μ_s and higher dynamic stress drop both lead to faster, more energetic rupture. It is possible that the μ_s of the numerical model is lower than the true value, which would compensate for the higher dynamic stress drop in the physical model. An independent measure of the μ_s of foam rubber would be valuable for confirming or disproving this hypothesis. Figure 4 shows schematically the difference between friction and stress drop in the two models. In spite of the significant differences in the behavior of friction in the two models, the numerical model is successful in reproducing many of the important results of the foam rubber model (see Chapter IV).

Prestress τ_0 was set to the value measured by Brune and Anooshehpour (1998) in the foam experiment (7.3 mbar), and the coefficient of dynamic friction μ_d was set to the ratio of the final stress τ_f (6.1 mbar) and initial normal stress σ_{n0} (3.3 mbar) measurements. The slip-weakening distance d_0 was set equal to the typical size of the vesicles in the foam rubber (1 mm). In keeping with the approach of attempting to validate the numerical simulation method in its established form, velocity-strengthening

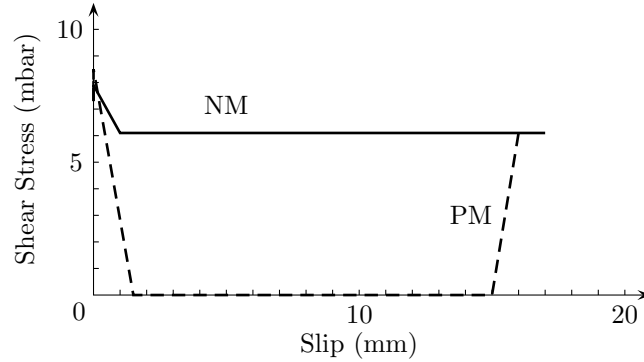


Figure 4: Comparison of friction in the numerical (NM) and physical (PM) models.

was not incorporated into the weak zone. However, it was roughly approximated by setting the slip-weakening parameter d_0 to zero and using a constant coefficient of friction equal to the estimated value for typical sliding velocity ($\mu_w = 0.6$). Figure 5 compares the measured and extrapolated slip-rate dependence of the physical model weak zone friction with the corresponding behavior of the numerical model.

The fault ends at a depth of 140 cm. Events are artificially nucleated at the bottom of the fault by reducing the coefficient of friction to μ_d over an expanding semi-circular area. The radius of the semi-circle expands at 15 m/s until it reaches 40 cm, after which the rupture is able to continue spontaneously. Modeling parameters are summarized in Table 1.

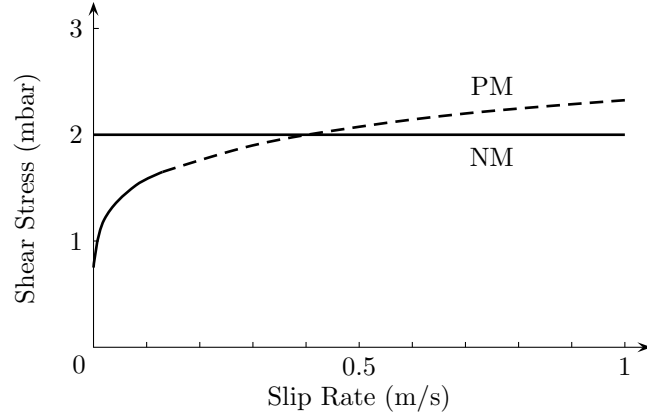


Figure 5: Comparison of friction in the weak zone in relation to sliding velocity for the physical model (PM) and the numerical model (NM). The solid part of the PM curve was measure in the laboratory; the dashed part is extrapolated. Friction is constant in the weak zone for the numerical model.

Table 1: Model Parameters

General Parameters		
Nodes	220 × 220 × 140	
Cell spacing, dx	2 cm	
Time step, dt	0.15 ms	
S wave speed, V_s	30 m/s	
P wave speed, V_p	56 m/s	
Density, ρ	16 kg/m ³	
Damping factor, β	0.5	
Hypocenter depth	140 cm	
Weak zone depth range, h	0–40 cm	
Fault Parameters	Plain Foam	Weak Zone
Prestress, τ_0	7.30 mbar	0.66 mbar
Initial normal stress, σ_{n0}	3.30 mbar	3.30 mbar
Coefficient of static friction, μ_s	2.4	0.6
Coefficient of dynamic friction, μ_d	1.85	0.6
Slip weakening distance, d_0	1 mm	n/a

CHAPTER III

CODE PARALLELIZATION

Numerical earthquake models need to capture a wide range of length scales to be useful. The overall dimensions must be long enough to contain the size of the fault as well as the distance to any locations at which a record of ground motion is desired. They also must be detailed enough to resolve short wavelength particle motion, the rupture process, material property variations, and complicated fault zone geometries. Since it takes multiple grid points to resolve one wavelength, this range of length scales requires huge amounts of computer memory and processing time to run effective finite-difference (FD) problems. An increase in spatial resolution of the model requires an increase in the number of operations that goes as the fourth power of the increase factor. For example, doubling the spatial resolution requires an eight fold increase in memory and a doubling of the number of time steps (the time increment must be reduced proportionally to the smallest length increment to preserve numerical stability). FD models can never fully represent a continuum, but the finer the spatial discretization, the more fully we will be able to incorporate small-scale processes into the physical model, and the greater our demands on computer processing power will be.

Parallel supercomputers are an increasingly available resource for high performance computing. Over the decade of the 1990s multiprocessor computers displaced vector computers as the dominant architecture of the worlds fastest supercomputers (see Dongarra et al., 2000 for statistics). Parallel supercomputers have benefited from the large research and development investment in commodity microprocessors. Vector supercomputers, on the other hand, require costly engineering efforts to boost performance, and consequently are not advancing at the same pace. In addition to parallel supercomputers, multiprocessor workstations and servers are now commonly used for day-to-day computing. Parallel computers can be categorized into shared memory and distributed memory machines. In a shared memory system, all processors have access to a common

memory. Shared memory machines benefit from very rapid communication of data between processors, but are limited in their scalability (Culler et al., 1999). The scalability limit arises from problems coordinating access to the same memory locations by large numbers of processors. Alternative to shared memory systems are distributed memory systems, in which each processor can only access its own local memory. Information must be transferred between processors through a message passing scheme. Distributed memory systems are more easily scaled to large numbers of processors. Massively Parallel Processing (MPP) systems are one type of distributed memory computer that have high numbers of processors with relatively little amounts of memory per processor. Many of the fastest computers are of this type. Another strategy for constructing supercomputers is interconnecting multiple smaller machines through fast networking to form a cluster. The individual machines of a cluster can themselves be shared memory multiprocessors. This forms a hierarchy in which memory is shared within each groups of processors and distributed among the groups. Clusters are currently the fastest growing type of supercomputer (Dongarra et al., 2000). Modest clusters can be assembled relatively inexpensively from commodity components, although specialized networking equipment is often used. The performance of clusters varies widely depending on the speed of the communication network.

An impediment to using parallel computers is that codes must be written specifically to take advantage of parallel processing (although in some cases compilers can do this automatically). This introduces new complexities and pitfalls. Not all programs can in fact be parallelized. One must be able to identify portions of an algorithm that can run independently and simultaneously on different processors. Fortunately, FD algorithms of the general type used in the current study are readily parallelizable for two reasons: FD operators are local operators, with only nearest neighbor interactions among nodes, and the discretization is on a grid where the nearest neighbors are in simple logical relationship with each other. The current trends in the computer industry combined with the need for bigger and more accurate numerical earthquake models provides ample motivation for parallelizing our earthquake model.

Writing a parallel computer code often involves adapting a working serial version

the code rather than building a parallel code from scratch. The original serial version of the Fortran code used for this study has been used in numerous successful research projects over two decades. Two recent examples are Magistrale and Day (1999) and Harris and Day (1999). In the process it has been scrutinized for errors and tested on a variety of problems, and in some simple cases has been validated against analytical solutions. Clearly, for a well-established code like this, it is desirable to avoid major reprogramming when going parallel. This makes automatic parallelizing compilers attractive since they require no alterations to the original source code. Unfortunately, tests (to be described below) reveal that our code is not well-suited for automatic parallelization. Another consideration is that auto-parallelizing compilers are not always installed on every system, so relying on them hurts code portability. For these reasons explicit parallelization was undertaken using the Message-Passing Interface (MPI) described in Snir et al. (1998) and Gropp et al. (1999). Message passing programs are directly analogous to distributed memory parallel computers in which each processor has its own memory space, and information can be passed between processors. MPI is universally available for nearly all current multiprocessor platforms including shared and distributed memory systems. Well-coded MPI programs are highly portable.

Parallel Algorithm

The finite-difference algorithm used in this study is derived by discretizing the continuum equations of motion onto a 3D rectilinear mesh. Calculations progress as time is stepped over discrete values. At each time step, the new values at a node in the mesh are calculated from the current values at that node and the nearby nodes. The result is that the new values for any node are not dependent on any nodes outside its immediate vicinity. In other words, the FD operators are local operators. Because different regions of the mesh can be worked on independently, FD algorithms of this type are highly parallelizable.

We use a domain decomposition scheme, where the mesh is divided into subdomains, and each processor works on a different area of the mesh. This approach has been used previously in FD solutions to wave propagation problems (e.g., Olsen et al., 1995).

Domain decomposition can be done along any combination of the three coordinate axes. That is, the mesh can be subdivided along a combination of planes of constant x , y or z . For this study the mesh is only subdivided along planes of constant z . For nodes that lie within the interior of the subdomains, the nearest neighbor nodes lie within the same subdomain, and computations are straightforward. However, for nodes on the subdomain boundary, some of the nearest neighbors lie in the adjacent subdomain. Thus communication of information between processors is necessary to complete the computation of the boundary nodes.

When considering communications between processors, one must tailor the algorithm to the target computer platform in order to achieve good performance. Two important considerations are the number of processors and communication latency of the machine. Latency is the time it takes to initiate a communication between two processors, and can be measured by timing a communication with a content of zero length. On a system with significant latency it is advantageous to save up communications and send large blocks of data together. Then the cost of initiating a communication is incurred only once rather than multiple times for a group of data. This can be accomplished in a domain decomposition algorithm by setting up overlapping subdomains. The boundary nodes of each subdomain are duplicated on the adjacent subdomain forming a buffer zone.

Single axis domain decomposition with a buffer is illustrated schematically in Figure 6. For simplicity, two-dimensions (2D) are shown instead of three. In the figure, the circles represent nodes of the discrete model. The complete model space is described by the combination of all the solid circles. The open circles represent redundant nodes that form the buffer zones along the boundary of each subdomain. Every node of the model space (filled circles) is located in the interior of a subdomain and can be correctly updated in a given time step without interprocessor communication. Then, at the completion of each time step, the buffer zone nodes are updated by copying the updated values from the adjacent processor (where the corresponding nodes are interior nodes). This exchange of updated values is indicated by the arrows in Figure 6. In this way interprocessor communication only takes place once per time step instead of individually

for each node. The scheme works well only if the number of processors is small compared to the number of nodes that are being distributed between processors. If the number of processors is comparable to the number of nodes, then the buffer zone becomes a dominant part of the total storage and computation cost, reducing the solvable problem size. For example, with single axis domain decomposition if the number of nodes in the decomposition direction is equal to the number of processors, a three fold increase in storage and computational operations would be required for buffers. A three way decomposition could require as much as a 27 fold increase in memory and computation. MPP systems have high numbers of processors and low latency communications. One would not use a buffer when optimizing for a MPP system, but would rather program for individual communications as needed. Existing serial codes often require extensive rewriting to perform efficiently on MPP systems. Clusters, on the other hand, are good candidates for buffered algorithms since they generally have low numbers of processors, large amounts of memory per processor, and relatively slow communications.

Our code uses buffers and thus is optimized for efficient communications, but not for high numbers of processors. This choice was made for two reasons. First, the buffer scheme required little modification of the existing serial code. Communications routines were able to be contained in a separate subroutine rather than dispersed throughout the code. This makes it a simple matter to compile the code with or without parallel support. A few well-placed preprocessor commands in the code allow the inclusion of the

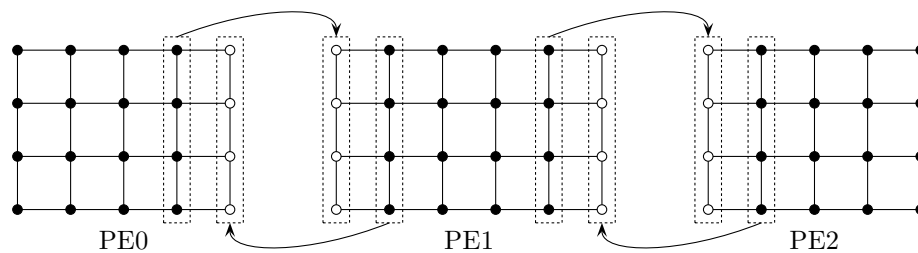


Figure 6: Diagram of communications between processors for the parallel algorithm. The lattice represents a 2D model space distributed between three processors elements with the circles representing nodes in the model. The open circles are buffer zone nodes that are duplicated from the adjacent processor. Dashed boxes and arrows show the transfer of data between processors.

parallel processing routines to be controlled by command line options at compile time. The second reason for using a buffer is that none of the systems available for this study are MPP systems where a buffer would be detrimental. And, in the future, we anticipate increasing availability of cluster systems where efficient communications are the highest concern.

Although it is not relevant to the choice of algorithms (a choice between buffered and unbuffered communications), it should not be forgotten that in addition to the number of processors and latency, communication bandwidth is also an important factor in performance of a parallel code. Bandwidth is the rate at which data can be transmitted between processors. The total time to transmit a block of data is the latency time plus the amount of data divided the bandwidth. For the two algorithms (with and without a buffer) the total amount of data transmitted per time step is the same. The only difference is that in one scheme data are transferred in small blocks and in the other case data are transferred in large blocks. Since the net transfer is the same, bandwidth affects each method equally, and so is not a factor in the choice.

Benchmarks

In order to test the capability of the code, a series of benchmark tests were performed on different systems. For the tests, the number of processors was varied while timing model runs of a fixed size. A scalable code will continue to gain a performance advantage as more processors are added. For a perfectly scalable code, the speedup (serial runtime divided by parallel runtime) is equal to the number of processors. Poorly scalable codes may see speedup at low numbers of processors, but not at higher numbers. In order to utilize the full potential of a particular parallel computer, a code must scale well up to the number of processors on the system in use.

Scalability tests were performed on two shared memory machines: a four processor Sun Enterprise 450 named Moho, and a 96 processor SGI Origin 2000 at Los Alamos National Laboratory named Theta. Theta tests were limited to the maximum permitted number of processors per job—64. Since with our method, larger models have the potential to be more realistic, typical applications usually make use of the maximum

available resources of a system. So, a useful benchmark ought to estimate how the code will perform when the model is as large as possible and occupies the full machine memory. Moho can accommodate a model of 384^3 nodes and Theta can accommodate 768^3 nodes, with some memory to spare on each. Benchmarks were done with a problem size of $384 \times 384 \times 64$ on Moho and $768 \times 768 \times 64$ on Theta. These sizes were chosen as a compromise between benchmarking a full sized problem and completing the benchmarks in a reasonable amount of time. Because the sizes of the benchmark models differ by only one order of magnitude from the full sized models, it is likely that they provide an accurate estimate of performance for full sized models. Since all time steps are computed identically, the benchmarks need only compute one step. The time required for program start up and initialization routines is a large fraction of the runtime for a single time step benchmark, but is negligible for typical long running model applications. Therefore, only the main program loop was timed for the benchmarks.

Automatic parallelization using the Sun WorkShop f77 5.0 compiler (Sun, 1999) on Moho showed little performance gain (Figure 7). A maximum speedup of 1.6 was achieved with three processors. Three processors was also found to be the limit of scalability since the code ran slower with four processors than with three.

For the MPI benchmarks, one-dimensional (1D) domain decomposition was done along one of the longer problem dimensions. Since the length of the long dimension is equal to the length of a full sized model, the benchmark model exactly reproduces the parallel partitioning that would occur in a full sized model. Benchmarks for both Moho and Theta are shown in Figure 7. For some cases the code achieved better than ideal (super-scalar) speedup for small numbers of processors. Super-scalar speedup is seen on Theta for two processors, and on Moho for three and four processors. These cases are shown in Figure 7 by the places where the speedup curves rise above the ideal speedup curve (dashed line). This is likely due to using a fixed problem size for the benchmarks. With a fixed problem size, each processor must handle a smaller piece of the total problem as processors are added. This means that a larger percentage of the problem can reside in cache memory on each processor. Cache memory access is much faster than main memory access. When a processor requests a stored memory value, if that value is not

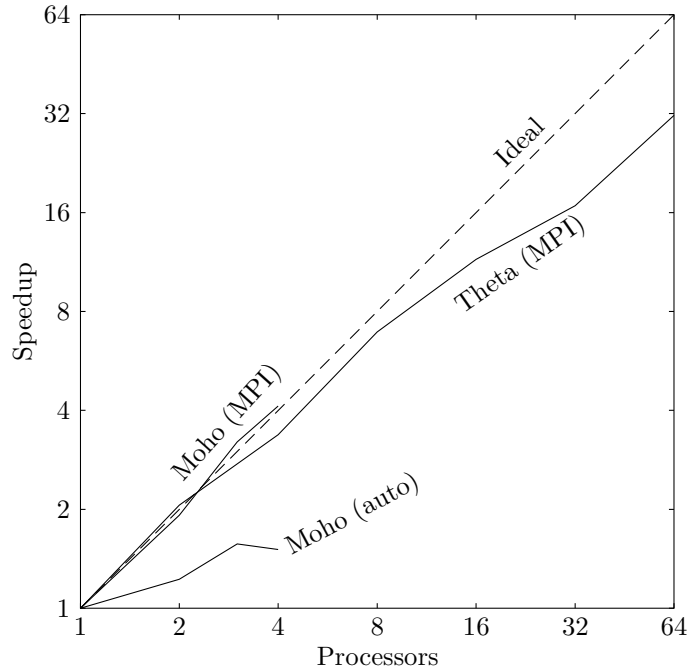


Figure 7: Parallel performance of the modeling code. Plot shows speedup as a function of the number of processors for two machines (Theta and Moho). For Moho automatic compiler performance is shown for comparison. Super-scalar (better than ideal) speedup is achieved in some cases for small numbers of processors.

presently in cache memory, it must then find the appropriate section in main memory and transfer it to the cache. Frequent swapping of cache memory with main memory can impact code performance.

Scaling on Theta is nearly ideal up to eight processors (yielding 87% efficiency). After eight processors, scalability degrades slightly. At 64 processors, speedup is 32 (50% efficiency). This is a modest sacrifice of efficiency in exchange for the capability that the parallel code provides for doing much larger problems. Together with the large memory capacity of computers like Theta, our modeling capabilities are significantly enhanced with the parallel code. Theta contains 96 gigabytes of RAM memory, which can accommodate a problem size of $800 \times 800 \times 800$ in our code. Typical problems of this size would compute in about 24 hours using 64 processors.

CHAPTER IV

NUMERICAL MODELING RESULTS

There are two goals for the numerical model. The first is to synthesize, as well as possible, the data from Brune and Anooshehpour's (1998) laboratory experiments in order to validate our modeling method. The second is to use the advantages of the numerical model to further analyze weak zone effects.

Validation

Validation was performed by comparing results from Brune and Anooshehpour's (1998) experiment to results from the numerical model. Most parameters are fixed based on the direct measurements of Brune and Anooshehpour. The exception is the coefficient of static friction, μ_s , for which no direct measurement was published. It is treated as a free parameter and is adjusted to improve the fit to the data. The data used for the comparison are acceleration records from the five near-fault, strike-parallel sensors and fault displacement records from the surface. Because the accelerometer locations do not coincide exactly with nodes of the numerical model, nearby nodes are used for the comparison. The accelerometers are located 3 cm from the fault, while the nodes are 4 cm from the fault. The acceleration data are evaluated in two ways. First, acceleration time histories are compared for two particular events, one with no weak zone ($h=0$) and one with a 20 cm wide weak zone ($h=20$). Second, peak acceleration averaged over many events is compared for a range of cases where h ranges from 0 to 30 cm.

The numerical model is successful in reproducing many of the important aspects of the physical model acceleration records. To achieve close agreement in absolute acceleration levels and rupture velocity, the coefficient of static friction (μ_s) in the foam part of the model (i.e., not in the weak zone) was set to 2.4. The acceleration records for the $h=0$ and $h=20$ cases are shown in Figure 8. The figure shows records from the foam experiment as well as numerical results from the same approximate locations as

the accelerometers. The peak acceleration (PA) for each record is noted on the curves. Also, apparent rupture velocity is estimated by picking the time of the maximum positive amplitude of adjacent traces and dividing the up-dip distance by the time difference of the picks. For the physical model, this is an estimate of the apparent velocity in the up-dip direction, and if the true rupture direction is oblique to this direction, the true rupture velocity may be slightly lower than this estimate. For the numerical model this is the true rupture velocity since the hypocenter is placed directly down-dip. The picks are connected by dashed lines in Figure 8 and the rupture velocity is noted on each line segment.

The acceleration records in Figure 8 show the strong similarities between the physical and numerical model results. A number of key features are common to both models. The main pulse is similar in shape, consisting of a positive acceleration pulse of about 6 ms duration, followed by a shorter-period negative pulse. The full pulse lasts about 10 ms. In the absence of a weak zone, the maximum acceleration grows monotonically as the rupture progresses up dip, and this behavior of the physical model is reproduced by the numerical model. With the addition of the weak zone, both numerical and physical models show an abrupt decrease in acceleration amplitude at the near-surface station. When no weak zone is present, the main rupture pulse reflects at the surface and continues back down the fault, producing a second, somewhat lower amplitude pulse of slip acceleration. This is most evident in the sensor just below the surface sensor, at about 15 ms after the main pulse, and the behavior is reproduced very closely in the numerical model. Even details such as the amount of time delay between the initial phase of slip and the surface-reflected rupture phase are reasonably well reproduced (within about 10%) in the numerical simulation. When a weak zone is present, the main pulse is suppressed as it enters the weak zone and there is no strong reflection of the rupture front. The reflection is also suppressed in the numerical model when the weak zone is present.

The estimates of apparent rupture velocities, in both physical and numerical models, are generally 27 ± 1 m/s which is slightly slower than (about 90% of) the S-wave velocity of 30 m/s. Two things are remarkable about the apparent rupture velocities in Figure 8. The first is that the numerical and physical models agree so closely in their

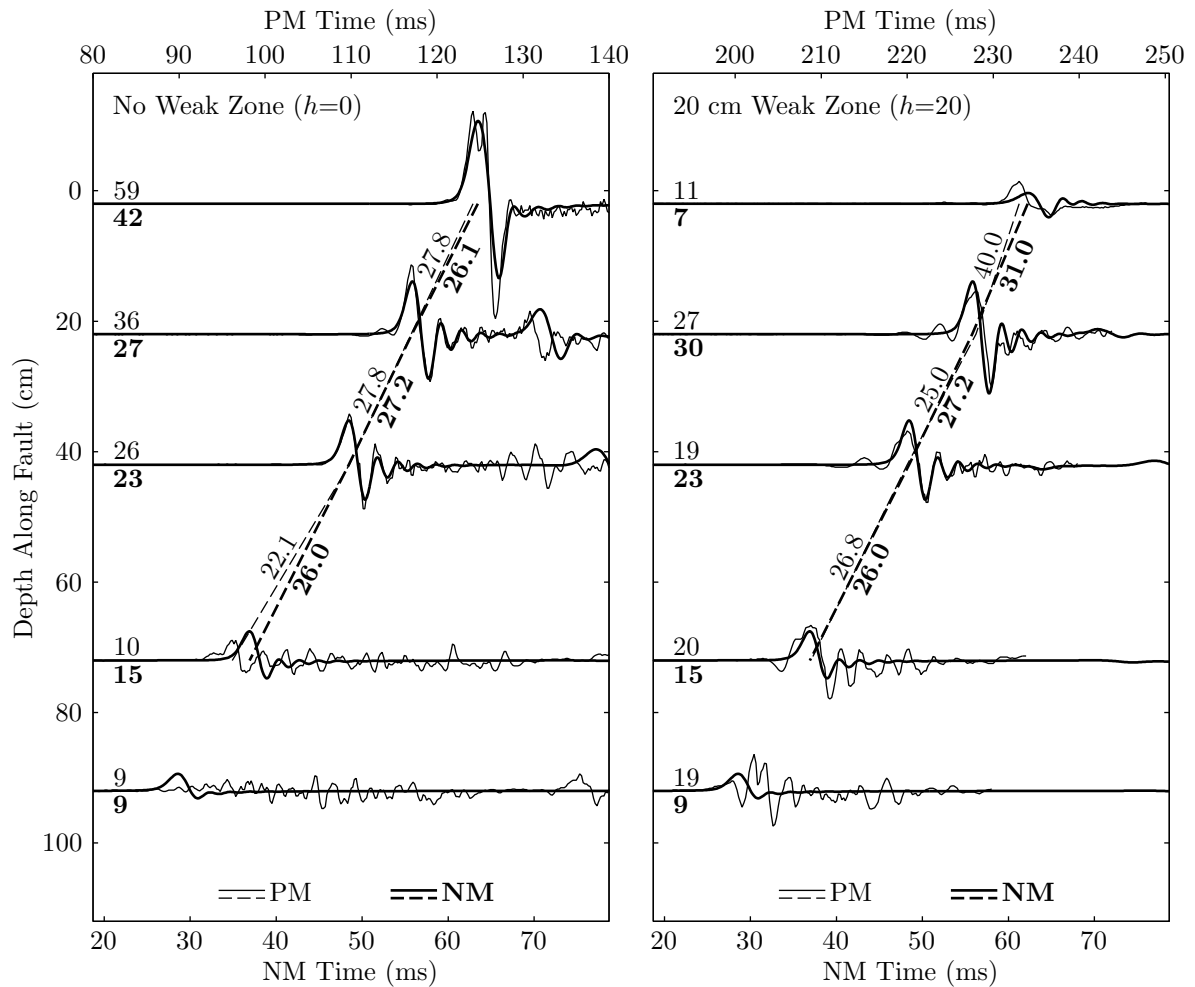


Figure 8: Comparison of acceleration records along the fault from the physical model (PM, thin curves) and the numerical model (NM, thick curves) for the cases $h=0$ (left) and $h=20$ (right). The curves have a common amplitude scale, and the peak acceleration (in units of g) is noted at the start of each curve (plain text for the PM and bold text for the NM). Dashed lines connect times of maximum positive amplitude and indicate the rupture velocity of the main pulse. The rupture velocity (in m/s) is noted on the dashed lines.

quantitative predictions. For the case of no weak zone, for example, the average rupture velocities in the two models are virtually indistinguishable. The second is that the introduction of the weak zone has a very similar qualitative effect in both models. When the rupture enters the weak zone, the apparent up-dip velocity increases to above 30 m/s. This effect is present in both physical and numerical models, being more pronounced in the physical model. The occurrence of apparent rupture velocity above the S-wave speed of 30 m/s as the rupture enters the weak zone requires further comment. Normally, the S-wave speed is considered the physical limit on the rupture velocity in mode III crack extension (antiplane strain), the mode which predominates here in the up dip direction (see Freund (1990) for a complete discussion). One consideration is that the rupture velocities for the physical model are apparent velocities in the up dip direction, and if the true rupture direction is oblique to this direction the apparent up-dip velocity that may be slightly higher than the true rupture velocities. However, the high velocity occurs when the weak zone is added to the model, it is localized at the weak zone, and it is present in the numerical model as well (where it is known that the actual rupture direction is up-dip). Therefore, the high rupture velocity is almost certainly not a simple geometrical artifact due to an oblique rupture direction. A velocity in the weak zone exceeding the S speed would not be physically precluded in this case. Since the rupture on the strong part of the fault travels slower than the S-wave speed, the S-waves from that part of the fault arrive in the weak zone ahead of the rupture front, and they could trigger early rupture within the weak zone. However, a more detailed analysis of the numerical solutions, which provides an alternative explanation of the apparent velocity estimates, will be given in a later section.

Acceleration amplitudes in the physical and numerical models can be compared quantitatively, although some caution is called for. Near the fault, acceleration falls off rapidly with distance from the fault, causing the amplitude of the acceleration records to be highly sensitive to sensor location. As previously noted, the node locations in the numerical model are not exactly coincident with the sensor locations of the physical model, so there should be some discrepancy in amplitude between the two models. Also, event magnitude and hypocenter location are variable in the foam rubber model, and

are not known for the two particular events for which we have examined the acceleration time histories. To minimize these variations, Brune and Anooshehpour (1998) looked at normalized peak acceleration (PA) averaged over many events. This offers a consistent comparison with the numerical model since numerical model parameters such as prestress and static stress drop are based on values given by Brune and Anooshehpour that were averaged over many events rather than values that they measured for the two particular events. Figure 9 shows normalized PA curves for both the physical model (left panel, reproduced from Brune and Anooshehpour (1998)) and the numerical model (right panel). The normalized curves are formed from the ratio of PA relative to a reference point at depth. The reference point is at 57 cm depth which is well below any weak zone effects. In addition to the two model events, $h=0$ and $h=20$, the PA curves also include data for weak zone thicknesses of $h= 7.5, 15,$ and 30 cm. The numerical model normalized PA curves show that for $h=0$ the amplitude also varies rapidly with depth near the rupture break-out at the free-surface, so the physical model amplitude at the near-surface sensor is probably also highly sensitive to sensor depth as well as distance from the fault, and possibly even affected by the finite size of the sensors. Figure 9 shows important similarities between the model results. In both models the PA at the surface is reduced by the presence of the weak zone, and the wider the weak zone the greater the reduction factor. At the near-surface sensor, for the no weak zone case ($h=0$), the normalized PA is 2.5 for the physical model and 1.9 for the numerical model, a 25% difference. For $h=20$ the normalized PA at the surface is 0.44 for the physical model and 0.31 for the numerical model, a 29% difference.

Figure 10 compares measurements from the displacement sensor located on the fault trace at the position shown in Figure 1. The sensor measures half of the total slip on the fault. Measurements by Brune and Anooshehpour (1998) are shown by the thin curves, and the numerical model results are shown by the thick curves. The left panel shows the no weak zone case ($h=0$) and the right panel shows the 20 cm weak zone case ($h=20$) The two models are in good agreement for the no weak zone case. For the 20 cm weak zone case the numerical model under-predicts slip. Later in the Sensitivity Study section, friction in the weak zone is adjusted to help bring the results into better

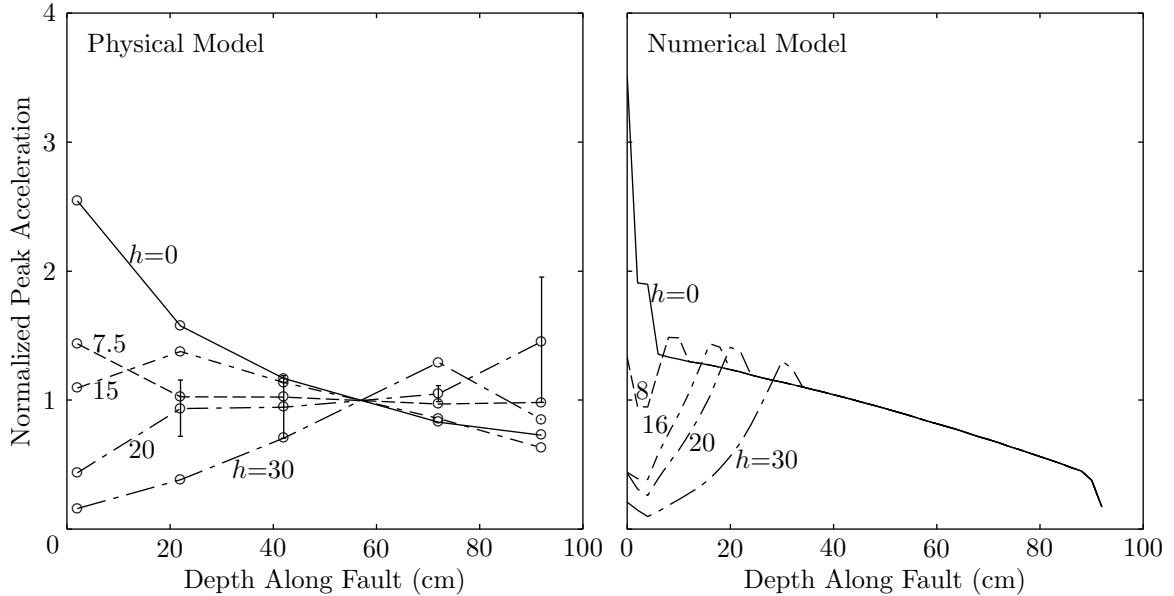


Figure 9: Comparison of peak acceleration along the fault. Physical model values are normalized to the value halfway between the sensors located at 40 and 70 cm depth and are averaged over multiple events. Numerical model curves are normalized to the recorded value at 56 cm depth. Weak zone thickness is indicated by h .

agreement.

Further Analysis of the Numerical Model

A unique benefit of the numerical model is that each node in the finite-difference discretization is, in effect, a sensor. This provides a high resolution record of motion for the entire model. One can use the high resolution provided by the numerical solution to learn more about the dynamics of the fault model. In the acceleration records (Figure 8) the rupture propagates from below up to the surface. It appears in both the physical and the numerical model as though the velocity of the rupture increases when it enters the weak zone. The high resolution data from the numerical model can be used to look at this more closely. Figure 11 compares synthetic acceleration records for the upper 40 cm of the fault. The top panels shows vertical record sections that are located 4 cm from the fault for $h=0$ and 20. These are more detailed views of the same profile that is shown in Figure 8, and again rupture apparently speeds up as it enters the weak zone. The bottom panels show what happens at the nodes directly on the fault. Each curve is

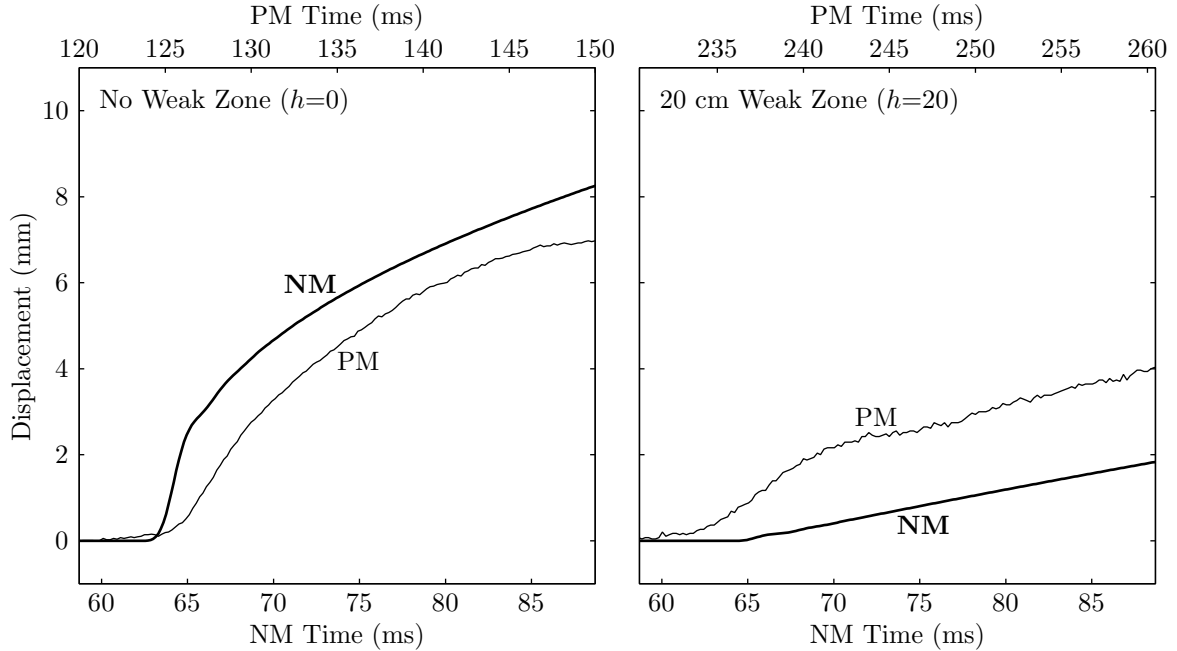


Figure 10: Comparison of displacement (half of the slip) at the surface for the physical foam rubber model (thin line) and the numerical model (thick line). The left panel shows the no weak zone case ($h=0$) and the right panel shows the 20 cm weak zone case ($h=20$).

normalized by its PA which is noted (in g) on each curve. The larger accelerations for the profile at the fault illustrate how rapidly acceleration decreases with distance from the fault. In contrast to the near-fault records, the on-fault records show that the rupture actually slows down in the weak zone. The apparent increase in rupture velocity recorded near the fault merely reflects the fact that when rupture dies out, a strong S-wave pulse continues to travel up dip and is observed at the sensor locations 3 cm (4 cm in the numerical model) off the fault plane. This effect can also be seen in Figures 12 and 13 which show snapshots of particle velocity on a cross-section of the fault through the hypocenter. Figure 12 shows the $h=0$ case and Figure 13 shows the $h=20$ case. The fault vertically bisects each image. The color indicates the strike-parallel component of velocity (particle motion is in and out of the page). Since, the plotted plane is also the plane of symmetry for the bilateral rupture, the motion is purely strike-parallel. For the $h=20$ sequence the rupture enters the weak zone at the 58.5 ms frame and particle velocity on the fault begins to decrease. By 61.5 ms the rupture has clearly slowed while the S-wave pulse remains strong a few centimeters from the fault. The unbroken part of

the fault can be seen as a narrow white strip in the top 20 cm of the model in the 61.5 ms panel of Figure 13. This unbroken strip then gradually narrows as the rupture continues to the surface between 61.5 ms and 66 ms. When the weak zone ruptures, there is no local release of strain energy to offset the frictional work done. Therefore, the rupture front is weakened and no strong reflected front is evident following the surface breakout (at about 66 ms) in Figure 13. In comparison, the no-weak-zone ($h=0$) case in Figure 12, a very strong reflected rupture front runs back down the fault beginning at about 64.5 ms.

We now turn our attention to motion at the ground surface of the model. Figure 14 shows surface profiles of PA and peak velocity (PV) (strike parallel component) for a range of weak zone thicknesses. The profiles are perpendicular to the fault on a line through the epicenter and extend 80 cm away from the fault on the ground surface. No data were recorded in this area in Brune and Anooshehpour's (1998) experiment. The top curve is the $h=0$ case and successively lower curves are increments of 4 cm in weak zone thickness. The bottom curve is the $h=40$ case. When no weak zone is present, or when the weak zone is shallow, both PA and PV diminish rapidly with distance from the fault. In these cases, PA diminishes away from the fault much more quickly than PV. Both PA and PV near the fault are greatly reduced as the h get larger, and for very deep weak zones PA and PV are smaller at the fault than they are some distance away. In order to highlight the weak zone effects we look at the ratios of PA and PV relative to the case with no weak zone. Figure 15 shows the ratio of PA (upper panel) and PV (lower panel) to values for no weak zone, as a function of distance from the fault. From this it is clear that the lateral distance from the fault over which PA and PV are diminished by the weak zone increases with weak zone thickness. For example, for $h=20$, PA is reduced for all distances inside 14 cm, and for $h=40$, PA is reduced inside 27 cm. Outside the zone of decreased PA lies a zone in which PA is increased by almost a factor of two in the presence of a weak zone, relative to the case with no weak zone. PV decreased everywhere relative to the no weak zone case, but the region of highest reduction is near the fault, and this region widens with weak zone depth. The zone of increased PA is a somewhat surprising prediction of the numerical model, and we have no measurements

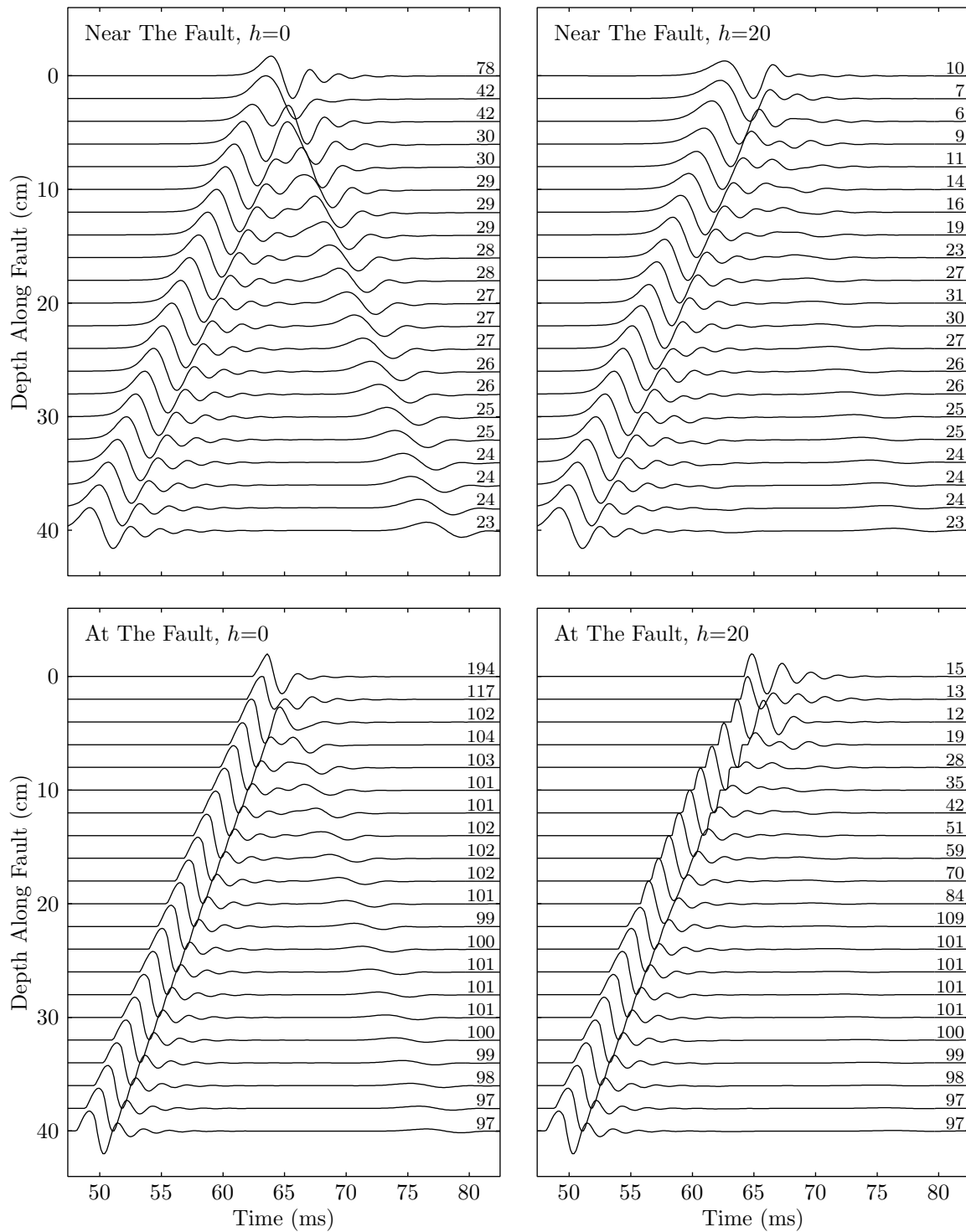


Figure 11: Comparison of acceleration records near the fault (top) with records at the fault (bottom). Peak acceleration (in units of g) is indicated by the number on each curve.

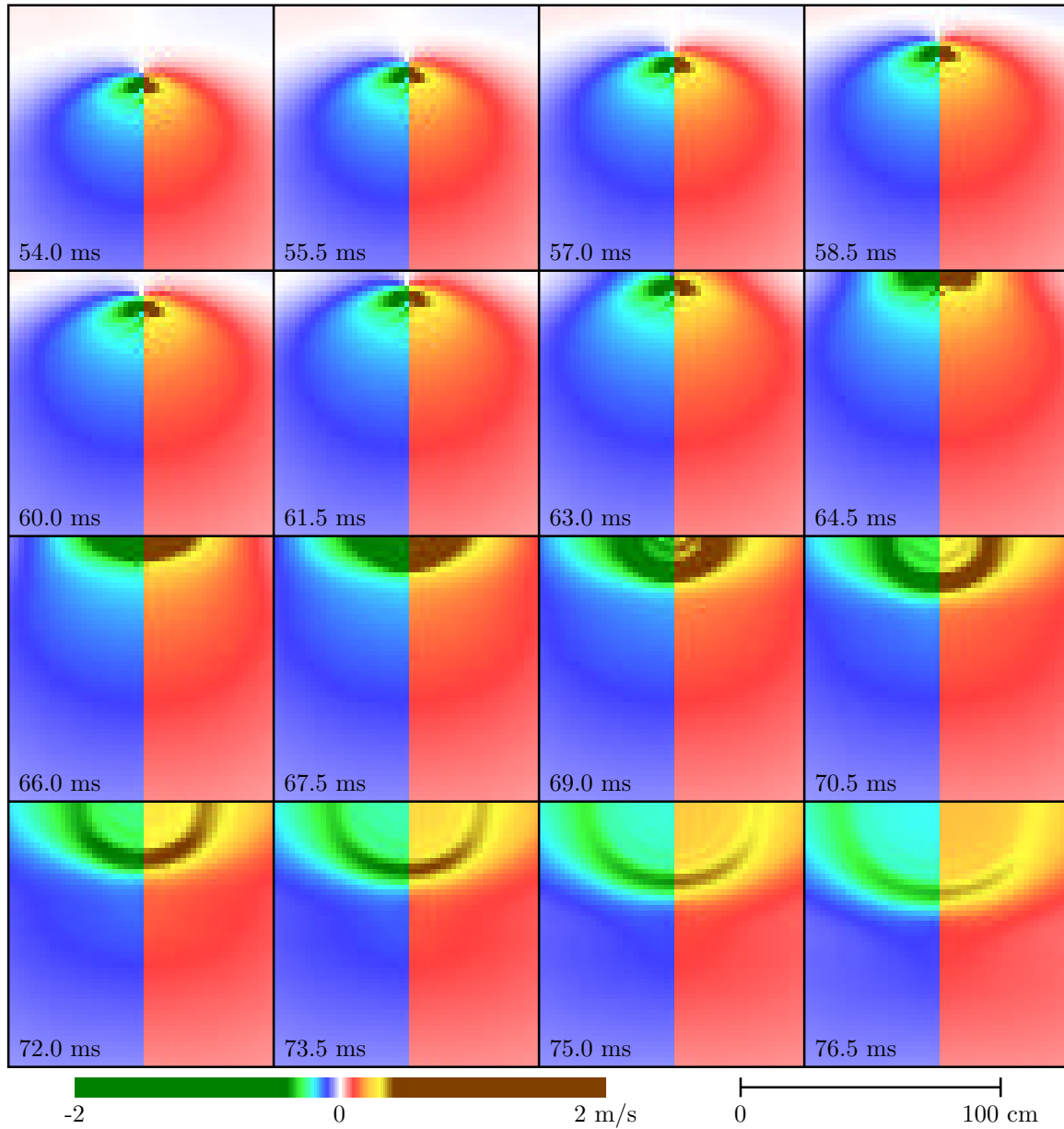


Figure 12: Cross-section snapshot images of along-strike particle velocity for the case with no weak zone ($h=0$). The fault bisects each pane and particle motion is in and out of the plane.

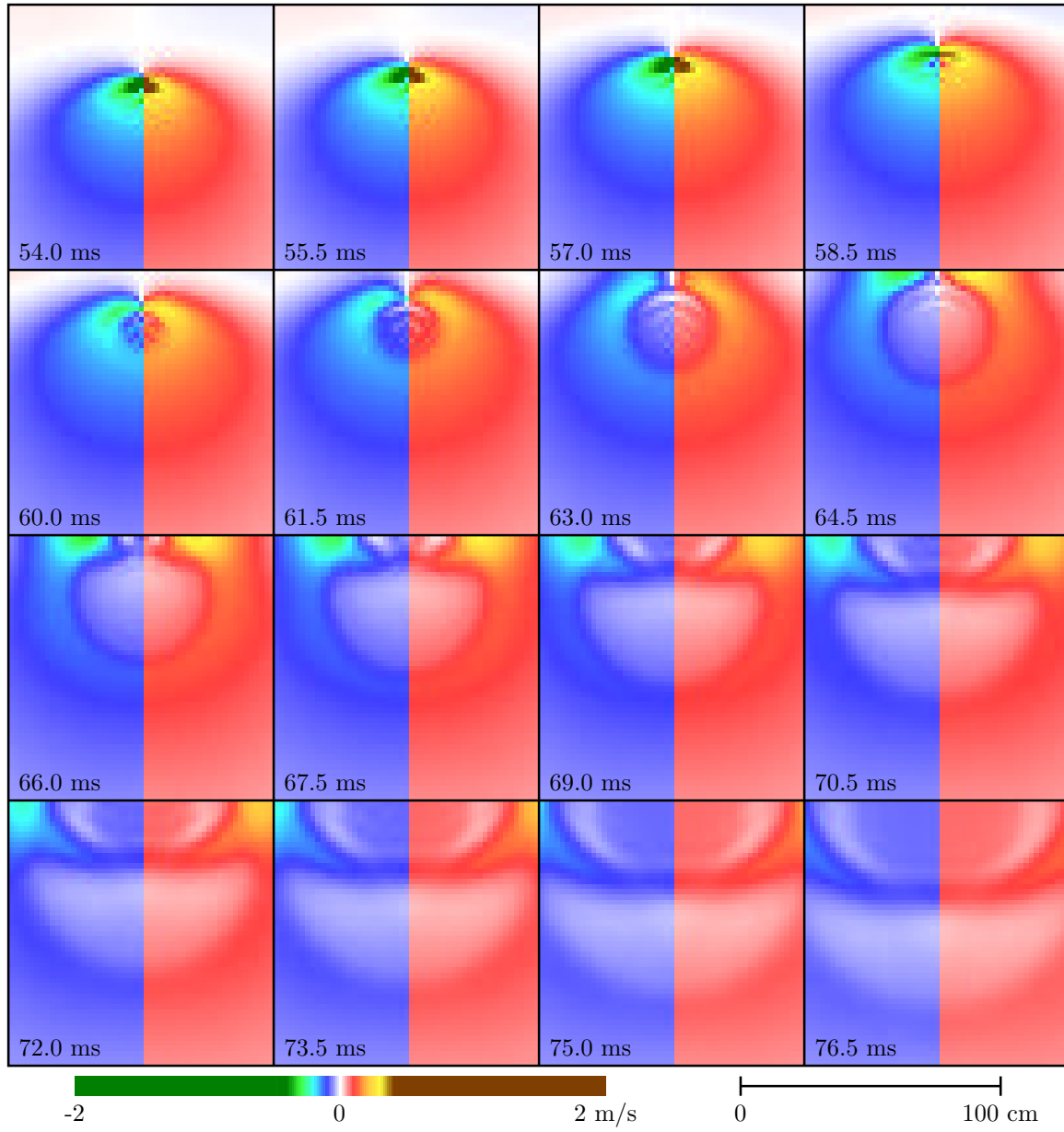


Figure 13: Cross-section snapshot images of along-strike particle velocity for the case with a 20 cm weak zone.

in the foam model with which to compare it. We can get some physical insight into the origin of this effect by examining the the cross-section snapshots of particle velocity in Figure 13. The snapshots reveal that when rupture enters the weak zone, a "stopping phase" propagates out from the fault. Beginning at 60 ms, just after the rupture has entered the weak zone, the stopping phase appears as a widening circle of low particle velocity. At 66 ms the stopping phase begins to reflect at the surface, and the intersection of the phase front with the surface propagates away from the fault. At the phase front is a relatively abrupt reduction in particle velocity that leads to the higher accelerations seen away from the fault in the presence of a weak zone.

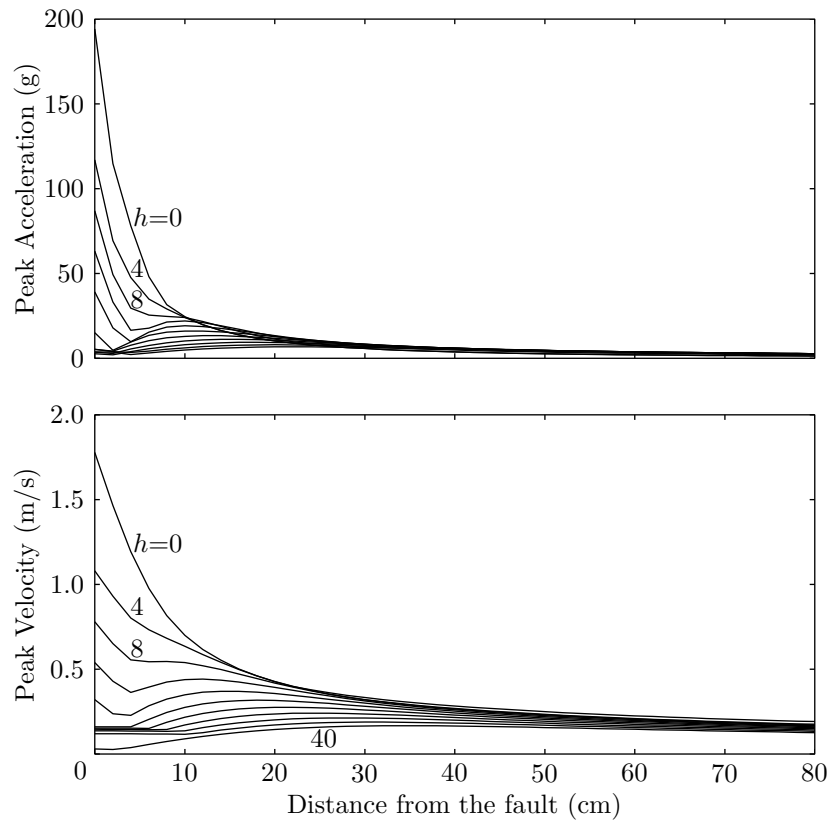


Figure 14: Peak acceleration (PA) and peak velocity (PV) at the surface. Different curves represent different weak zone widths (indicated by h) ranging from 0 to 40 cm.

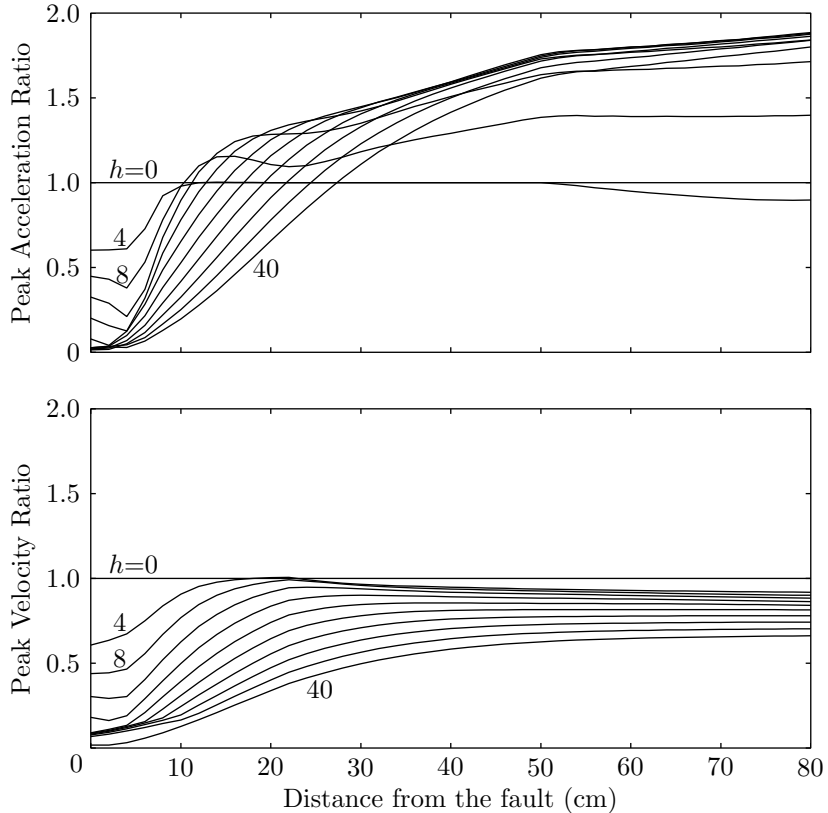


Figure 15: Ratio of PA (top) and PV (bottom) at the surface relative to the case with no weak layer. The width of the weak layer is correlated to the width of the zone in which motion PA is reduced. Outside the zone of reduction, PA is increased by the presence of the weak layer.

Sensitivity Study

In constructing the numerical model, there are two unconstrained or poorly constrained parameters: the coefficient of static friction in the strong part of the fault μ_s , and the coefficient of friction in the weak zone μ_w . Because the coefficients are not well known, it is important to understand how sensitive the results of the modeling are to variations in the coefficients. To quantify the sensitivity, two supplemental sets of numerical models were run, one with a different μ_s , and one with a different μ_w .

For the μ_s sensitivity test, μ_s was lowered from 2.4 to 2.3. This change cuts in half the difference between the prestress and failure stress $|\tau_u - \tau_0|$. Figure 16 shows acceleration records from the modified model. In comparison to the accelerograms from

the original model in Figure 8, the change brings larger peak accelerations and higher rupture velocity, although the effect is mild. The effect on the surface slip shown in Figure 17 is also quite small.

The coefficient of friction in the weak zone μ_w was measured by Brune and Anooshehpour (1998), but it was based on a lengthy extrapolation of measured data. Figure 5 shows this extrapolation. Also, in the numerical model, the coefficient of friction is constant, so at best it will be a compromise: too high when the slip velocity is low and too low when the slip velocity is high (assuming velocity strengthening holds true as the extrapolation suggests). These considerations suggest that it may be possible to find a coefficient of friction that effectively models the foam rubber data better than our

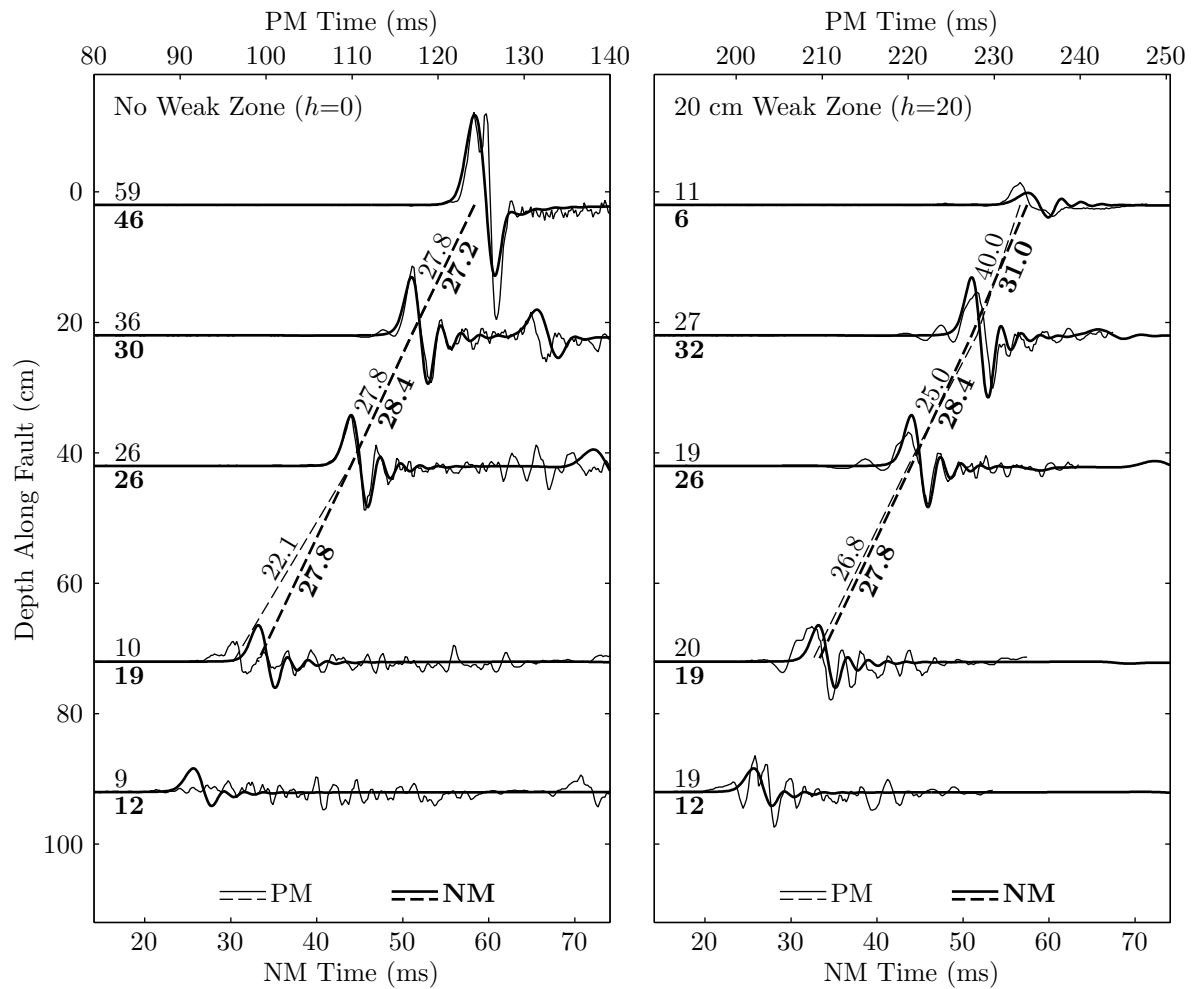


Figure 16: Comparison of acceleration records for the physical model (thin curves) and the numerical with modified μ_s (thick curves).

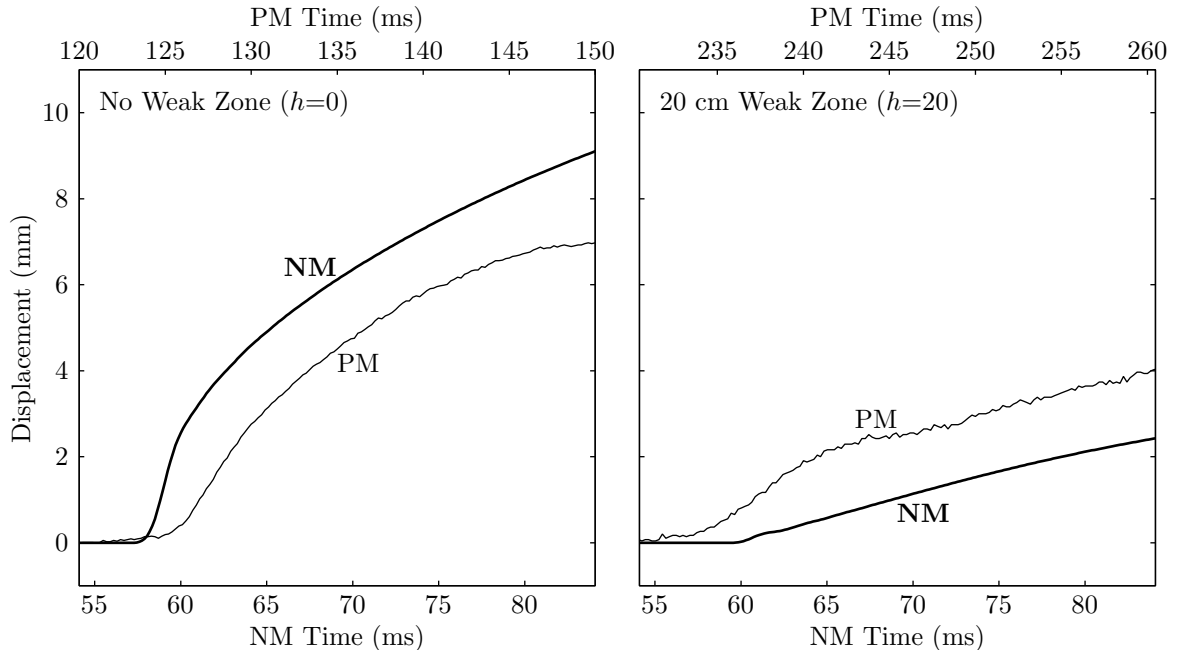


Figure 17: Comparison of displacement at the surface for the physical model (thin curves) and the numerical model with modified μ_s (thick curves).

initial estimation. As shown in Figure 10 for $h=20$, the numerical model significantly underestimates the amount of slip on the fault. By changing the coefficient of friction in the weak zone from 0.6 to 0.4, a better fit to slip and normalized surface PA was attained. Figure 18 shows much better agreement in surface slip between the numerical and physical models for the $h=20$ case. Figure 19 shows acceleration records from the modified model. Since the modification of μ_w only affects the weak zone, the accelerograms are nearly identical to those from the original model shown in Figure 8 except for the two near-surface records in the $h=20$ case. The lower μ_w increases PA at the near-surface trace and lowers the apparent rupture velocity in the weak zone. Also, the reflected phase in the second trace down is more prominent than in the original model, contrary to the physical model observations.

The origin of these effects can be seen in cross-section snapshot of the model (Figure 20). At 61.5 ms, the rupture has not been drastically reduced as it was in the original model (Figure 13), and remains strong enough to reflect off the free-surface (64.5 ms) and rupture back down the fault. Table 2 summarizes PA at the near-surface sensor

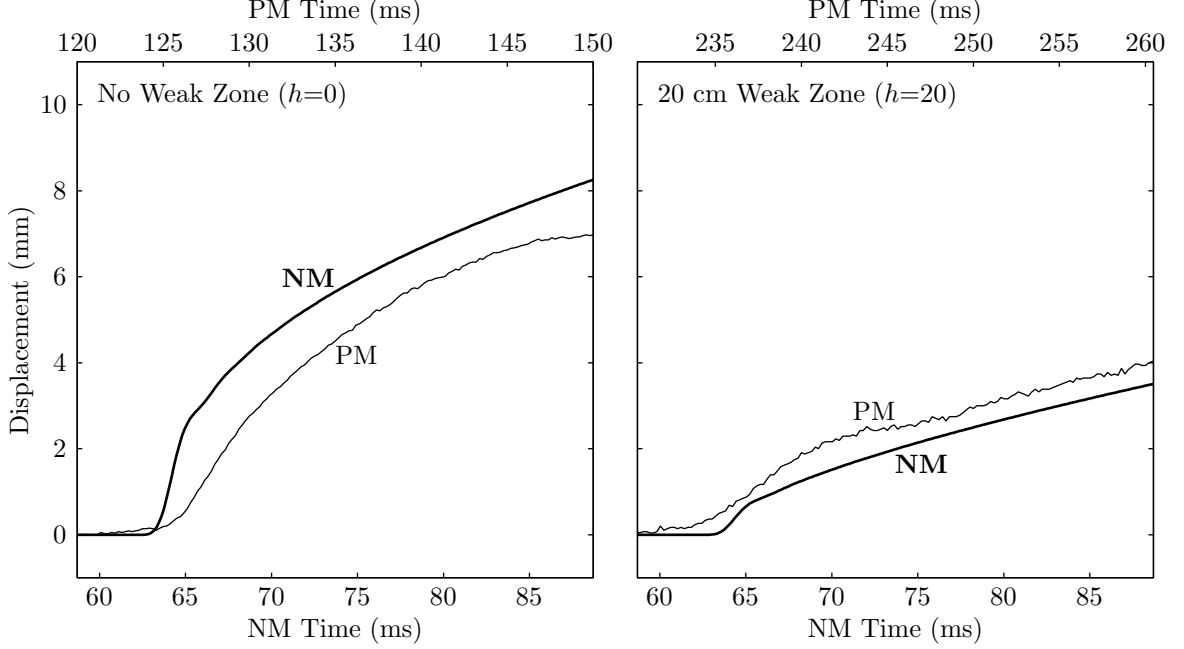


Figure 18: Comparison of displacement at the surface for the physical model (thin curves) numerical model with modified weak zone friction μ_w (thick curves).

for the physical model and the two numerical model cases when the coefficient of friction μ_w equals 0.6 and 0.4. When h differs between the two models, the numerical model h is indicated by parentheses. Normalized PA is consistently lower for the $\mu_w=0.6$ case than for the physical model. The $\mu_w=0.4$ case shows better agreement with the physical model, with the shallower weak zones underestimating PA and the deeper weak zones overestimating PA.

We conclude that the ground motion at the surface is relatively insensitive to the static friction of the foam μ_s , but is significantly sensitive to the weak zone frictional

Table 2: Normalized Surface PA

	Weak Zone Thickness, h				
	0	7.5(8)	15(16)	20	30
Physical Model	2.54	1.44	1.09	0.44	0.16
Numerical Model, $\mu_w=0.6$	1.91	0.95	0.39	0.31	0.15
Numerical Model, $\mu_w=0.4$	1.91	1.24	0.71	0.53	0.20

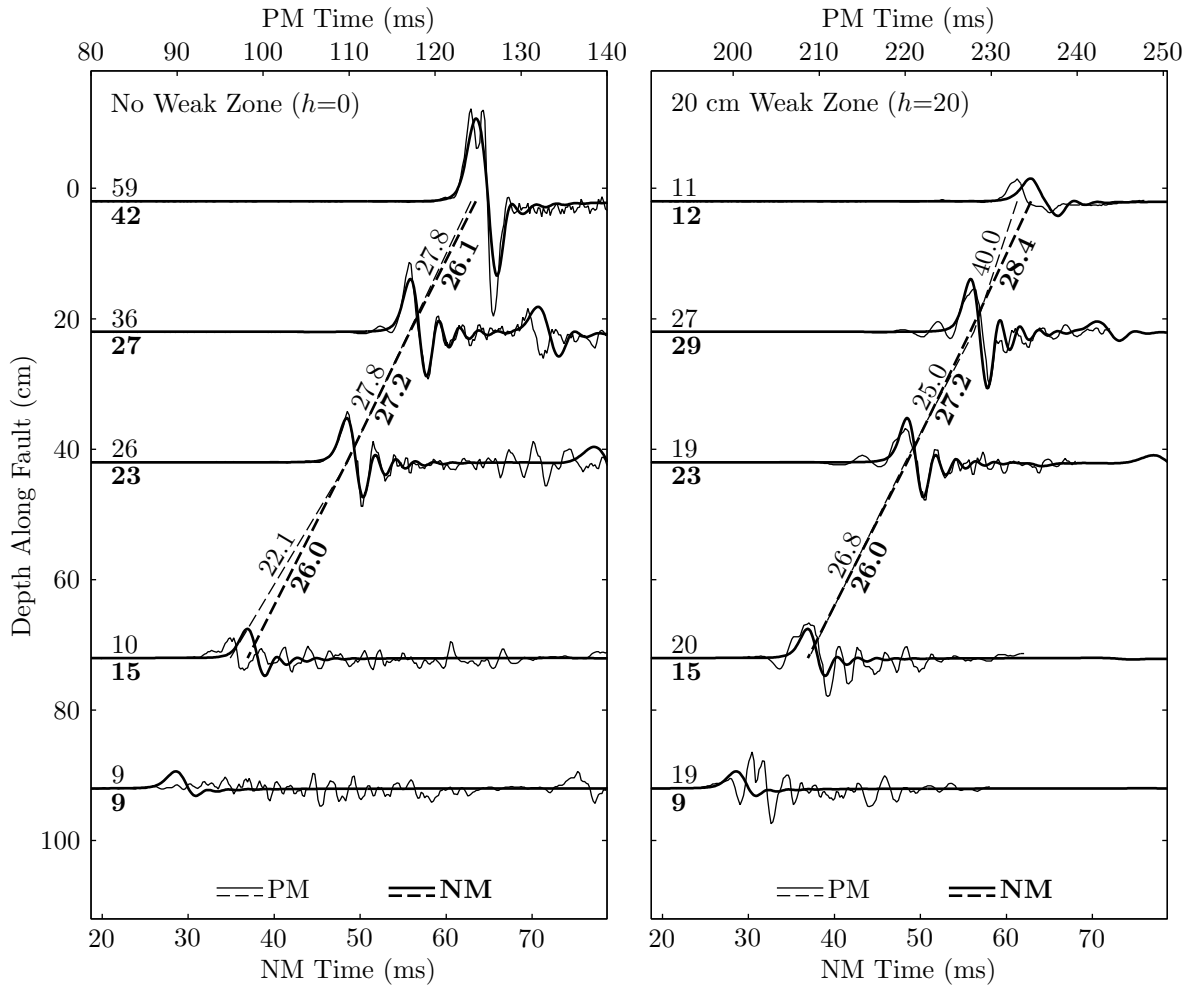


Figure 19: Comparison of acceleration records for the physical model (thin curves) and the numerical model with modified weak zone friction μ_w (thick curves).

behavior. In our modified model (with $\mu_w=0.4$) some results agree better with the physical model (surface PA) and other results agree worse (e.g., no suppression of reflected rupture), compared with the $\mu_w=0.6$ case. It is possible that these model behaviors would be brought into even better agreement by incorporating the observed velocity-strengthening into the numerical model weak zone friction law.

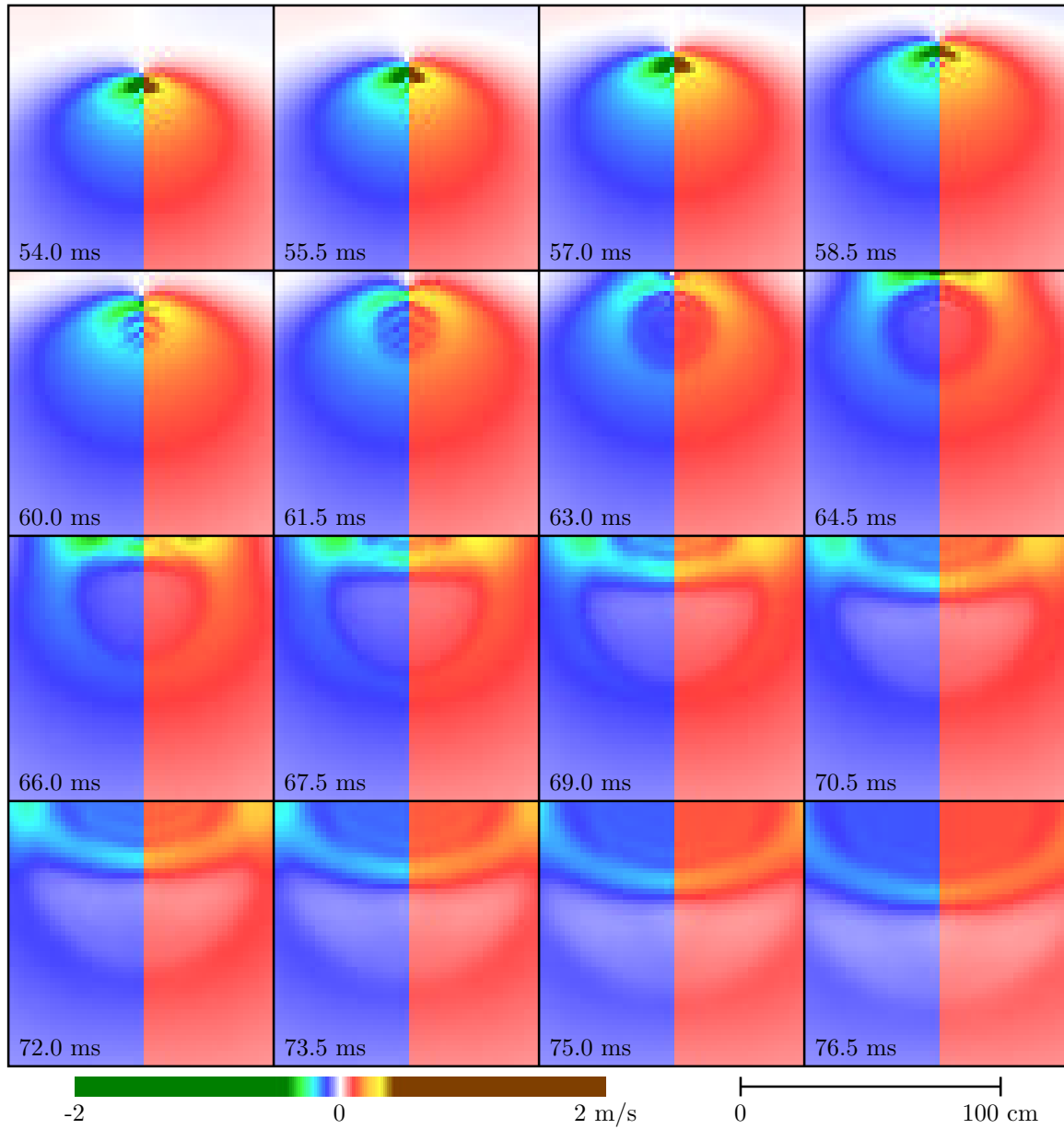


Figure 20: Cross-section snapshot images of along-strike particle velocity for the case with a 20 cm weak zone, and modified weak zone friction μ_w .

CHAPTER V

CONCLUSIONS

Numerical modeling of earthquakes is an important complement to other means of understanding source processes. Ground motion recordings are scarce for the near-source region of large earthquakes. Observations alone do not provide an adequate basis for characterizing near-source ground motions for use in earthquake engineering studies, and only the simplest theoretical models are solvable by analytical methods. Numerical modeling can help fill these gaps in understanding. We have improved our present capabilities for numerically modeling strong ground motion in three areas: parallel processing, model validation, and weak-zone fault dynamics.

By adding parallel processing, our computer code can now run efficiently on large multiprocessor supercomputers. A dynamic, 3D finite-difference rupture propagation code was parallelized using MPI, and benchmarked on 2 shared-memory multiprocessor systems. Benchmarks on a 96 processor computer yielded 87% of ideal speedup using 8 processors and 50% of ideal speed-up on 64 processors. The reward is that we can now run larger models that can incorporate a greater range of length scales leading to increased realism and usefulness.

In order to establish the validity of the numerical model formulation and implementation, we reproduced the results of foam rubber scale model experiments by Brune and Anooshehpour (1998). These experiments provide a basis for validation of numerical models which complements ground motion observations from real earthquakes. The advantages gained by using laboratory experiments include the availability of (1) acceleration histories recorded at depth, directly adjacent to the fault surface, (2) measurements of the regional stresses acting on the model, and (3) direct measurements of the bulk properties of the medium and the frictional properties of the fault surfaces. For foam rubber models with and without a weak zone, we successfully reproduce the shape and duration of the acceleration pulses recorded adjacent to the fault surface. Observed rupture veloc-

ities are also reproduced accurately in the numerical model, typically within about 10% (with maximum discrepancies in local rupture velocity of 25%). Absolute amplitudes of the acceleration pulses are typically matched within 40% (with maximum discrepancy of about a factor of 2). Surface slip across the fault after the passage of the main rupture pulse is matched to within about 20% when there is no weak zone. In the presence of the weak zone, the slip is quite sensitive to the assumed friction coefficient of the weak zone μ_w . When μ_w is adjusted within limits consistent with the slip-velocity dependent laboratory measurements, the numerical model slip (relative to the corresponding foam rubber experiment) ranges from a factor of 2 underprediction (for high friction coefficient 0.6) to less than 10% underprediction (for lower friction coefficient 0.4). Other common results are increased suppression of ground-surface peak acceleration with weak zone depth, and suppression of the reflected slip acceleration pulse by the weak zone.

The near-source region is subject to the strongest shaking during an earthquake, and consequently is of great concern to civil engineers and seismic hazard analysts. It is also the region most influenced by the shallow part of the fault. So, to be useful, models must properly handle the shallow part, where the dynamics generally differ from the rest of the fault. The combination of the foam rubber model of Brune and Anooshehpour (1998) and our numerical model aid in our understanding of the effects of making the shallow part weak and velocity-strengthening. The most notable observations from this study are, (1) the lateral distance from the fault in which peak acceleration is reduced scales up with weak zone thickness, (2) the abrupt drop in rupture velocity upon entering the weak zone leads to enhanced acceleration at some distance from the fault, even though peak acceleration decreases very near the fault, (3) accelerations recorded near, but not directly on, the fault may misrepresent the behavior of the actual rupture, (4) the effects at the surface are quite sensitive to the frictional properties of the weak zone. Inclusion of a weak zone is an important step toward realism for earthquake models. We plan in future work to move beyond these test cases and incorporate a weak zone in our models for real earthquakes.

APPENDIX

CODE USAGE AND INPUT FILES

Running The Code

The code is called is called 'DFM', which stands for 'Dynamic Fault Model'. It is a Fortran code that runs under UNIX like operating systems.

Requirements

If you will be using parallel processing, MPI must be installed on the system. Many vendors provide their own implementation of MPI. A good alternative, or for systems without MPI is a free version of MPI called MPICH available from <http://www-unix.mcs.anl.gov/mpi/mpich/>. You do not need to be the system administrator to install MPICH. MPICH may be installed on single processor machines for debugging DFM.

Installation

```
gunzip dfm.tar.gz
tar xvf dfm.tar
cd src
make dfm
```

This will make a shell script called `dfm`. This script must be in your `PATH`.

Usage

When executed `dfm` looks in the current directory for a file called `dfmin` (shown below) which contains the model input parameters. Depending on how many processors are called for in `dfmin`, `dfm` compiles either the serial or the parallel version of `dfm` (called `dfm_ser` and `dfm_par`), and runs the code. A parallel run creates separate files for each processor that are appended with the letter `p`. After a parallel run finishes, `dfm` calls a supplementary program called `unite` which pieces the separate files together. As a safeguard `unite` does not erase the separate files, and if everything looks good you can erase these files: `/bin/rm *p`. Another supplementary program for creating snapshot images call `snap` is supplied with DFM, but is not described here.

Input Files

The following page shows the scripts used to run the first model describe in this paper. The script performs multiple runs of the code while varying the weak zone thickness (controlled by the `(slip zone)` parameter). The two subsequent models were created by modifying the `(slip type)` parameter.

```

#!/bin/sh
# For a log file use: ./foam.sh >& log &
nk='2'; nk2='1'; load='2 1 2'; rupvy='1e5' # Use this line for 2D
nk='221'; nk2='111'; load='1 1 1'; rupvy='15.' # Use this line for 3D
runs='00 11 16 05 09 03 07 13 15 17 19 21' # Weak zone thicknesses
for runno in $runs
do
cat << END > dfmin
# foam rubber 3D strike-slip
# (material type) = vp vs rho betha yeild
# (slip type) = kind smin smax dzero gax gay gaz cohes
    npes3d(3) buffsz = 3 1 1 1
    maxg(3) ncycle2 = 141 $nk 221 999
    unifzn(6) = 1 141 1 $nk 1 221
    dx0 dy0 dz0 dt = .02 .02 .02 .00015
    x0 y0 z0 = 0. 0. 0.
    rj1 rj2 rk1 rk2 rl1 rl2 = 1. 1. 1. 1. 1. 1.
    velcut = 1e-30
    (material zone) = 1 141 1 $nk 1 221 1
    (material type) = 56. 30. 16. .5 1e12
    (load zone) = 1 141 1 $nk 1 221 1
    (load type) = $load
    (pload zone) = 1 141 1 $nk 1 221 1
    (pload) = 0.
    (slip plane) = 111
    (slip zone) = 1 141 1 $nk 111 1
    (slip zone) = 1 71 1 $nk 111 2
    (slip type) = 0 1e5 1e5 .001 0. 730. -330. 0.
    (slip type) = 0 1.85 2.4 .001 0. 730. -330. 0.
    (slip type) = 0 0.6 0.6 .001 0. 66. -330. 0.
    [jk]focus rcrit rupv[xy] = 71 $nk2 0.4 15. $rupvy
    (print zone) = 1 1 $nk2 $nk2 111 161 1 0 1
    (print zone) = 1 51 $nk2 $nk2 111 113 1 0 1
    outname = fm$runno
END
if [ $runno != "00" ]; then
    echo "(slip zone) = 1 $runno 1 $nk 111 3" >> dfmin
fi
if [ $runno = "11" -o $runno = "00" ]; then
    echo "(print zone) = 1 51 $nk2 $nk2 86 136 10 0 0" >> dfmin
fi
dfm
done

```

REFERENCES

- Fortran programming guide, FORTRAN 77 5.0 - Fortran 90 2.0*, Sun Microsystems, 1999.
- Allen, C. R., J. N. Brune, L. S. Cluff, and A. G. Barrows, Jr., Evidence for unusually strong near-field ground motion on the hanging wall of the san fernando fault during the 1971 earthquake, *Seism. Res. Lett.*, *69*, 524–531, 1998.
- Anderson, J. C., and V. V. Bertero, Uncertainties in establishing design earthquakes, *J. Structural Engineering*, *113*, 1709–1724, 1987.
- Anderson, J. G., and J. E. Luco, Parametric study of near-field ground motion for a strike-slip dislocation model, *Bull. Seism. Soc. Am.*, *73*, 23–43, 1983.
- Andrews, D. J., Rupture propagation with finite stress in antiplane strain, *J. Geophys. Res.*, *81*, 3575–3582, 1976.
- Anooshehpour, A., and J. N. Brune, Frictional heat generation and seismic radiation in a foam rubber model of earthquakes faulting, friction, and earthquake mechanics; part 1, *Pageoph*, *142*, 735–747, 1994.
- Archuleta, R. J., and S. H. Hartzell, Effects of fault finiteness on near-source ground motion, *Bull. Seism. Soc. Am.*, *71*, 939–957, 1981.
- Brune, J. N., and A. Anooshehpour, A physical model of the effect of a shallow weak layer on strong ground motion for strike-slip ruptures, *Bull. Seism. Soc. Am.*, *88*, 1070–1078, 1998.
- Brune, J. N., P. A. Johnson, and C. Slater, Nucleation, predictability, and rupture mechanism in foam rubber models of earthquakes, *J. Himalayan Geol.*, *1*, 155–166, 1990.
- Brune, J. N., S. Brown, and P. A. Johnson, Rupture mechanism and interface separation in foam rubber models of earthquakes; a possible solution to the heat flow paradox and the paradox of large overthrusts new horizons in strong motion; seismic studies and engineering practice, *Tectonophysics*, *218*, 59–67, 1993.
- Culler, D. E., J. P. Singh, and A. Gupta, *Parallel computer architecture: a hardware/software approach*, Morgan Kaufmann Publishers, San Francisco, 1999.
- Day, S. M., Three-dimensional finite-difference simulation of fault dynamics: rectangular faults with fixed rupture velocity, *Bull. Seism. Soc. Am.*, *72*, 705–727, 1982a.
- Day, S. M., Three-dimensional simulation of spontaneous rupture: the effect of nonuniform prestress, *Bull. Seism. Soc. Am.*, *72*, 1881–1902, 1982b.

- Day, S. M., G. Yu, and D. J. Wald, Dynamic stress changes during earthquake rupture, *Bull. Seism. Soc. Am.*, *88*, 512–522, 1998.
- Dieterich, J. H., Modeling of rock friction; 1, experimental results and constitutive, *J. Geophys. Res.*, *84*, 2161–2168, 1979.
- Dongarra, J. J., H. W. Meuer, and E. Strohmaier, Top500 supercomputer sites, *University of Tennessee Computer Science Technical Report UT-CS 00-442*, Knoxville, Tenn., 2000.
- Freund, L. B., *Dynamic fracture mechanics*, Cambridge Univ. Press, New York, 1990.
- Gropp, W., E. Lusk, and A. Skjellum, *Using MPI: portable parallel programming with the message-passing interface, second edition*, MIT Press, Cambridge, Mass., 1999.
- Hall, J. F., T. H. Heaton, M. W. Halling, and D. J. Wald, Near-source ground motion and its effects on flexible buildings, *Earthquake Spectra*, *11*, 569–605, 1995.
- Hanks, T. C., and H. Kanamori, A moment magnitude scale, *J. Geophys. Res.*, *84*, 2348–2350, 1979.
- Harris, R. A., and S. M. Day, Dynamic three-dimensional simulations of earthquakes on en echelon faults, *Geophys. Res. Lett.*, *26*, 2089–2092, 1999.
- Ida, Y., Cohesive force across the tip of a longitudinal-shear crack and griffith's specific surface energy, *J. Geophys. Res.*, *77*, 3796–3805, 1972.
- Ide, S., and M. Takeo, Determination of constitutive relations of fault slip based on seismic wave analysis, *J. Geophys. Res.*, *102*, 27,379–27,391, 1997.
- Magistrale, H., and S. M. Day, Three-dimensional simulations of multisegment thrust fault rupture, *Geophys. Res. Lett.*, *26*, 2093–2096, 1999.
- Marone, C., Laboratory-derived friction laws and their application to seismic faulting, *Annu. Rev. Earth Planet. Sci.*, *26*, 643–696, 1998.
- Marone, C., and C. H. Scholz, The depth of seismic faulting and the upper transition from stable to unstable slip regimes, *Geophys. Res. Lett.*, *15*, 621–624, 1988.
- Okubo, P. G., Dynamic rupture modeling with laboratory-derived constitutive relations, *J. Geophys. Res.*, *94*, 12,321–12,335, 1989.
- Olsen, K. B., R. J. Archuleta, and J. R. Matarese, Three-dimensional simulation of a magnitude 7.75 earthquake on the san andreas fault, *Science*, *270*, 1628–1632, 1995.
- Olsen, K. B., R. Madariaga, and R. J. Archuleta, Three-dimensional dynamic simulation of the 1992 landers earthquakes, *Science*, *278*, 834–838, 1997.
- Snir, M., S. Otto, S. Huss-Lederman, D. Walker, and J. Dongarra, *MPI: the complete reference: volume 1, the MPI core, second edition*, MIT Press, Cambridge, Mass., 1998.

Somerville, P. G., N. F. Smith, R. W. Graves, and N. A. Abrahamson, Modification of empirical strong ground motion attenuation relations to include the amplitude and duration effects of rupture directivity, *Seism. Res. Lett.*, 68, 199–222, 1997.

ABSTRACT

ABSTRACT

Near-fault earthquake ground motion can be strongly enhanced relative to more distant sites, due to both the proximity of the source and the presence of pronounced directivity effects. Reliable predictions of these motions are impeded by the scarcity of near-source recordings for large earthquakes. Numerical simulations can play a useful role in quantifying near-source effects for use in engineering studies. In order to provide credible near-source ground motion estimates, simulations will need to account for the effect of reduced strength on the dynamics of rupture and slip on the shallow part of faults. While the numerical methods underlying dynamic earthquake simulations have frequently been tested against analytic solutions to linear problems, few opportunities exist to validate the methods for appropriate nonlinear, frictional-interface problems. Researchers have constructed foam rubber scale models of earthquakes which include a weak zone in the shallow part of the fault, and these controlled experiments provide detailed, subsurface recordings of rupture propagation and fault motion unavailable for real earthquakes. I show here that 3D finite difference (FD) simulations can replicate the main dynamic features observed in the physical models. There is notable similarity between physical and numerical models in rupture velocity and slip acceleration time histories on the fault. Local rupture velocities typically agree within 10%, with maximum discrepancies of 25%. Peak accelerations typically match within 40%, with maximum discrepancies of approximately a factor of two. In both models, peak acceleration is greatly diminished near the fault when the weak zone is present (relative to motion in the absence of the weakened zone), and the reduction factor scales up with weak zone thickness.

Numerical simulations aid in the physical interpretation of the foam rubber experiments and can extend their parameter range. The results show that the lateral distance over which ground motion parameters are diminished increases with the depth of the weak zone. They further show that ground acceleration is sensitive to the strength of the weak zone, with decreased weak-zone strength leading to stronger ground motion.

The ability of the numerical simulations to mimic key behaviors of the physical model provides a good starting point for simulation-based interpretation of earthquake ground motion recordings. Additionally, through parallelization of the computer code and the use of multiprocessor supercomputers, models can now be larger than previously possible. Larger FD models can accommodate a greater dynamic range of wavelengths, leading to better approximations of the underlying physics, and increasing the range of practical applications.