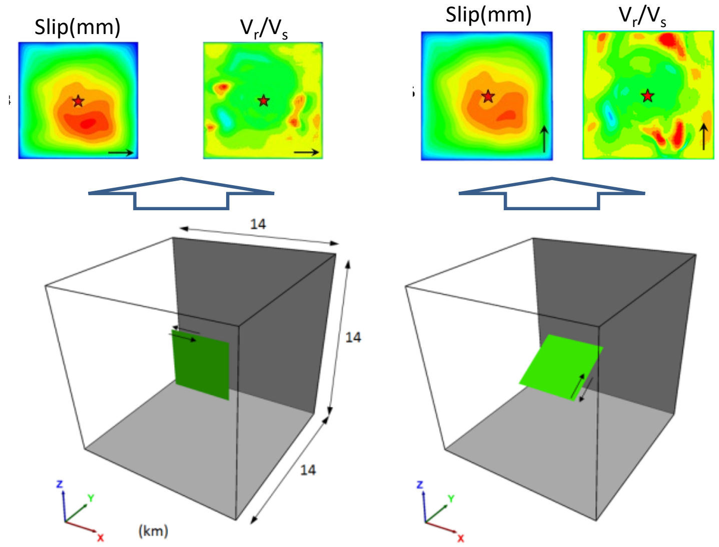
DocuSign Envelope ID: F90AD6A6-9F1D-411F-82E4-0B573C18F616

RESEARCH REPORT

VTT-R-00102-23



(*) Modelled strike-slip and normal fault-ruptures. Slip and rupture velocity based on Jussila *et al* (2021), https://doi.org/10.1785/0120210081

Benchmarking and evolving earthquake fault-rupture simulations

Authors: Ludovic Fülöp[1], Vilho Jussila[2], Outi Kaisko[3]

[1]VTT Technical Research Centre of Finland, [2]Seisneering, [3]Rock Mechanics Consultants Finland Oy

Confidentiality: VTT Public

Version: 31.1.2023

beyond the obvious

Report's title Benchmarking and evolving earthquake fault-rupture simulations	
Customer, contact person, address KYT, Finnish Research Programme on Nuclear Waste Management 2022	**Order reference** VN/14858/2021
Project name Benchmarking and evolving earthquake fault-rupture simulations (BEEFS)	**Project number/Short name** 131791/ KYT2022_BEEFS_2022
Author(s) Ludovic Fülöp, Vilho Jussila, Outi Kaisko	**Pages** 48/
Keywords earthquake rupture modelling, simulation, 3DEC, FLAC3D	**Report identification code** VTT-R-00102-23

Summary

Finland is located on the seismically quiet Fennoscandian shield with no modern-time observations of large earthquakes. The low level of seismicity has forced the nuclear facilities to conduct seismic hazard analyses with an insufficient amount or completely lacking relevant observations of surface or underground ground motions caused by earthquakes. The Finnish underground repositories suffer from even lesser data available as no observations on underground ground movements caused by earthquakes exist and therefore the analyses of underground earthquake hazard have been built solely on simulations (e.g., *Hökmark et al.*, 2019).

With recent advances in earthquake fault-rupture simulations, advanced software applications became available, and many publications were dedicated to simulating the rupture of earthquake faults (e.g., a review in IAEA, 2018). This development also led to large benchmarking exercises being conducted to compare the fault-rupture predictions of different software, and to compare these predictions with the limited observations available.

In this work more agile simulation methodologies are proposed, and benchmarking based on the existing models is carried out. The existing methodology is quite laborious and computationally expensive. Benchmarking of suitable FEM-DEM/DEM code is given to large benchmarking exercises (e.g., published by Harris et al, 2018); confirming that the code is simulating the rupture of earthquake faults similarly to codes developed specifically for such applications.

Confidentiality	VTT Public

Espoo, 5.4.2023

On behalf of the authors	**Reviewed by**
Ludovic Fülöp	Naumer Sami
Principal Scientist	Research Scientist

VTT's contact address

VTT Technical Research Centre of Finland Ltd., P.O Box 1000, FI-02044 VTT, Finland

Distribution (customer and VTT)

Customer	1 coy
VTT/Archive	1 copy

The use of the name of "VTT" in advertising or publishing of a part of this report is only permissible with written authorisation from VTT Technical Research Centre of Finland Ltd.

Approval

VTT TECHNICAL RESEARCH CENTRE OF FINLAND LTD

Date:	06.04.2023
Signature:	DocuSigned by: Marko Mäkipää EF95EB85E9804AC...
Name:	Marko Mäkipää
Title:	Research Team Leader

Preface

This work is the result of the project "*Benchmarking and evolving earthquake fault-rupture simulations (BEEFS)*" carried out in the year 2022.

Seismic hazard assessment in Finland is carried out with very limited observational data. One way to augment this data pool is to use numerical modelling to generate inputs for the hazard assessment. Earlier work targeted the development of numerical modelling methods to generate ground motions relevant for nuclear power plant safety (Jussila et al., 2021), and underground fracture movements, relevant for the safety of the final-disposal facility (Fälth et al., 2015).

In this work we developed more agile simulation methodologies and compared them against internationally recognized benchmarks (e.g., Harris et al, 2018). Agility, speed, and flexibility of the modelling is important to generate statistically significant results, incorporating some uncertainties. The intention is also to develop a single workflow for ground-motion modelling and movements of the underground fracture, enabling the optimal use of the modelling capabilities.

The results can be applied for generating relevant ground motions for improving the probabilistic seismic assessment of NPPs. They can also be used in strengthening the safety case for the underground repository in supervision by the authorities, to further strengthen the transparency, repeatability, and reliability of the existing conceptual models for induced secondary movements of fractures. The research outputs provide capabilities for future assessment of the long-term safety of the ONKALO facility and is aligned with the KYT Framework Programme 2019-2022 (TEM 2018), which is focused on the safety, feasibility, and acceptability of nuclear waste management.

This was an ambitious skill development project. We wish to thank the kind help of Billy Fälth, from Claytech Oy, for assisting with the understanding of some backgrounds used with 3DEC for calculating secondary fracture movements. The contributions of Jim Hazzard from ITASCA providing the 3DEC TPV5 material, and Matt Purvance from ITASCA, developing the novel FLAC3D/PFC zone joint logic allowing this simulation before the commercial launching of the method were crucially important. The support of Lukas Krenz, from the Technical University of Münich, to provide support with the deployment of SeisSol is also acknowledged.

Espoo, 5.4.2023

Authors

Contents

Preface 3
1. Introduction 5
2. Goal 5
3. Limitations 6
4. Methods 6
  4.1 Available benchmarking cases 6
  4.2 Description of case TPV5* and monitored outputs 10
    4.2.1 Output monitoring points 11
5. Protocols and software for simulating dynamic rupture 16
  5.1 General purpose rock mechanics simulation software, 3DEC, PFC3D and FLAC3D 16
  5.2 Fault rupture simulation with SeisSol 19
    5.2.1 Modelling TPV5 with SeisSol 20
    5.2.2 Running the model 22
    5.2.3 Rupture of a primary-fault initiating rupture in a secondary-fault 23
    5.2.4 Can secondary-fault rupture be modelled with SeisSol? 23
6. Results of the example fault-rupture simulation 25
  6.1 Results with FLAC3D zone joint logic 25
  6.2 Results with SeisSol 29
7. Discussion and conclusions 34
References 37
Appendix A 39
  Installing SeisSol 39
  Requirements for computer 39
  Installation prerequisites 40
  Installing virtual environment 40
  Should problems occur 46
  Glossary 47

1. Introduction

Finland is located on the seismically quiet Fennoscandian shield, with no modern-time observations of large earthquakes (M>5…5.5) and moderate earthquakes (M>4) occurring only rarely. The low level of seismicity has forced the nuclear facilities to conduct seismic hazard analyses with an insufficient, or completely lacking, observations of surface or underground ground-motions caused by significant earthquakes.

The seismic hazard analyses of nuclear power plants (NPPs) have either relied on ground motion prediction equations derived using data from corresponding intraplate regions (Varpasuo et al., 2001) or on equations using such observations to supplement the Fennoscandian data on the large magnitude end of the prediction equations (e.g., Fülöp et al., 2019).

The Fennoscandian underground repositories suffer from even lesser data available, as no observations of underground ground movements caused by earthquakes exist. Therefore, the analyses of underground earthquake hazard have been built on simulation alone (e.g., Hökmark et al., 2019). The underground seismic hazard of the spent nuclear fuel repository used in Finland and Sweden can be constrained to the so-called secondary movements on fractures induced by earthquakes taking place at large fault zones. Fracture movements induced by earthquake may damage the deposition canisters, with failure criterion being in the range of 50 mm shear-slips (SKB, 2011), while calculation models suggest slips in the range of tens of millimetres (POSIVA, 2019).

The assessment of such hazard rests on extensive modelling of earthquake effects on fractures using the Itasca 3DEC software (e.g., Fälth and Hökmark, 2011, Fälth et al., 2015, Hökmark et al., 2019). In these studies, 3DEC was used to model the rupture of the fault of a primary earthquake, the consequent propagation of the stress waves near the primary fault, and the potential of these stress waves to induce secondary movements on nearby fractures. The 3DEC models were based on sound rock mechanics principles, relying on the up-to-date understanding of the stress state in the Earth's crust, and assumed evolution of those stresses in the future.

With recent advances in earthquake fault-rupture simulations, advanced software applications became available, and many publications were dedicated to simulating the rupture of earthquake faults (e.g., a review in IAEA, 2018). This development also led to benchmarking exercises being conducted to compare the fault-rupture predictions of different software, and to compare these predictions with the limited observations available. Code verification for the dynamic problem of fault-rupture has a few established community dynamic-rupture benchmarks (e.g., Harris et al., 2009, Harris et al., 2018).

The available benchmark exercises provide a valuable tool to demonstrate the performance capability of multi-purpose codes (e.g., 3DEC, FLAC3D or PFC3D) against codes specifically developed for simulating fault rupture processes, like SeisSol. In addition, benchmarking of the DEM or FEM-DEM code such as FLAC3D, 3DEC or PFC3D complement their verification in fault rupture applications.

2. Goal

The general aim of the work here was to improve code verification, with this recently introduced higher international standard, extend expertise in Finland and update simulation techniques with more versatility and accessibility for future simulation demands of the ground-motions caused by earthquakes.

The work summarized here had the following objectives:

- To develop a more agile FEM-DEM/DEM simulation methodology and benchmarking based on the existing models. The existing methodology is quite laborious and computationally expensive; it would be of great importance to identify a less complex modeling path to facilitate parametric modeling.

- This implies identifying the modeling needs, i.e., what are the core features and results that the modeling must be able to provide. Based on the needs, the possible modeling tools are to be shortlisted.
- The suitable FEM-DEM/DEM software codes have to be benchmarked against one of the large benchmarking exercises (e.g., published by Harris et al., 2018); confirming that the code is simulating the rupture of earthquake faults similarly to codes developed specifically for such applications.

Benchmarking increases the reliability of seismic hazard analyses, built on the simulations performed with the code and empower the use of the code in corresponding applications, like studying the effects of seismicity on infrastructure. In addition, dynamic earthquake simulations provide a tool for studying the near-field effects of earthquakes possibly occurring close to NPPs.

As a result, the Finnish simulation expertise is extended from static applications to dynamic simulations, which is a key area of rock mechanics. FLAC3D, 3DEC and PFC3D are routinely used for static stress simulations at different scales, stability analyses and thermomechanical simulations in Finland. The acquired knowledge for earthquake simulations is useful for studying seismic hazards of NPPS and nuclear repositories, as well as in increasing safety and social acceptance in projects including blasting or induced seismicity close to population, infrastructure, or vulnerable constructions.

3. Limitations

While a broad range of modelling tools and methods were explored, the reach of this work was necessarily limited by the number of experts in the project. Luckily, international outreach to ITASCA and TUM helped streamline some of the modelling efforts. The other limitation of the project is that we are only comparing models with models, the enriching of observational dataset, if at all possible, was not in the scope of this work.

4. Methods

4.1 Available benchmarking cases

In previous studies, an attempt was made for modelling strike-slip and dip-slip faults in 3DEC and derive generated ground motions (GM) in relation to nuclear power plant (NPP) safety (Fülöp et al., 2016; Fülöp et al., 2017). These faulting types were shown to generate ground motion patterns conforming to the ones expected from theoretical solutions. The modelled moment magnitude was M_w5.5, with a fault size of 5×5 km and a stress drop of 5 MPa. Due to the imposed symmetry of the stress field and rupture, the generated ground-motions had only academic value. In the next stage oblique fault-ruptures were modelled in 3DEC, based on a predefined stress field (Fülöp et al., 2019; Jussila et al., 2021). The range of moment magnitudes was M_w4 to M_w5.5. The fault sizes 1×1 km to 3.5×3.5 km and the stress drops of 10 MPa and 50 MPa were considered. These GMs could be considered for practical use. The median ground motions from the models compared reasonably with ground-motion prediction equations and observed randomness also compared reasonably, though it was somewhat lower than, the randomness of the regional ground motion prediction equation (Fülöp et al., 2020). This proved that the diversity of modelled cases in the study were significant.

In this study, the main resource for identifying suitable benchmarking cases has been Harris et al. (2009), with the comprehensive list of benchmark cases given in Table 1. The selection criteria for Fennoscandia should be based on the geological conditions, seismicity, and targeted hazard outputs.

Table 1. List of available benchmark cases from Harris et al. (2009).

Case name	Description
TPV3 / TPV4 / TPV5*	Spontaneous rupture on a vertical strike-slip fault in a homogeneous full-space. With or without heterogeneities.
TPV6 / TPV7	Spontaneous rupture on a vertical strike-slip fault in a bimaterial half-space, with high shear modulus contrast across the fault (a "well-posed" problem).
TPV8 / TPV9	Spontaneous rupture on a vertical strike-slip fault in a homogeneous half space. Initial stress linearly dependent on depth.
TPV10 / TPV11 / TPV12* / TPV13 *	Spontaneous rupture on a 60-degree dipping dip-slip fault (normal fault) in a homogeneous half-space. Initial stress linearly dependent on depth.
TPV14 / TPV15	Spontaneous rupture on a right-lateral, vertical, strike-slip fault with a rightward branch forming a 30-degree angle. Linear elastic properties in a homogeneous half-space. Material properties and initial stresses are like TPV5.
TPV16 / TPV17	Spontaneous rupture on a vertical strike-slip fault in a homogeneous half-space. There are randomly generated heterogeneous initial stress conditions.
TPV18 / TPV19 / TPV20 / TPV21	Spontaneous rupture on a right-lateral, vertical, strike-slip fault with a rightward branch forming a 30-degree angle. There are linear elastic material properties or Drucker-Prager plastic material properties in a homogeneous half-space.
TPV22 / TPV23	Spontaneous rupture on a right-lateral, vertical, strike-slip fault with a stepover (extensional or compressional step). Linear elastic material.
TPV24 / TPV25	Spontaneous rupture on a right-lateral, vertical, strike-slip fault with a rightward branch forming a 30-degree angle. There are linear elastic material properties in a homogeneous half-space.
TPV26 / TPV26v2	Spontaneous rupture on a vertical strike-slip fault. Linear elastic properties in a homogeneous half-space. Linear elastic version of TPV27.
TPV27 / TPV27v2	Spontaneous rupture on a vertical strike-slip fault. There are Drucker-Prager viscoplastic material properties in a homogeneous half-space.
TPV28	Spontaneous rupture on a non-planar vertical strike-slip fault. The fault is a vertical plane, except for two hills. Initial shear and normal stresses on the fault are obtained by resolving a regional stress tensor onto the fault surface.
TPV29 / TPV30	Spontaneous rupture on vertical strike-slip fault with stochastic roughness. Linear elastic material properties in a homogeneous half-space. Shear and normal stresses obtained by resolving a regional stress tensor onto the fault surface. TPV29 is linear elastic and TPV30 with Drucker-Prager viscoplastic.
TPV31 / TPV32	Spontaneous rupture on vertical strike-slip fault. Discontinuous 1D velocity structure in linear elastic half-space. Stress proportional to shear modulus.
TPV33	Spontaneous rupture on a vertical strike-slip fault. 3D velocity structure with low-velocity fault zone in linear elastic half-space. Shear stresses tapered so that rupture stops spontaneously at the earth's surface or edge of the fault.
TPV34	Imperial Fault.
TPV35	Parkfield 2004 M6 Earthquake.
TPV101 / TPV102 / TPV103 / PV104	Spontaneous rupture on a vertical strike-slip fault in a homogeneous halfspace. Rate-state friction, using an ageing law or slip law with strong rate-weakening.
TPV105 / TPV105-3D	Spontaneous rupture on a strike-slip fault. Thermal pressurization, with rate-state friction, slip law with strong rate-weakening. 2D and 3D benchmark.
TPV205 *	Same problem as TPV5, except performed at multiple resolutions.
TPV210	Same problem as TPV10, except performed at multiple resolutions.

The magnitude values in the earlier studies were chosen to correspond to high contributors of hazard for nuclear power plants. In different probabilistic seismic hazard analysis studies, the assumed largest magnitude ranges were between M_w5.5…7.7 (e.g. Fülöp et al, 2022). Post-glacial faults (PGF) indicate the occurrence of M_w6.5…8.2, following the deglaciation (e.g. Ojala et al, 2019).

The median value of the maximum magnitude is between M_w6.2…6.5, with a considerable uncertainty around this median. Given the design lifetime of the underground repository, the modelled moment magnitudes should be in the range of the largest magnitudes from NPP studies.

With regards to the benchmark cases in Table 1, the following selection criteria were applied:

- Strike-slip or gently dipping faults are relevant for Finnish NPPs and the repository (Jussila et al, 2021). There is no need to use an oblique case for the benchmarking, but the developed modelling method should be capable to include oblique faults.
- No need to use sophisticated initial stress distribution (e.g., random heterogeneities) for the benchmarking, but the modelling should be able to accommodate it later.
- A case with very-hard rock condition (i.e., at least V_s>2000 m/s) should be chosen.
- Moment magnitude of the earthquake should be in the upper range of maximum magnitude for NPPs. E.g., M_w6…7, would make the case relevant for both NPPs and the repository. The approximate fault area could be estimated using Leonard (2014), at between 100–1000 km^2, for the magnitudes listed.
- The benchmark case must have sufficiently detailed description, both inputs and results. A case reporting underground movements would be optimal, but not likely available.

The characteristics of the different benchmark cases by Harris et al. (2009) are listed in Table 2. The relevance to Finnish conditions can be ascertained by comparing these characteristics to the targets described.

It can be noticed that there are many strike-slip cases, while there are very few normal faulting cases, and no reverse faulting case. Strike-slip and reverse-faulting can both be relevant, with a gently dipping reverse fault being modelled for the repository site in Sweden (Fälth et al, 2010), and both types for NPPs in Finland (Jussila et al, 2021). With regards to fault size, the 800–936 km^2 faults are in the upper range of interests, and perhaps a more relevant 400–600 km^2 fault should be used for benchmarking. Also, for the initial benchmarks a less complex case (i.e., not branched, step-over or inhomogeneous etc.) is proposed. The shear wave velocities, case description details and stress-drops do not give good guidance for the initial selection, as they fulfil the stated conditions or are ambiguously defined in the description of the benchmarking case.

The short-list of most suitable benchmark cases is highlighted with bold in Table 2, and it was decided that the strike-slip case TPV5 to be modelled. It was decided that the use of a variable stress drop case, or a fault with more complex geometric features (e.g. branching, or roughness) introduces unnecessary complexities in this benchmarking stage. The smaller faults were considered more compatible with the intended magnitude range, and a stress drop of at least 10 MPa was targeted.

Table 2. Characteristics of the benchmarking cases from Harris et al. (2009).

Case name	Faulting type	Initial stress drop (MPa)	Vs(m/s)	Fault size (km2)	Enough details
TPV3 / TPV4 / TPV5*	**Strike-slip**	**18.6**	**3464**	**450**	**Yes**
TPV6 / TPV7	**Strike-slip**	**18.6**	**2165**	**450**	**Yes**
TPV8 / TPV9	Strike-slip	Variable	3300	450	Yes
TPV10 / TPV11 / **TPV12*** / TPV13 *	Normal	Variable	3300	450	Yes
TPV14 / TPV15	Strike-slip branched	-	3464	420	Yes
TPV16 / TPV17	Strike-slip	-	3464	936	Yes
TPV18 / TPV19 / TPV20 / TPV21	Strike-slip branched	-	3300	420	Yes
TPV22 / TPV23	Strike-slip stepover	-	3464	2×600	Yes
TPV24 / TPV25	Strike-slip branched	-	3464	420	Yes
TPV26 / TPV26v2	Strike-slip	-	3464	800	Yes
TPV27 / TPV27v2	Strike-slip	-	3464	800	Yes
TPV28	Strike-slip hills	-	3464	540	Yes
TPV29 / TPV30	Strike-slip roughness	-	3464	800	Yes
TPV31 / TPV32	Strike-slip	-	Variable	450	Yes
TPV33	Strike-slip guided wave	-	Var. – min 2165	160	Yes
TPV34	Strike-slip Imperial	-	Var. – min 1400	450	Yes
TPV35	Strike-slip Parkfield	-	Var. – min. 1100	620	Yes
TPV101 / TPV102 / TPV103 / PV104	Strike-slip	25	3464	648	Yes
TPV105 / TPV105-3D	Strike-slip	-	3464	648	Yes
TPV205*	Strike-slip	-	3464	450	Yes
TPV210	Normal	Var.	3300	450	Yes

4.2 Description of case TPV5* and monitored outputs

The details of benchmarking case TPV5* can be found online (TPV5_forwebsite.pdf (scec.org)). The geometry is given in Figure 1.

The velocity model is homogeneous with pressure wave velocity of V_p = 6000 m/s, shear wave velocity of V_s = 3464m/s and density of δ = 2670 kg/m^3. The fault is a vertical right-lateral strike-slip planar fault within the half-space, reaching the Earth's surface (Figure 1).

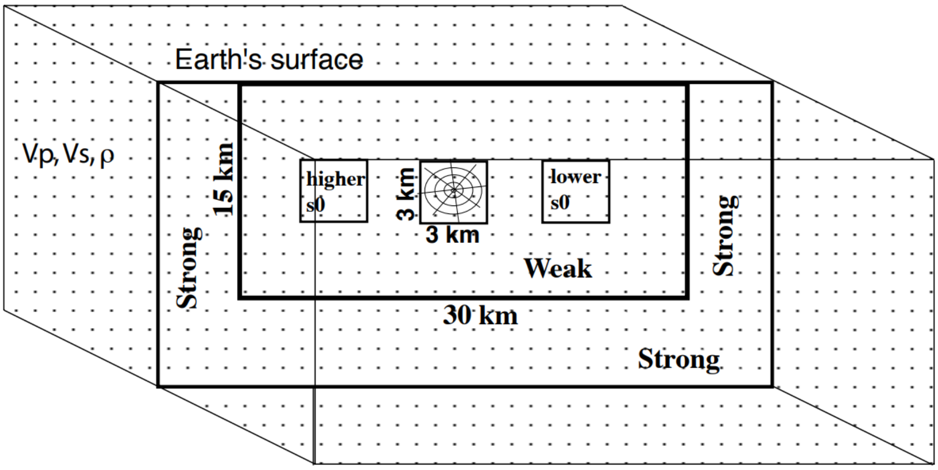


Figure 1. Geometry of the strike-slip fault TPV5. (Figure: SCEC, 2009)*

The rupture may occur within a rectangular area of 30×15 km (Figure 1). The boundaries of the potential rupture area are defined as strength (i.e., high-friction) barriers, able to arrest the rupture. The nucleation of the earthquake rupture is located 15 km along-strike and at 7.5 km depth. Nucleation occurs because the initial shear stress in a 3×3 km nucleation patch is set to be higher than the initial static yield stress in that patch. Failure occurs on the entire 15×30 km rupture plane, following a linear slip-weakening friction law.

Table 3. Properties of patches in TPV5.

Parameter	Patch with offset -7.5 km	Patch with offset 0.0 km (nucleation patch)	Patch with offset 7.5 km	Outside of patches
Static coefficient of friction	0.677	0.677	0.677	0.677
Dynamic coefficient of friction	0.525	0.525	0.525	0.525
Initial shear stress in the along-strike-direction	78.00 MPa	81.60 MPa	62.00 MPa	70.00 MPa
Initial shear stress in the along-dip direction	0.00 MPa	0.00 MPa	0.00 MPa	0.00 MPa
Initial normal stress	120.00 MPa	120.00 MPa	120.00 MPa	120.00 MPa
Initial static yield stress	81.24 MPa	81.24 MPa	81.24 MPa	81.24 MPa
Initial dynamic friction stress	63.00 MPa	63.00 MPa	63.00 MPa	63.00 MPa
Initial stress drop	15.00 MPa	18.60 MPa	-1.0 MPa	7.00 MPa
Slip-weakening critical distance	0.40 m	0.40 m	0.40 m	0.40 m

At the starting time of the modelling (t=0), the static and dynamic coefficients of friction are 0.677 and 0.525, respectively. Within the nucleation patch, the initial shear stress in the strike direction is 81.6 MPa, while it is 0 MPa in the dip direction. The normal stress across the fault is 120 MPa. There are two anomaly patches in the rupture area, a diminished and enhanced shear stress patches of 3×3 km. They are located at the right and left side of the nucleation patch, at the depth of 7.5 km. The shear stresses in these patches are 62 MPa and 78 MPa, respectively. The friction coefficients and normal stress are identical to those in the nucleation patch. Outside of the three patches, the shear stress is 70 MPa on the fault, and friction coefficients and normal stress are unchanged. The strong barrier to arrest the rupture is modelled by setting a high friction coefficient, outside of the intended rupture area. The properties of the different areas withing the TPV5 fault are given in Table 3.

4.2.1 Output monitoring points

The output points selected for monitoring the results of TVP5 are shown in Figure 2, and are summarised by coordinates in Table 4.

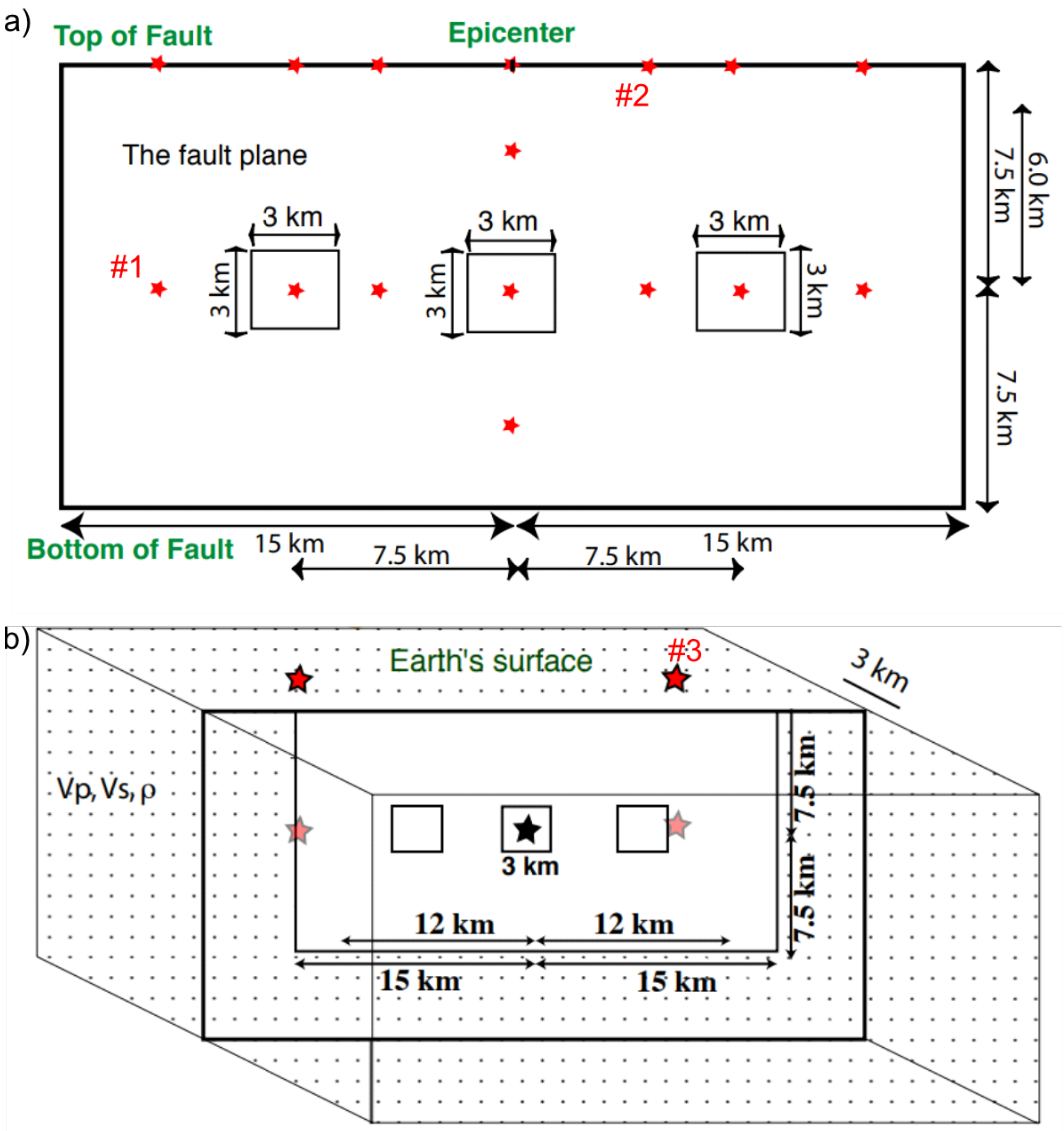


Figure 2. (a) On-fault and (b) off-fault monitoring points for the results of benchmark TPV5. The off-fault monitoring points are 3km away from the vertical rupture plane. The motion in points 1 (-12/7.5/0km), 2 (4.5/0/0km) and 3 (12/0/3km) are monitored in this study. (Figure: SCEC, 2009)*

Table 4. List of on fault and off fault receivers. The receivers used in this study as the reference monitoring points are marked as 1:, 2:**, and 3:***.*

On fault receivers		
X (m)	**Y (m)**	**Z (m)**
-12000	0	0
-7500	0	0
-4500	0	0
0	0	0
4500**	0	0
7500	0	0
12000	0	0
0	0	-3000
-12000*	0	-7500
-7500	0	-7500
-4500	0	-7500
0	0	-7500
4500	0	-7500
7500	0	-7500
12000	0	-7500
0	0	-12000
Off fault receivers		
X (m)	**Y (m)**	**Z (m)**
-12000	3000	0
12000***	3000	0
-12000	3000	-7500
12000	3000	-7500

The solutions available for the TPV5 benchmark are listed in Table 5. The overall rupture contour in the rupture plane is consistent between the different benchmark models (Figure 3). The rupture is presented as time when slip-rate first changes from zero to greater than 1 mm/s at a point. The contour is almost symmetric in relation to the depth of 7.5 km, due to the constant shear strength with depth; the symmetry is broken by the free surface at the top of the fault, compared to the boundary at the depth. The influence of the two strength anomalies of 3×3 km is clearly visible.

Table 5. Solutions available for the benchmark TVP5. (Harris et al., 2009)

Name of author and decryption of available results	
Brad Aagaard – Finite Element – EqSim	Yoshihiro Kaneko – Spectral Element – SPECFEM3D
Victor Cruz Atienza – Finite Difference – AWM	Yuko Kase – Finite Difference
Michael Barall – Finite Element – FaultMod	Duo Li (ADER-DG ExaSeis h = 506 m, O4)
Luis Dalguer – Finite Difference – DFM	Yi Liu – Boundary Integral
Eric Daub – Daub Finite Difference Code – 100 m	Shuo Ma – Finite Element – MAFE
Benchun Duan – Finite Element – EQdyna	Christian Pelties – Discontinuous Galerkin
Eric Dunham – Finite Difference – SGFD	Jan Premus – Finite Difference – FD3D_TSN
Kenneth Duru – WaveQLab3D – 100 m	Rihab Sassi – Boundary Integral – Hok code
Geoff Ely – Support Operator – SORD	Seok Goo Song – Dynelf
Jim Hazzard – Finite Volume – 3DEC	

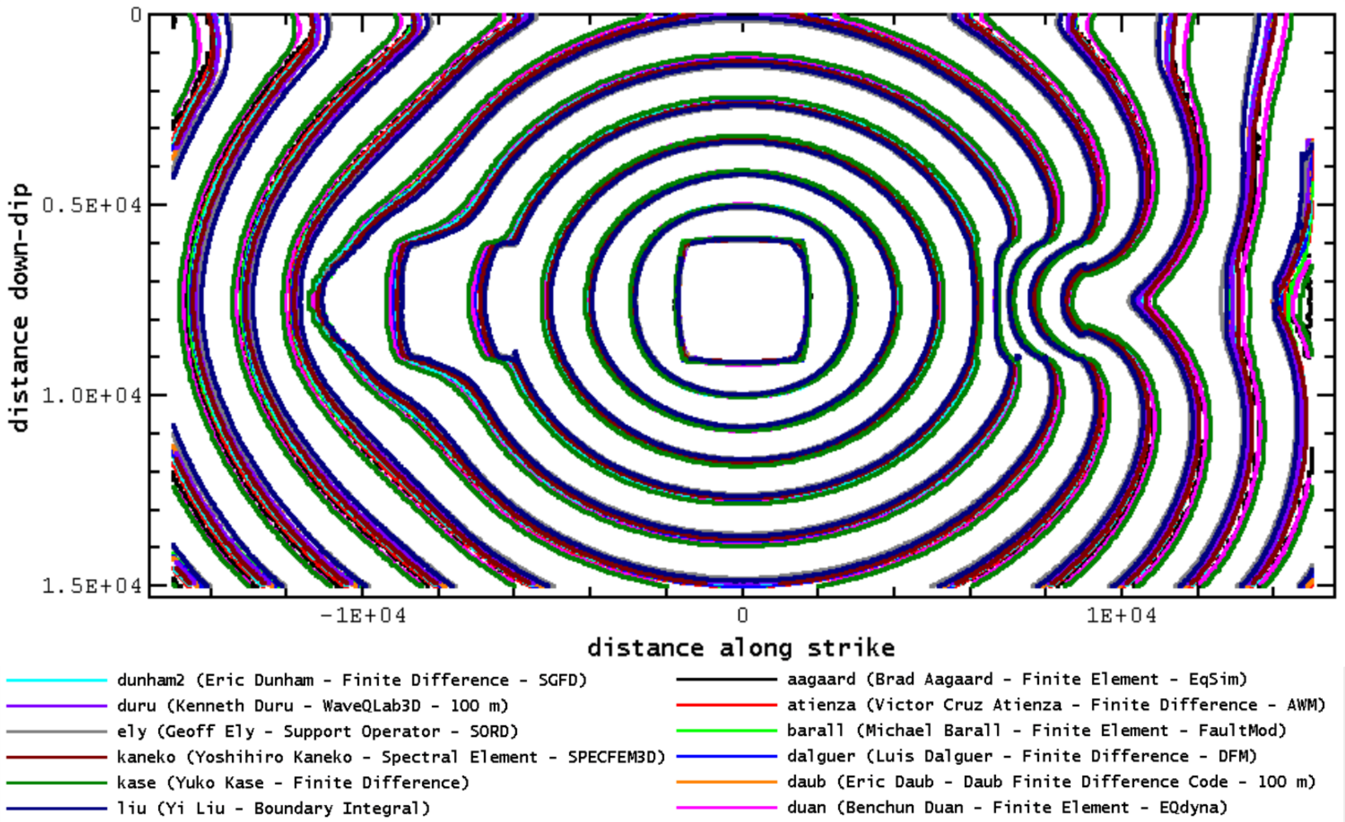


Figure 3. Rupture contour plot of the fault-plane according to different benchmark solutions. The contour shows the time of rupture velocity exceeding 0.001 m/s on the fault-plane. (Figure can be plotted at: https://strike.scec.org/cvws/cgi-bin/cvws.cgi)

The horizontal and vertical slip (in *m*), horizontal and vertical slip rate (in *m/s*) are shown for reference point #1 (distance along fault's strike -12.0 km, and dip 7.5 km – Figure 2) is given in Figure 4. The location of point #1 on the symmetry axis at the depth of 7.5 km is consistent with the slip values, with the vertical component being two magnitudes smaller compared to the horizontal component. The slip motion at point #2 initiates slightly after 4s after rupture initiation.

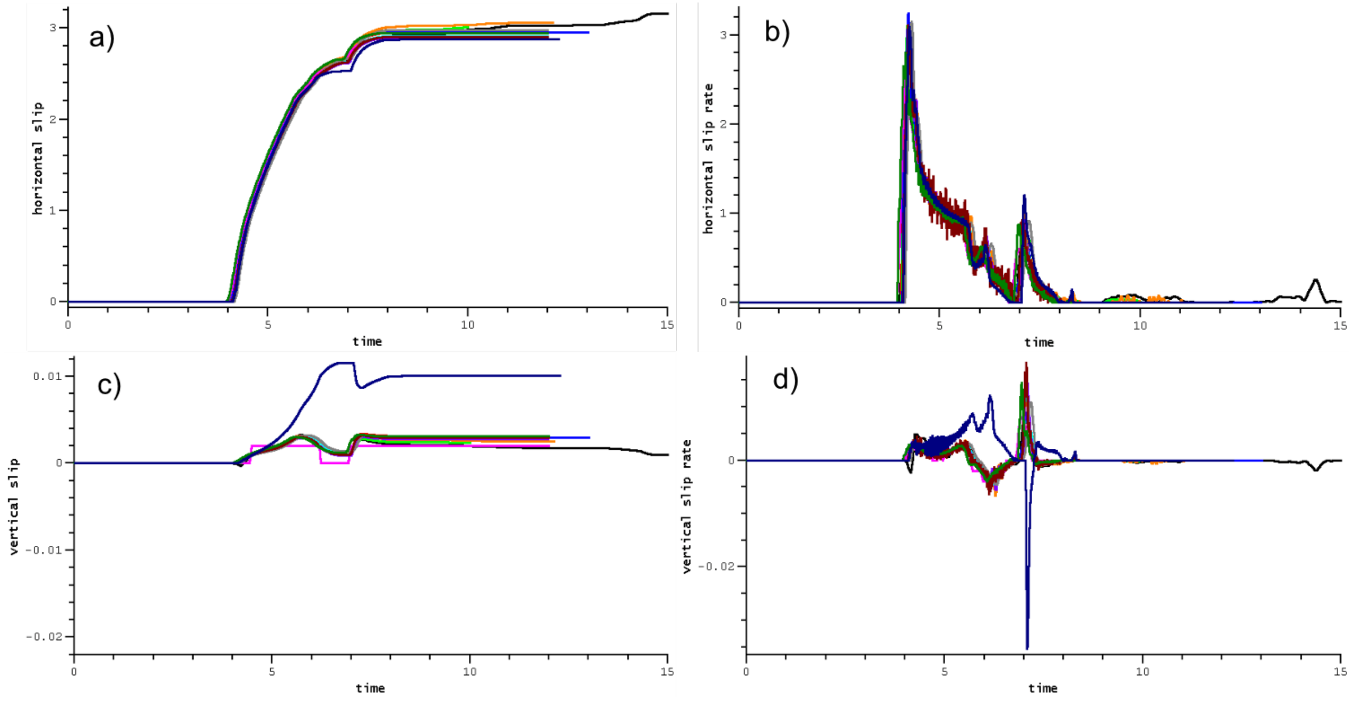


Figure 4. Horizontal slip /shear-displacement (in m) (a) slip-rate / displacement velocity (in m/s) (b) and vertical slip (c) and slip-rate (d) in monitoring point #1 (-12 km along strike, 7.5 km down dip) from Figure 2. Color-codes for the different solution are identical to those in Figure 3. (Figure: SCEC, 2009)

The horizontal and vertical slip (in *m*), horizontal and vertical slip rate (in *m/s*) are shown for reference point #2 (strike 4.5 km, dip 0 km – Figure 2) is given in Figure 5. The point is on the fault plane, at the ground surface. Due to the closer proximity of point #2, the horizontal slip initiates before 4 s. The horizontal slip amplitude is larger, compared to that in point #1, and there is a distinct vertical motion component (Figure 5c).

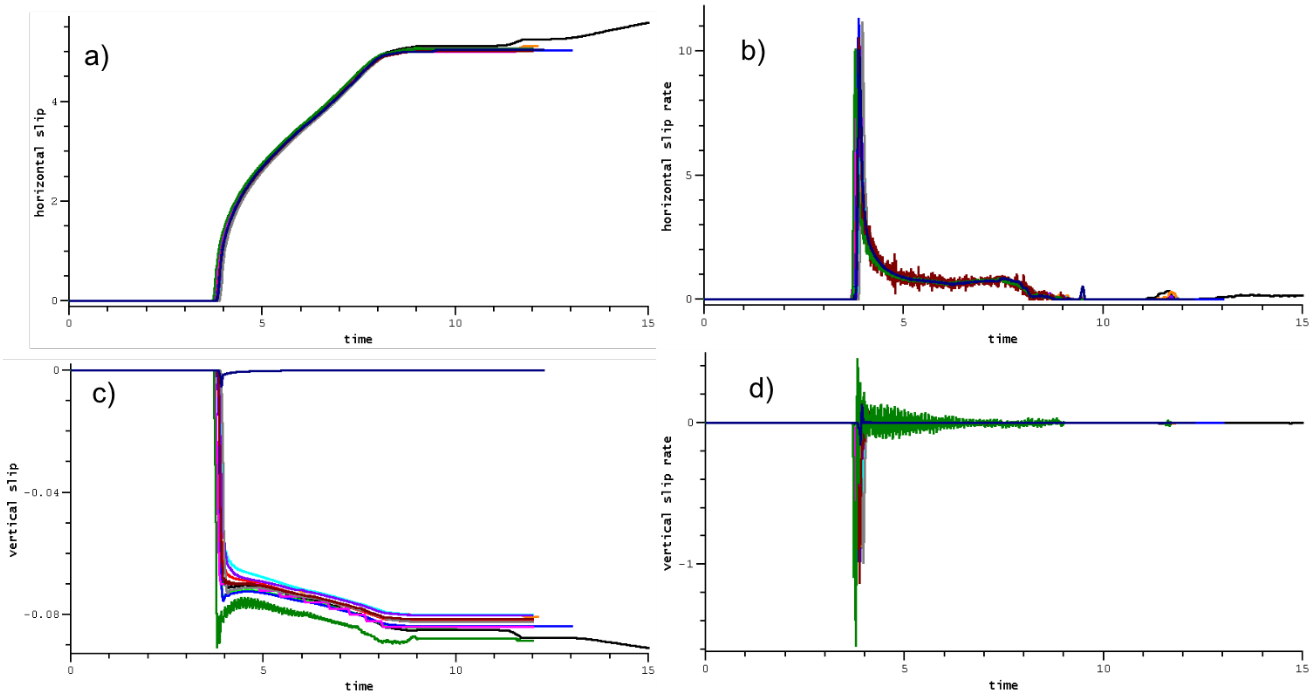


Figure 5. Horizontal slip (a) slip-rate (b) and vertical slip (c) and slip-rate (d) in monitoring point #2 from Figure 2. Color-codes are identical to those in Figure 3. Slips are in meters (m) while slip rate in (m/s).

DocuSign Envelope ID: F90AD6A6-9F1D-411F-82E4-0B573C18F616

The motion characteristics for the off-fault reference point #3 (strike 12 km, dip 0 km, off-fault 3 km – Figure 2) is given in Figure 6. The point is on the side of the fault plane at the distance of 3 km, at the ground surface. The distance to the zone of rupture initiation is slightly larger than that of point #1. Additionally, point #3 not being located on the fault itself, it only experiences displacement due to wave propagation in the solid. Hence, the motions at point #3 are ground motion properties. The wave arrival time is very close to 4 s. The horizontal displacement at point #3 (Figure 6a) is, naturally, smaller than the slip displacement on the fault (Figure 4a, Figure 5a). In addition, since point #3 is not on the fault-plane, which acts as a plane of symmetry, there is a significant normal displacement component in its motion (Figure 6b and c).

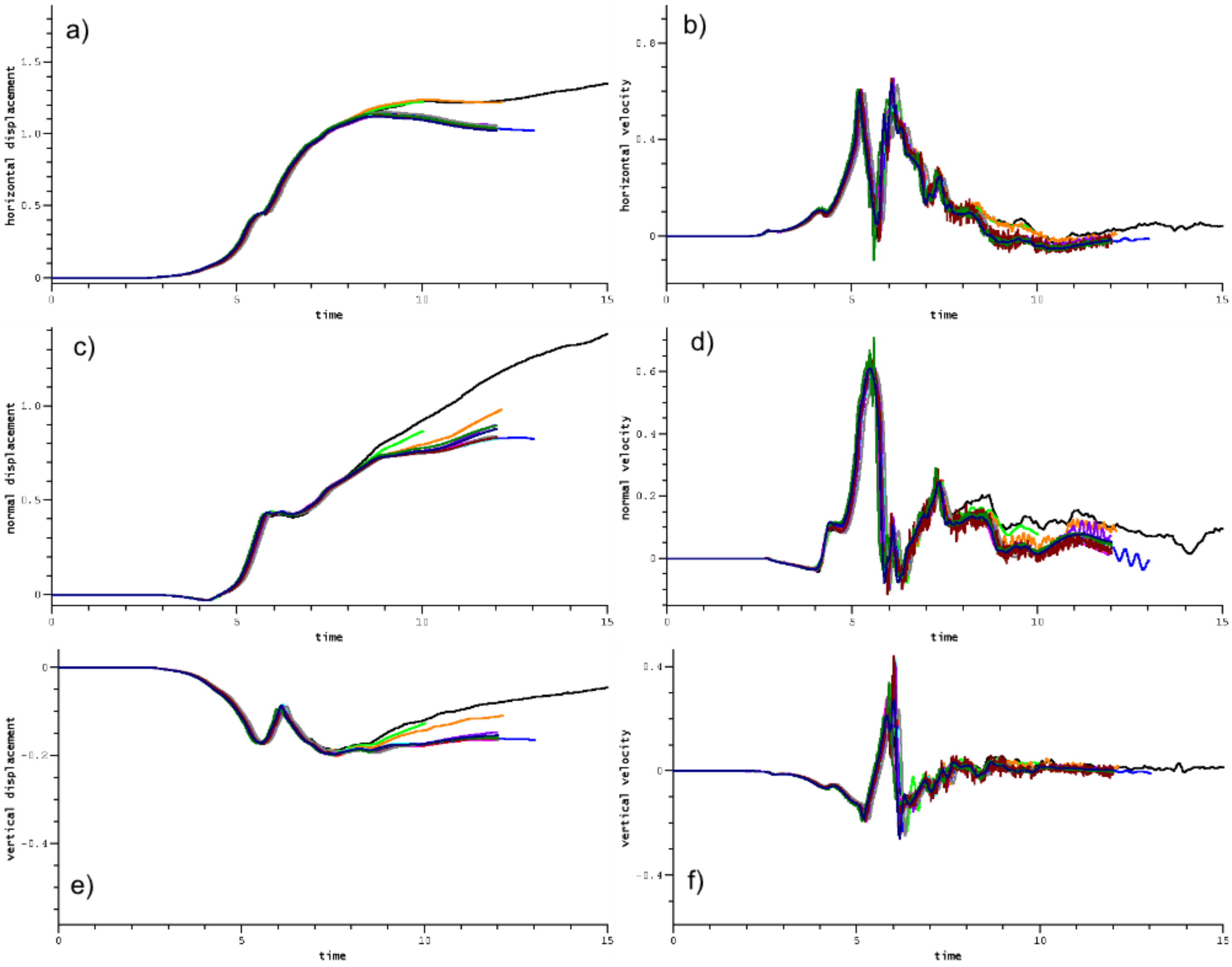


Figure 6. Horizontal displacement (a) velocity (b), vertical displacement (c) and velocity (d), and normal-to-fault displacement (e) and velocity (f) in monitoring point #3 from Figure 2. Color-codes are identical to those in Figure 3.

5. Protocols and software for simulating dynamic rupture

5.1 General purpose rock mechanics simulation software, 3DEC, PFC3D and FLAC3D

ITASCA (Itasca Consulting Group, Inc, n.d.) provides several commercial codes for numerical simulations. The codes all follow a specific workflow, with first setting up a model with fundamental elements (particles in PFC, a mesh of zones in FLAC3D and a set of jointed blocks in 3DEC), then defining the model constitutive behaviour and material properties, and assigning boundary and initial conditions. The solution of the simulation is reached after a series of computational steps, controlled by the user by monitoring the behaviour of the model. ITASCA's codes are using an explicit time marching method to solve the algebraic equations, which is different than the solution in conventional implicit-solution programs.

The code called 3D Distinct Element Code or 3DEC (3DEC, n.d.) provides a tool for numerical modelling for analysis of jointed and blocky material, such as jointed rock. It is commonly used for advanced geotechnical analyses of soil, rock, ground water, structural support, and masonry. In 3DEC, the discontinuous material is represented as an assemblage of discrete blocks. The discontinuities are treated as boundary conditions between blocks; large displacements along discontinuities and rotations of blocks are allowed. Individual blocks behave, based on constitutive and joint models, as either rigid or deformable material (i.e., meshed into finite difference zones). 3DEC has been frequently used for earthquake rupture and off-fault fracture displacement simulations for both Posiva and SKB (e.g., Hökmark et al., 2019). The exercise TPV5 has been successfully run and the code benchmarked, and off-fault fracture response studied for an earthquake at a hypothetical geological repository site in Canada (Blanksma et al., 2022).

The code called (Fast Lagrangian Analysis of Continua in 3D (FLAC3D, n.d.), is a code for geomechanics continuum modelling. It can solve complex geotechnical problems for three-dimensional analyses of soil, rock, concrete, structural ground support, and groundwater flow. FLAC3D provides a tool to simulate geotechnical conditions for engineering applications, such as slope stability, behaviour of underground excavations, and simulation of seismic response.

Particle Flow Code in 3D (PFC, n.d), is a general purpose, distinct element modelling framework. PFC models synthetic materials composed of an assembly of variably-sized rigid particles that interact at contacts to represent both granular and solid materials. PFC models simulate the independent movement (i.e., translation and rotation) and interaction of many rigid particles that may interact at contacts based on an internal force and moment. Contact mechanics obey particle-interaction laws that update internal forces and moments.

The performance of FLAC3D/PFC3D combination in fault-rupture and off-fault fracture simulations has been studied by Darcel et al. (2019). They compared the performance of coupled PFC/FLAC3D environment to the results acquired with 3DEC simulations by Fälth et al. (2015). By using PFC in a fault-rupture and off-fault fracture domain surrounded by a partly overlapping FLAC3D continuum domain, they found good agreement with the previously widely used 3DEC method.

In this study, a novel modelling method combining FLAC3D and PFC is utilized in order to run models more quickly and easily. A new contact logic for discrete joints within a FLAC3D continuum model utilizing PFC contacts is introduced by Itasca and provided for the work before it's commercial launching. The logic fully combines the codes describing the model domain as a FLAC3D continuum and the joints as PFC discontinuities. No distinct domains of FLAC3D and PFC are separated in the model. The joint constitutive models used in 3DEC are introduced for these so-called *zone joints.*

The new FLAC3D zone joint simulation method was first studied by comparing the performance to the published TPV5 results gained with 3DEC (Blanksma et al., 2022). The aim was to demonstrate that the new zone joint logic is performing equally well as the 3DEC code, since no previous experience of the dynamic behaviour of such simulations exists.

The model volume with the fault geometry was built first in Rhino (McNeel et al., 2022) using the Griddle mesher (Griddle, n.d). The simulation model was compiled in the coding platform of FLAC3D. The fault was run into a prestressed state with fault patches having different friction properties able to maintain the predefined amounts of shear stress on the patches given in the exercise description (Table 3). The model was run into static plastic equilibrium, letting the fault shear and reach the stress-setup given in the exercise. Then the monitoring parameters were initiated, dynamic behaviour of the model was allowed, model boundaries were set absorbing/viscous, and the spontaneous rupture initiated by setting the friction angle equal on the entire rupture plane, i.e., to 34.1° also on the right, left and nucleation patches where it deviated earlier in order to preserve the prespecified amount of shear stress (cf. Figure 1.). Setting the friction angle to 34.1 on the nucleation patch, initiated the rupture. Along the rupture front, friction angle was decreased linearly from the static friction of 34.1° to the dynamic value of 27.7° while the slip was reaching the slip weakening distance $d_c = 0.4$ m. With larger slips and after slipping, the friction angle was set to remain as the dynamic value, i.e. no healing of fault was allowed. The rupture was arrested outside the predefined 15 km x 30 km rectangular area by a friction angle of 90°. The shear stress, shear slip and slip rate histories on the monitoring points on the fault, and velocities and displacements at the off-fault monitoring points were gathered.

The first models were run with a coarse mesh to find stable and effective means to introduce the stresses on the fault, to refine the rupture initiation and propagation on the fault and to code the softening-healing Mohr joint constitutive model. In addition, means to track the monitoring parameters set in the exercise were studied. Significant efforts were required to gain a stable FLAC3D/PFC run, performing equally with the 3DEC code. The required softening-Mohr joint model was first lacking and was required to be scripted with the Itasca's in-built scripting code (.FISH) in the simulation model in order to run the dynamic simulation as given in the TPV5 exercise. Before running the model with the given mesh size of 100 m on the fault (cf. Ch. 6.1), a model with a coarser mesh having an uniform 500 m element size on the fault and at the 3 km distance set for the off-fault receivers from the fault (Figure 2, Table 4), and an gradually increasing element size up to 4 km along the increasing distance towards the model boundaries, i.e. within ca. 25 km distance from the off-fault receivers. It is noted, that such a coarse model setup limits the frequency bound which the simulation is able to produce, and was purposed only to demonstrate the equal performance of the previously benchmarked code 3DEC and the new zone joint logic in FLAC3D.

The testing resulted in several modelling setups with only minor differences in the response (Figure 7, Figure 8, Figure 9). To reach faster simulation performance, the softening-healing Mohr model was later added in the code as an in-built selection by Itasca. Also, optimisation of the model discretisation was tested to further improve the simulation run time.

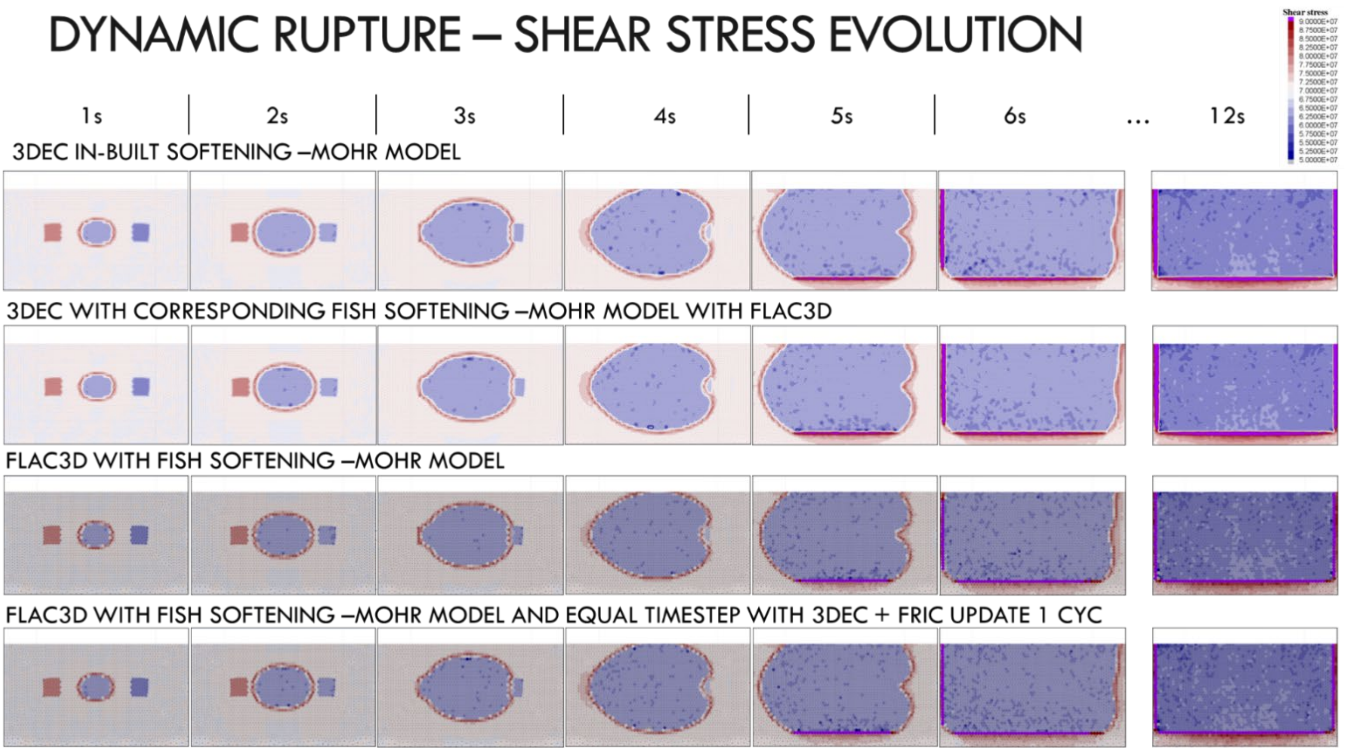


Figure 7. An example of shear stress evolution on the fault with comparative runs with a coarse mesh. Same model was run with four different set-ups: in 3DEC with the in-built and .FISH scripted softening-Mohr joint model and in FLAC3D with two differently updating .FISH scripted softening-Mohr models.

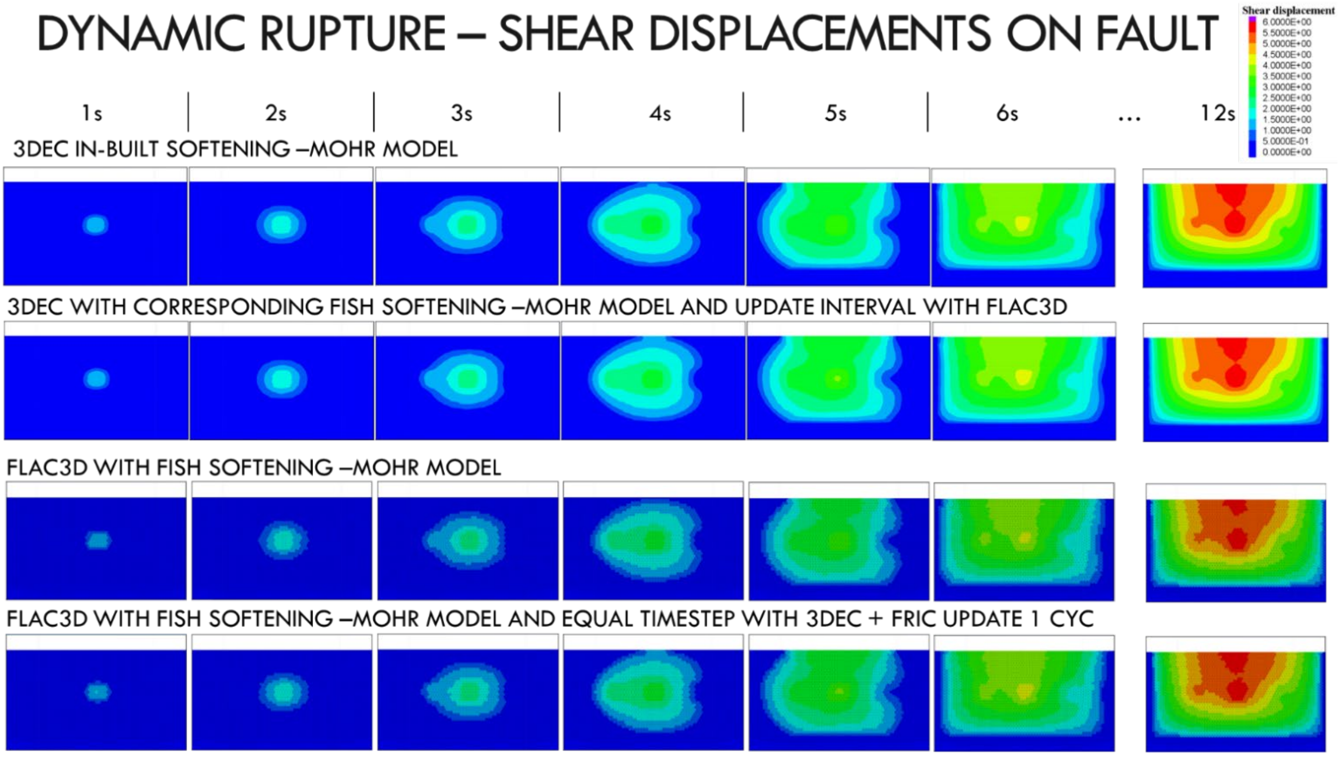


Figure 8. An example of shear displacement evolution of the fault with comparative runs with a coarse mesh. Same model was run with four different set-ups: in 3DEC with the in-built and .FISH scripted softening-Mohr joint model and in FLAC3D with two differently updating .FISH scripted softening-Mohr models.

DocuSign Envelope ID: F90AD6A6-9F1D-411F-82E4-0B573C18F616

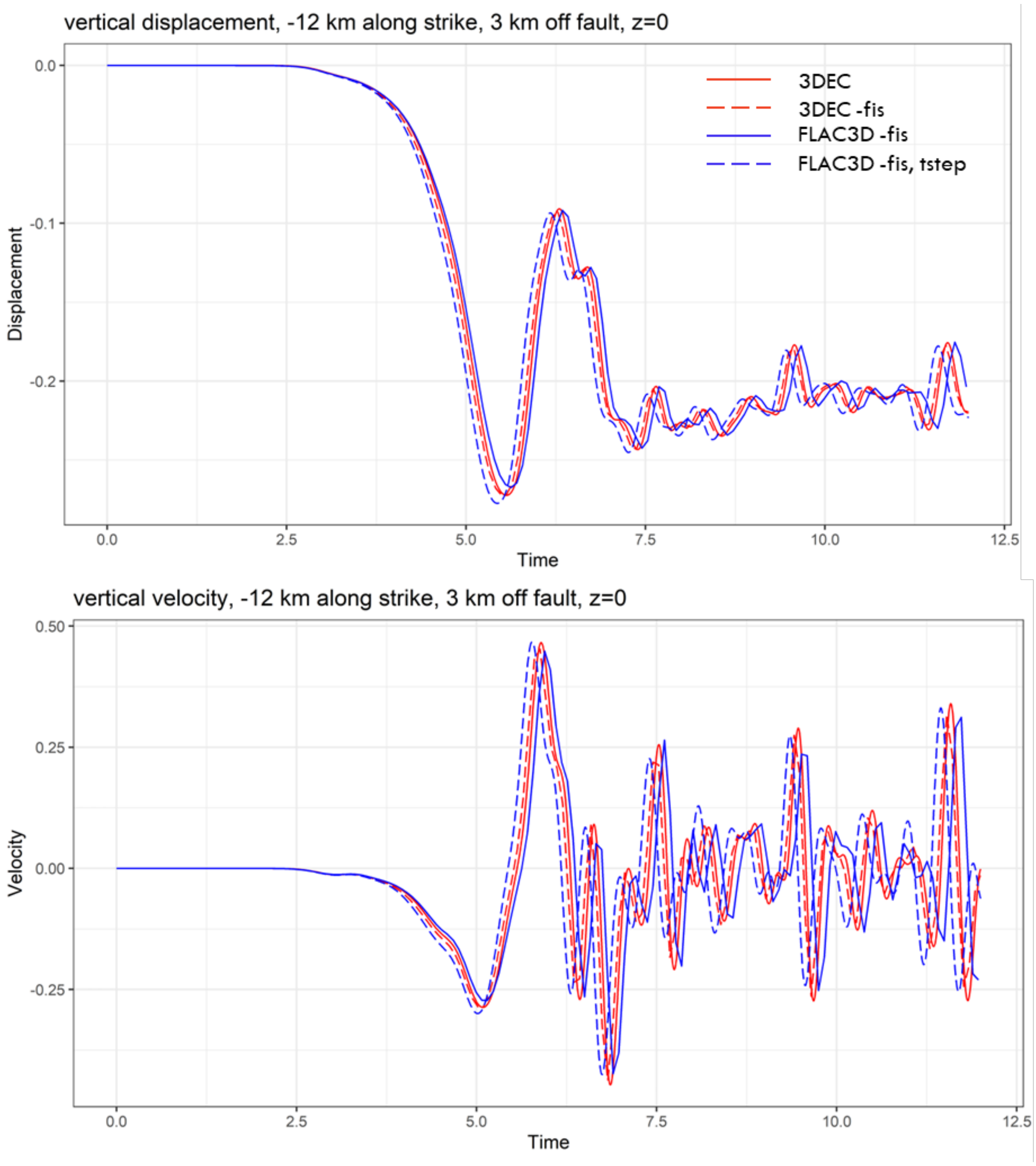


Figure 9. An example of off-fault response of the model in the comparative runs with a coarse mesh. Vertical displacement (m; top) and vertical velocity (m/s; bottom) at a point #3 in Figure 2, located 12 km along-strike and 3 km perpendicular from fault. The same model was run with four different setups: in 3DEC with in-built and .FISH scripted softening-Mohr joint model and in FLAC3D with two differently updating .FISH scripted softening-Mohr models. fis refers to the softening-Mohr model scripted with ,FISH, and tstep to a FLAC3D model with corresponding dynamic timestep as in the 3DEC runs.

5.2 Fault rupture simulation with SeisSol

SeisSol is a program (SeisSol, n.d.), which is capable simulate fault rupturing, wave propagation and ground motion. It can be considered as variation of the traditional finite element method (FEM). The word traditional implies that differential equations are solved with variational and weighted residual methods. Rayleigh-Ritz is an example of variational methods, whereas the Galerkin method is an example of weighted residual method. Instead of those, discontinuous Galerikin method is the core of SeisSol. An introduction to the method can be found in Igel (2017).

As the name suggest, solution fields of discontinuous Galerkin method are not continuous. The mass and stiffness matrices are defined for each element separately, resulting in high efficiency and parallelizing when powerful computers are available. Computational efficiency is improved because inversion of large system level mass and stiffness matrices are not required. Neighbouring elements do not share nodes, and elements are linked with flux scheme, which implies that an element is only interacting with neighbours.

The discontinuous Galerking method share with traditional FEM usage of weak formulation for differential equations, boundary conditions, and discretization of considered domain by finite elements Igel (2017). Utilizing finite elements SeisSol can solve ruptures, wave propagations, and ground motions of complex geometries including underground structures. The discrete Galerkin method has been adapted to Earth sciences in last 15 years. Currently the geometry can be discretized only with triangular and tetrahedral elements.

SeisSol is versatile solver. It can solve all typical fault typologies; friction may vary on a fault plane, and it may have branches. Available material models are elastic, anisotropic, poroelastic and viscoelastic. For longer off fault receiver distances attenuation can be taken account. Friction laws controlling the rupture process of a fault can be linear slip-weakening or rate-and-state friction. The latter has option of ageing law, slip law and strong velocity weakening. For rate-and-state law user may define thermal pressurization. User may define initial condition, however SeisSol is shipped with eight initial conditions in which zero, planar wave, superimposed planar wave and travelling wave seems to be most relevant. Available boundary conditions are free surface, dynamic rupture and absorbing.

5.2.1 Modelling TPV5 with SeisSol

The procedure for creating a model with SeisSol follows identical process as with any other FEM software. Typically, a geometry is created in the first place, which is followed by meshing. SeisSol does not have capabilities for creating geometry or mesh. These tasks can be done with Gmsh.

In the benchmark TPV5 the geometry is simple, consisting of a box of 120×120×60 km, a vertical fault plane with size of 30×15 km, and a patch for nucleating rupture of the fault having size of 3×3 km. The model geometry is shown in Figure 1. The meshing was done with the following command:

```
        $ gmsh -3 tpv5.geo
```

where .geo is Gmsh file format and TVP5 is the name of the file. The generated mesh is shown in Figure 10 and it have 1891980 tetrahedral elements for rock volume and 55962 triangle elements for the fault plane. SeisSol cannot interpret the format, and it was translated with PUMGEN with following command (see a note of the syntax from Appendix A):

```
        $ pumgen tpv5.msh -s msh2
```

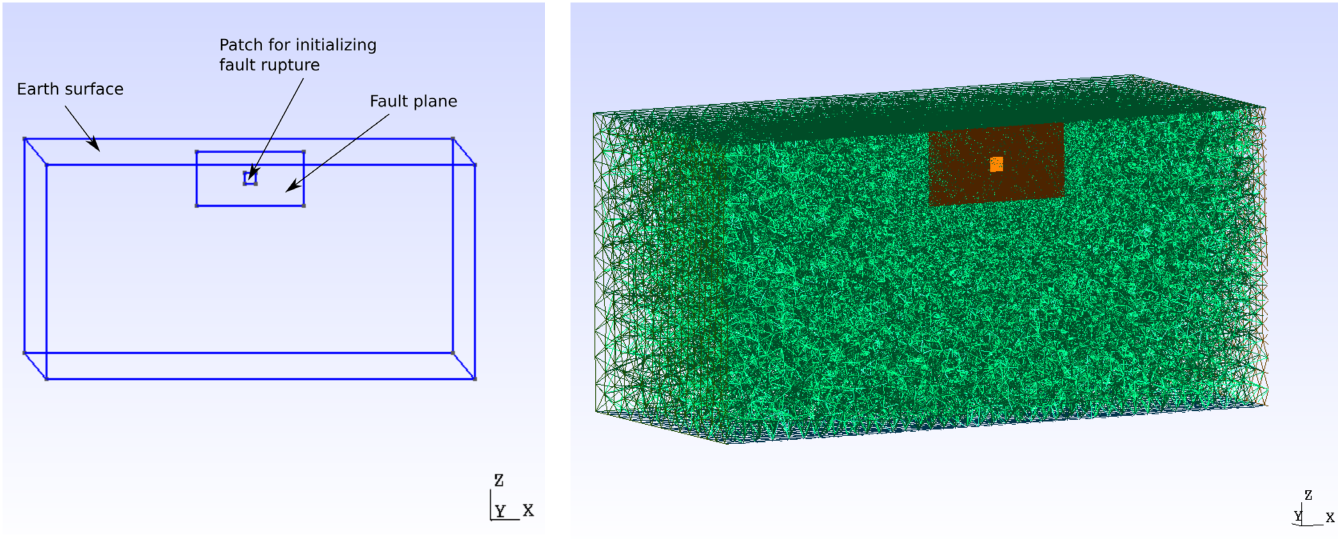


Figure 10. Benchmark TPV5: geometry and mesh for SeisSol.

SeisSol uses the yaml scripting language in definition of a material model and fault plane properties. The advantage of using yaml is the possibility to provide data in terms of constants and functions. In the TPV5 benchmark the material model of the surrounding rock is linear elastic, requiring three parameters, which are density ρ and Lamé parameters μ and λ. The initial stresses in the fault plane are given in Table 3, including the nucleation, rupture weakening and strengthening patches.

The boundary conditions are defined in Gmsh. The top of the rock volume is a free surface, bottom and sides are absorbing boundaries and the fault plane, including the patch, have dynamic rupturing boundary conditions (Figure 11). They are given manually by editing the .geo file with keywords given in Table 6.

Table 6. Boundary conditions of the SeisSol model.

Boundary condition	Keyword
Free surface	Physical surface (101)
Dynamic rupturing	Physical surface (103)
Absorbing boundary	Physical surface (105)

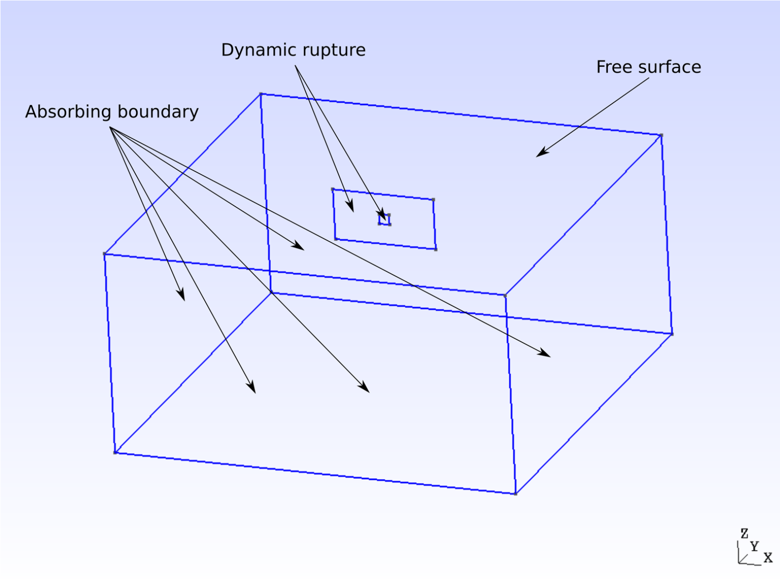


Figure 11. TPV5 boundary conditions: top of volume is free surface, sides and bottom are absorbing boundaries, the fault-plane and the patch have dynamic rupture boundary conditions. The exterior faces, expect the top-face, of the modelled rectangular-prismatic volume are absorbing boundaries.

SeisSol has several data formats to store receiver data. Although, developers are proposing use of Paraview for most analysis of results the traditional ASCII format is useful when data is required in a particular location. In TPV5 results were requested on and off fault locations given in Table 4. On-fault and off-fault receivers provide different type of output data. On fault receivers consist of data related to the rupturing of the fault, while off-fault receivers provide data of stress state and velocity. The displacement was integrated from velocity with the Scipy function *integrate.cumulative_trapezoid*.

As last step prior to submitting the SeisSol job, the user must define a parameter file, which has a suffix .par. This file contains information of material and fault model files indicated with suffix .yaml, user can switch on or off boundary conditions[1], request particular set of data from on and off fault receivers, define output file formats and their locations, set length of simulation and time step, and other parameters for controlling SeisSol.

5.2.2 Running the model

The SeisSol job is submitted from the launch_SeisSol folder where all input files and the SeisSol executable is copied. In addition, SeisSol requires unlimited stack size within a simulation, and it can be changed temporarily with the command:

```
        $ ulimit -s unlimited
```

The job is submitted with the command:

```
        $ OMP_NUM_THREADS=xx mpiexec -np yy ./SeisSol_Release_dhsw_6_elastic
parameters.par
```

where xx and yy is number of threads and cores available for the simulation and SeisSol_Release_dhsw_6_elastic is the SeisSol executable. The number of threads and cores are depending on the computer architecture.

---

[1] The parameter file has an option to switch on or off boundary conditions although they are defined in .geo file.

5.2.3 Rupture of a primary-fault initiating rupture in a secondary-fault

Let’s consider scenario of two independent faults having significant distance in between, and let’s assume the first fault ruptures at time 0.0 s. The rupture of the first fault induces a propagating wave field, which arrives to the second at time *t*. The second fault has an initial and passing wave field induced stress state. The second fault rupture nucleates only if a location or locations becomes unstable.

In TPV5 the model for nucleation is slip-weakening, which can be likened to a simple mass-spring model. If the mass is placed on a surface, the contact force to the surface and the static friction creates a friction force. As the spring pulls the mass, this friction force resists the movement. Once the pulling force exceed friction force the mass slips, and the static friction transforms to dynamic friction. As the mass slips, the spring relaxes reducing the pulling force, resulting in slowdown of the mass’s movement. Putting the slip-weakening condition in context of TPV5 then the slip occurs when

$$\tau > \sigma_n \cdot \mu \tag{1}$$

Where: τ is the shear stress in the strike direction and σ_n and μ are normal stress and static friction respectively.

When propagating wave is nucleating the second fault the problem is coupled implying that the stress state is strongly time dependent as long the wave trains pass the faulted region. If the condition in Eq. 1 fulfilled, then the rupture is nucleated.

Another important aspect is an assumption of a linear elastic material model of a fault surrounding volume. Plastic models are also included to the ground motion simulations, but they are typically taking place in a fault plane. An implication of linear elasticity is a superposition of any time dependent variable, which provides several possibilities consider the rupture of secondary fault.

As already mentioned, the rupture of secondary fault can be simulated as coupled problem or the nucleation of secondary fault is excluded in the simulation of rupture of the primary fault. The latter results some degree of inaccuracy to the stress field since the time dependency of shear and normal stresses in secondary fault are disregarded. In this case user simulates the rupture of the primary fault in the first run. The stress state is recorded in preselected nucleation point or set of points in the secondary fault. The total stress state would be the initial stress state of the secondary fault and the maximum stress state induced by the propagating wave field from the primary fault. If condition in Eq. 1 is fulfilled a new model shall be constructed, in which the secondary fault rupturing is simulated. Since superposition holds the results of these two simulations are summed in desired locations timewise keeping in mind the time shift of the nucleation.

Last consideration is whether the geometry of the secondary fault should be modelled in the simulation of the rupture of the primary fault. As a first best guess it can be omitted, which in terms of physics is incorrect since it creates a boundary condition. Although, the assumption may be incorrect it is common to model fault plane faces as welded indicating that they are perfectly connected (e.g. Fälth et al, 2015, Jussila et al, 2021).

5.2.4 Can secondary-fault rupture be modelled with SeisSol?

A standard problem with SeisSol is simulation of fault rupturing, wave propagation, and ground motions. The primary fault rupture can be extended to branch faults ruptures. However, as the header suggest is it possible to simulate a case consisting primary fault rupture and following wave propagation, which initiate a new fault rupture in a secondary fault plane not connected to the primary fault? Several notes can be made.

The discussion starts with the features required to initiate secondary-fault rupture. A user should be able define (1) the geometry and mesh of the secondary fault plane, (2) define a patch where the rupture begins,

(3) define initial stress state at the secondary fault plane, (3) propagate the stresses from the rupture of the primary fault and (4) define a trigger initiating the rupture.

Defining and meshing the secondary fault plane can be done in SeisSol. The patch initiating the rupture can be defined. In a user-friendly method it would be possible to define several faults, but not directly in the input file. However, yaml scripting language provides possibility for functions and logical statements, which provides flexibility to define an initial stress state at the patch and on the secondary fault plane. The parameter file .par shows that SeisSol computes the stress state in a rock volume. The user can request outputs off-fault, anywhere in the rock volume, and can mask the stress state reporting on or off. The last requirement would be to trigger a rupture in a secondary fault. Such a trigger would be easy to implement. SeisSol should define the stress state at the rupture initiating patch of the secondary fault on every time step. Once the stress exceeds the predefined threshold value, the secondary rupture begins.

The discussion above only describes possibilities provided by yaml scripting language, and the output from SeisSol but it does not imply SeisSol is able to interpret input data without modifications. The open question is whether SeisSol is able to associate stress state data to a secondary fault and if it is possible to initiate rupturing. Despite of SeisSol being an open-source project, the authors do not know the details of the SeisSol source code structure. SeisSol can be modified, however the difficulty level is depending on the structure of the source code. It is certain that at least with modified SeisSol, simulating triggered rupturing of independent faults is possible.

As discussed, SeisSol might be able to simulate a coupled problem in a rupture of a secondary fault with some modifications. However, the user should begin with simulating the primary fault rupturing and then construct an additional model for the simulation of the secondary fault if the total stress state exceeds condition in Eq. 1.

6. Results of the example fault-rupture simulation

6.1 Results with FLAC3D zone joint logic

After reaching a stable modelling setup with the new FLAC3D/PFC zone joint logic (cf. Ch. 5.1), a run with properties corresponding to the given TPV5 exercise and with the new in-built softening-Mohr joint constitutive model was set up. A denser mesh with an element size of 100m on the fault and in the positions of the off-fault stations was used. The element size of the model increased towards the model boundaries. The effect of mesh degradation is seen as small oscillations on the graphs presenting station or slip velocities.” As in the test runs with the coarse model discretization, the fault was run into a prestressed state with fault patches having different friction angles able to maintain the predefined amounts of shear stress on the patches.

The response of the simulation was compared to the set of results given by the TPV5 exercise. The rupture front was computed on the fault plane and drawn as 1 second intervals (Figure 12) and corresponded well with the ones given by the exercise. The slips and slip rates on the fault at the monitoring points #1 and #2, and horizontal, vertical and normal-to-fault displacement and velocity histories at an off-fault monitoring point #3 are shown in Figure 13, Figure 14 and Figure 15, respectively.

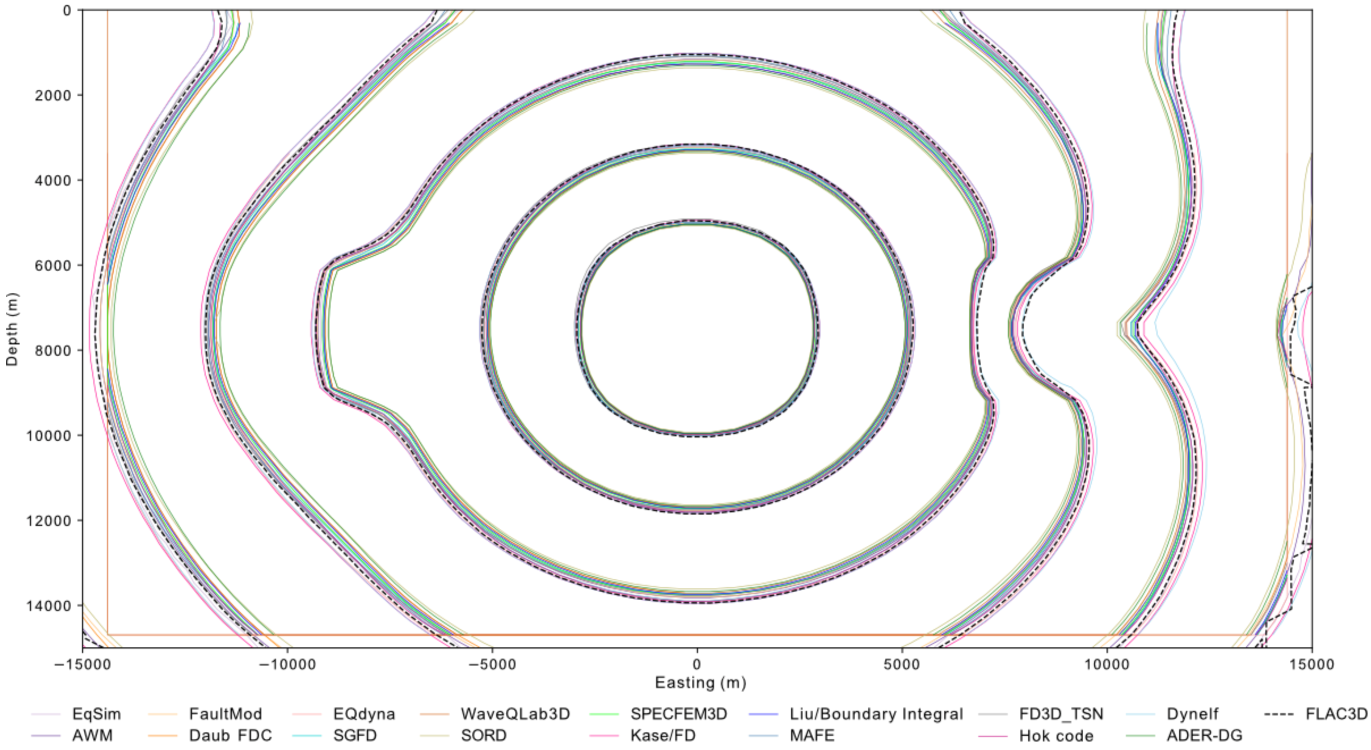


Figure 12. Rupture contour plot of the fault plane according to different benchmark models. The contour shows the time of rupture propagating on the fault plane with 1 s intervals. Black dashed lines refer to the simulation conducted with FLAC3D zone joint logic in this study. (Raw data: SCEC, 2009)

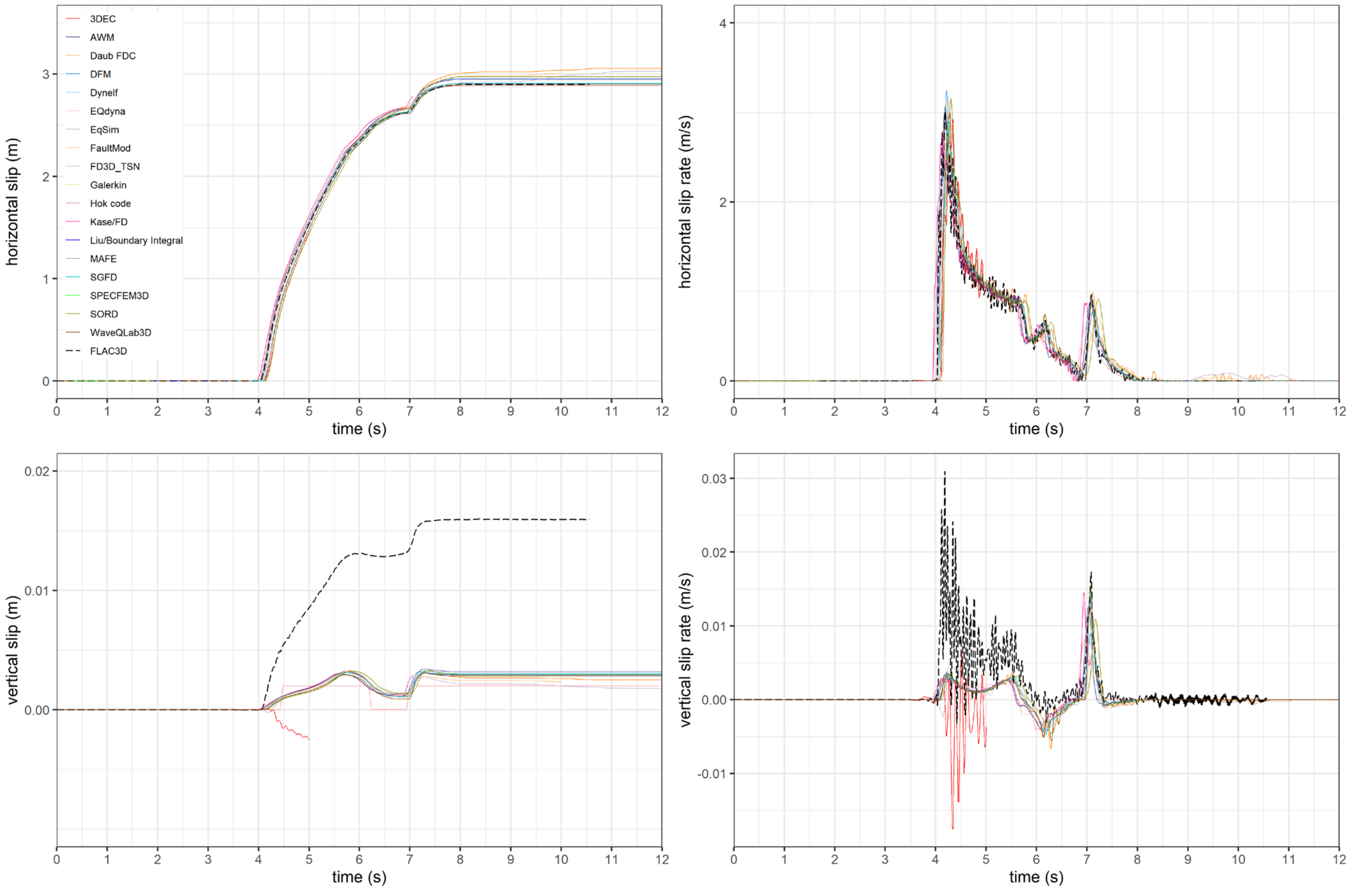


Figure 13. Horizontal displacement and velocity (top row), and vertical displacement and velocity (bottom) in the monitoring point #1 (-12 km along strike and 7.5 km down dip) from Figure 2. Black dashed lines refer to the simulation conducted with FLAC3D zone joint logic in this study. (Raw data: SCEC, 2009)

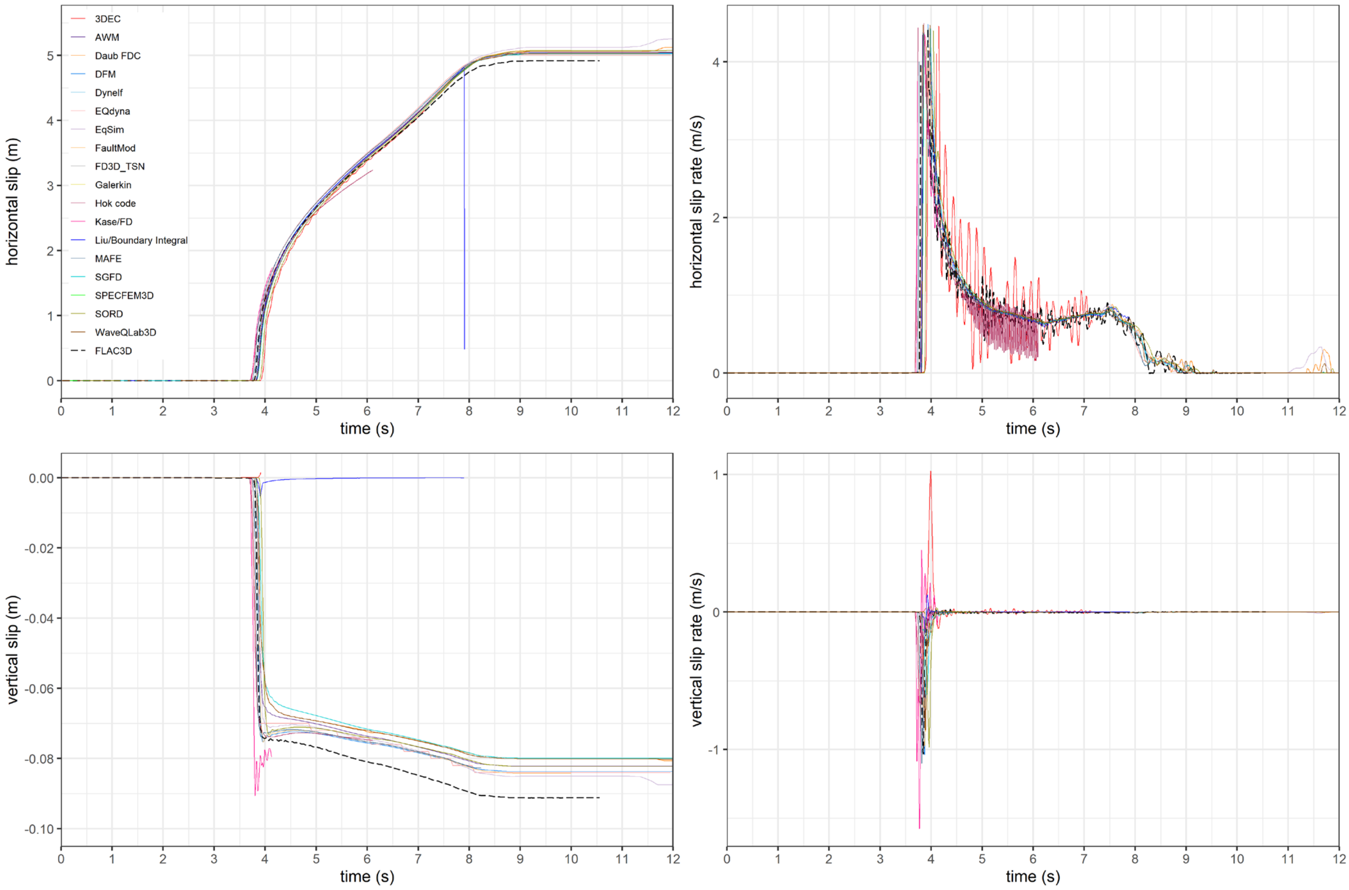


Figure 14. Horizontal displacement and velocity (top row), and vertical displacement and velocity (bottom) in the monitoring point #2 (4.5 km along strike at surface) from Figure 2. Black dashed lines refer to the simulation conducted with FLAC3D zone joint logic in this study. (Raw data: SCEC, 2009)

DocuSign Envelope ID: F90AD6A6-9F1D-411F-82E4-0B573C18F616

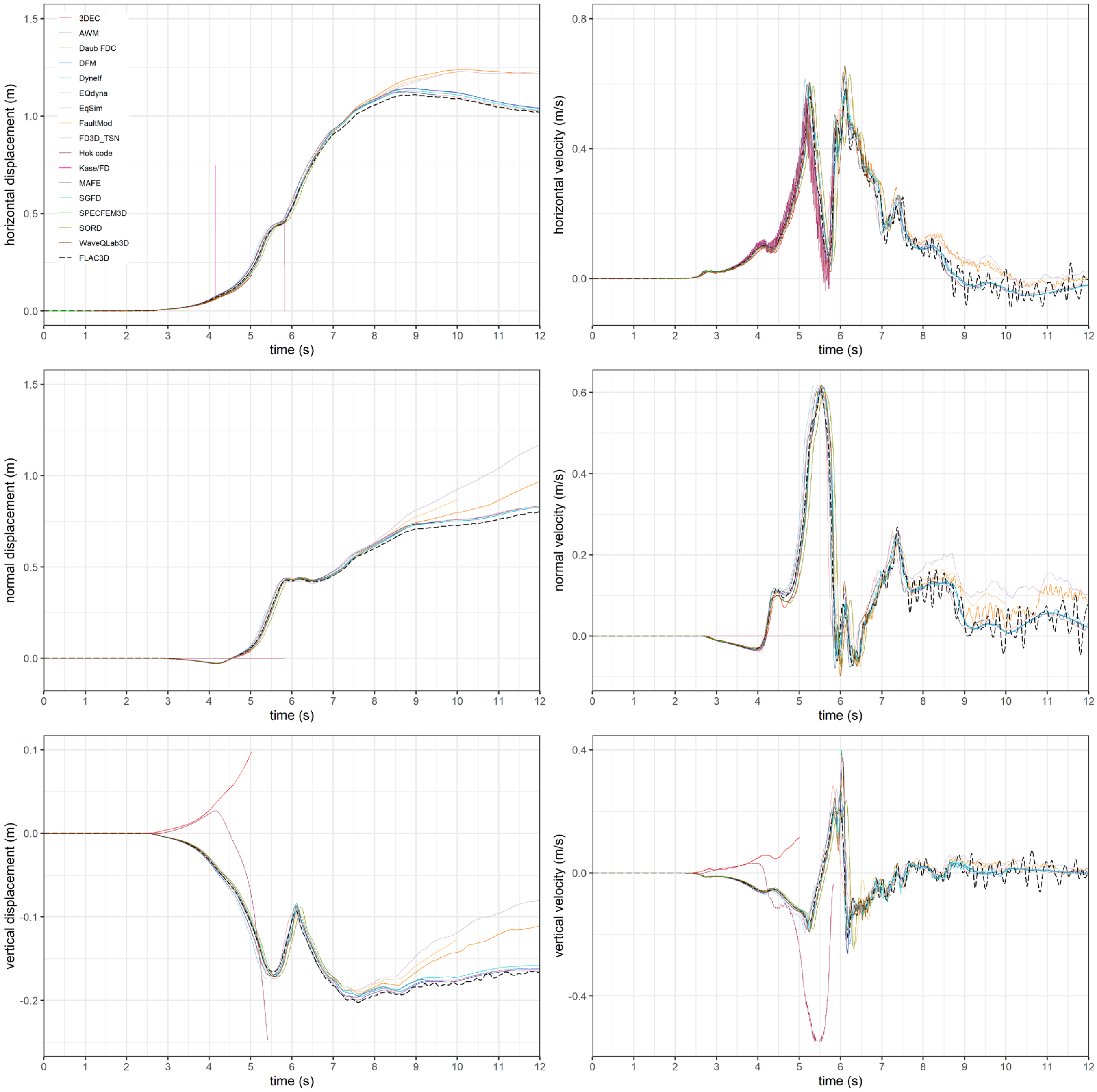


Figure 15. Horizontal displacement and velocity (top row), vertical displacement and velocity (middle), normal displacement and velocity (bottom) in the off-fault monitoring point #3 from Figure 2. Black dashed line indicates the results gained with the FLAC3D zone joint method. (Raw data: SCEC, 2009)

The dynamic run required approximately one hour of computation time, which is a great improvement if compared to 3DEC simulations. Model setup was quick after preliminarily defined scheme fort the setup and readily meshed model. Introducing the initial stress state on the fault required approximately two hours' computation time.

6.2 Results with SeisSol

Snapshots of slip rate in strike direction and displacement of ground in three directions are shown in Figure 16 and Figure 17 respectively. The propagation of the rupture with SeisSol, computed on the fault plane is shown in Figure 16. These propagation waves are replicating the contours from Figure 12, with good precision. The waves arriving to the ground-surface are shown in Figure 17. They are compatible with the expected ground-motion pattern of a strike-slip rupture (e.g. reported in Jussila et al. 2021).

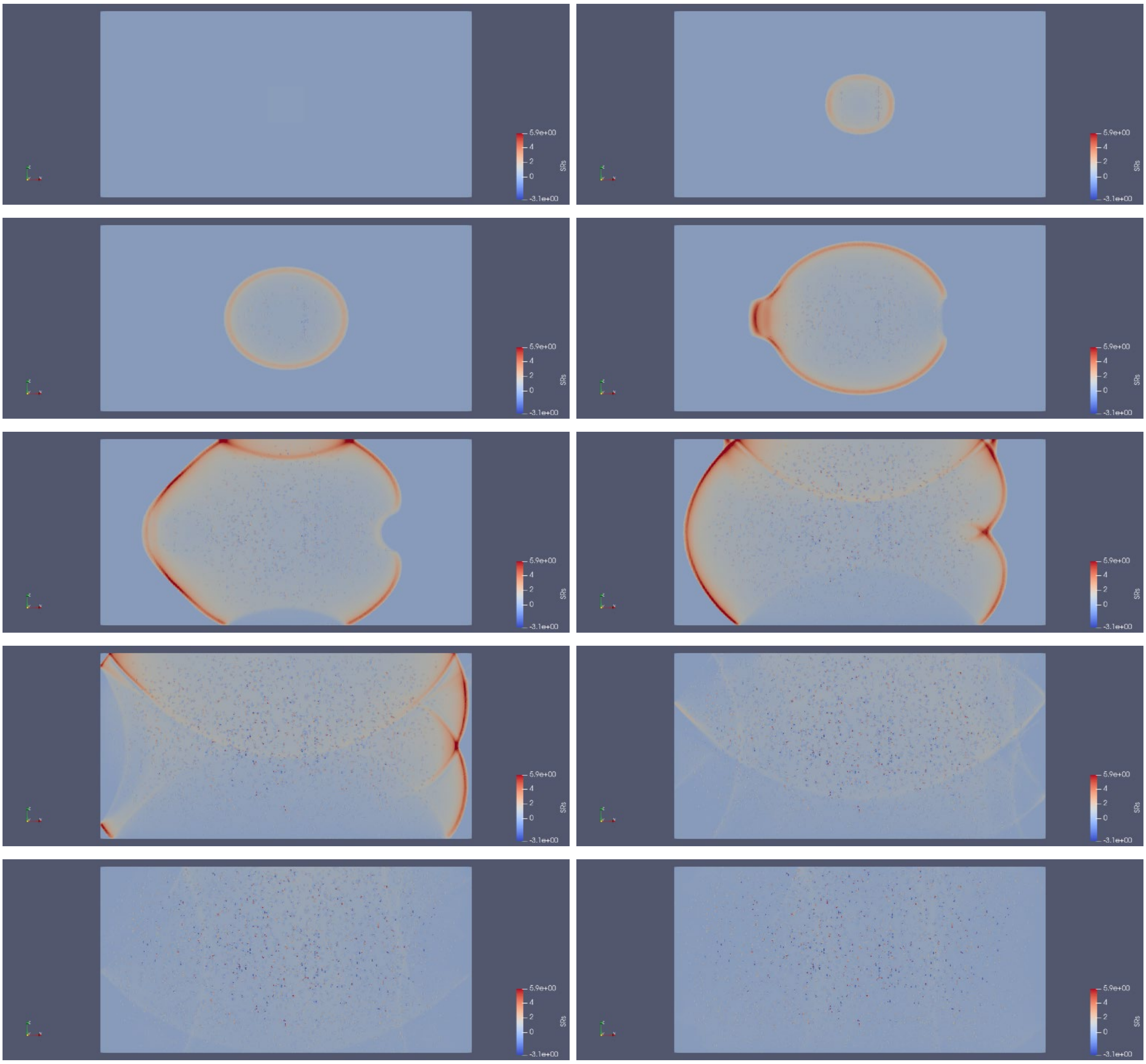


Figure 16. Series of snapshots of slip rates in the strike direction related to the fault rupture.

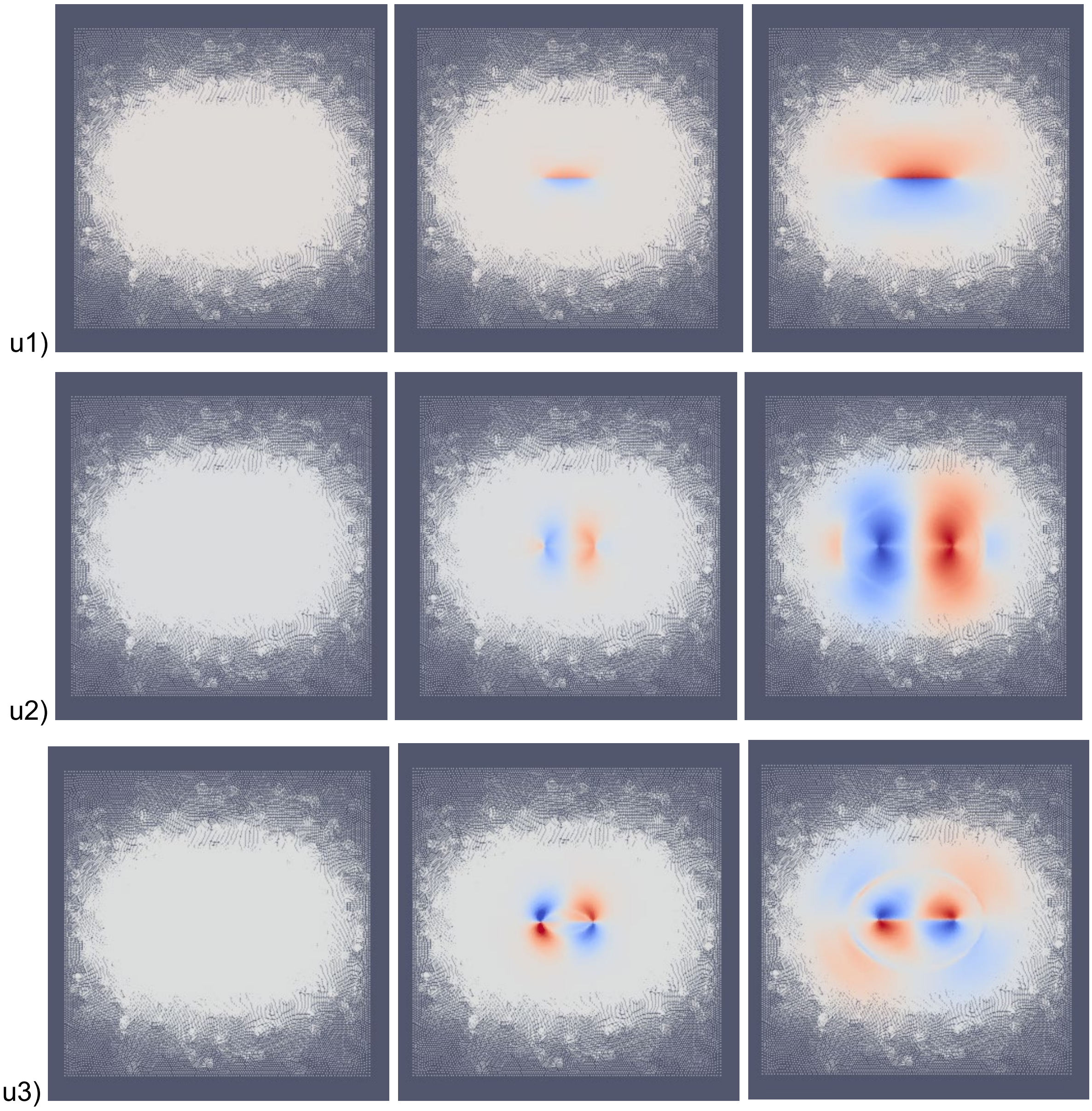
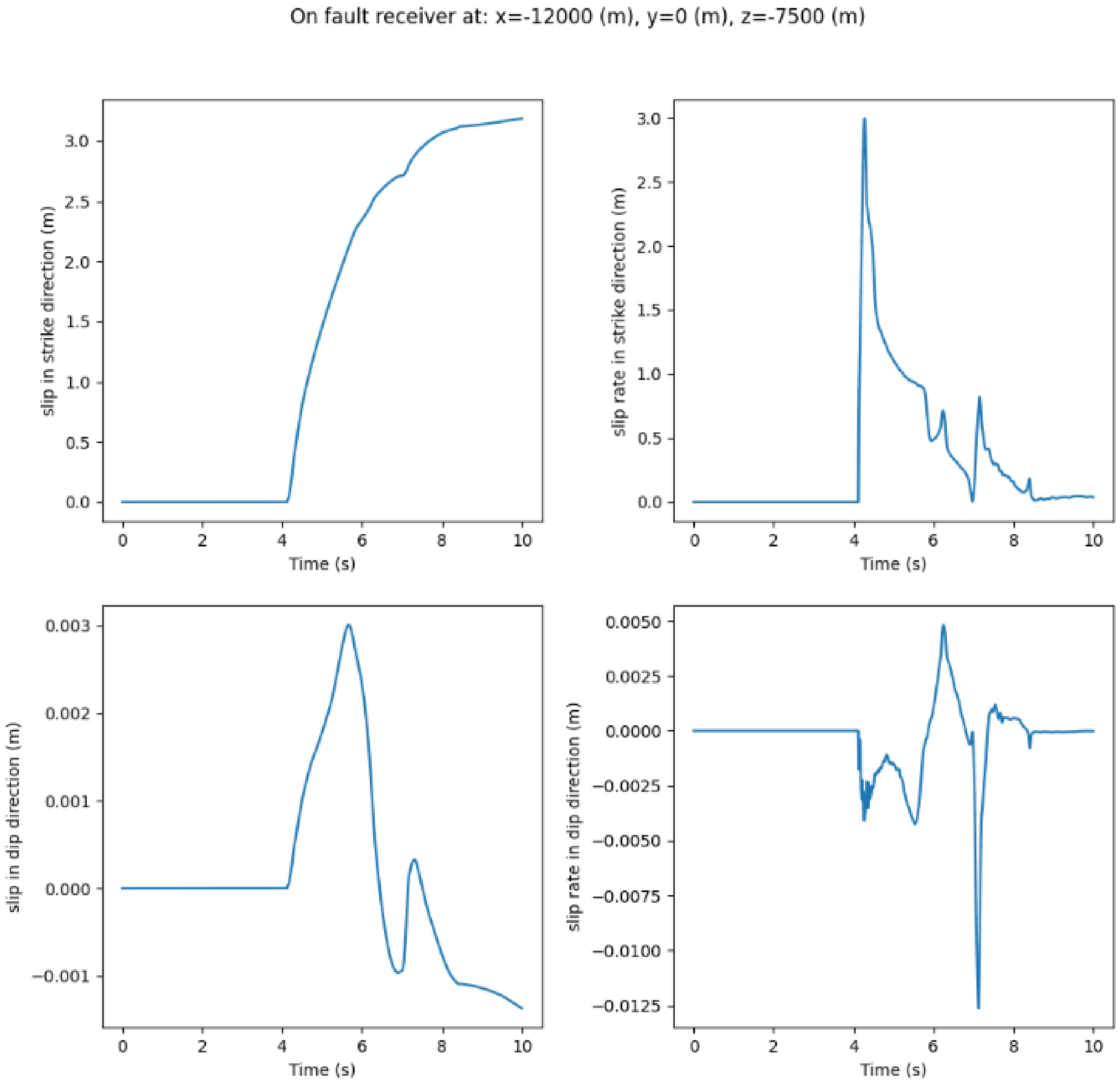


Figure 17. Displacement of ground in strike direction (u1), normal direction (u2) and dip direction (u3).

In terms or quantitative results, SeisSol recorded slip rates, slip and many other variables from on fault receivers while off fault receivers provided velocity of ground motion and stress state. The displacement was integrated with Scipy. Herein we show results from on fault locations of point #1 (-12000; 0; -7500) [2] and # 2 (4500; 0; 0) and off-fault location of point #3 (12000; 3000; 0). The results are shown in Figure 18 and Figure 19.

---

[2] In SeisSol the positive Z is upwards while in description of TPV5 it is downwards


On fault receiver at: x=-12000 (m), y=0 (m), z=-7500 (m)
slip in strike direction (m)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
slip rate in strike direction (m)
0.0
0.5
1.0
1.5
2.0
2.5
3.0
slip in dip direction (m)
−0.001
0.000
0.001
0.002
0.003
slip rate in dip direction (m)
−0.0125
−0.0100
−0.0075
−0.0050
−0.0025
0.0000
0.0025
0.0050
0
2
4
6
8
10
Time (s)

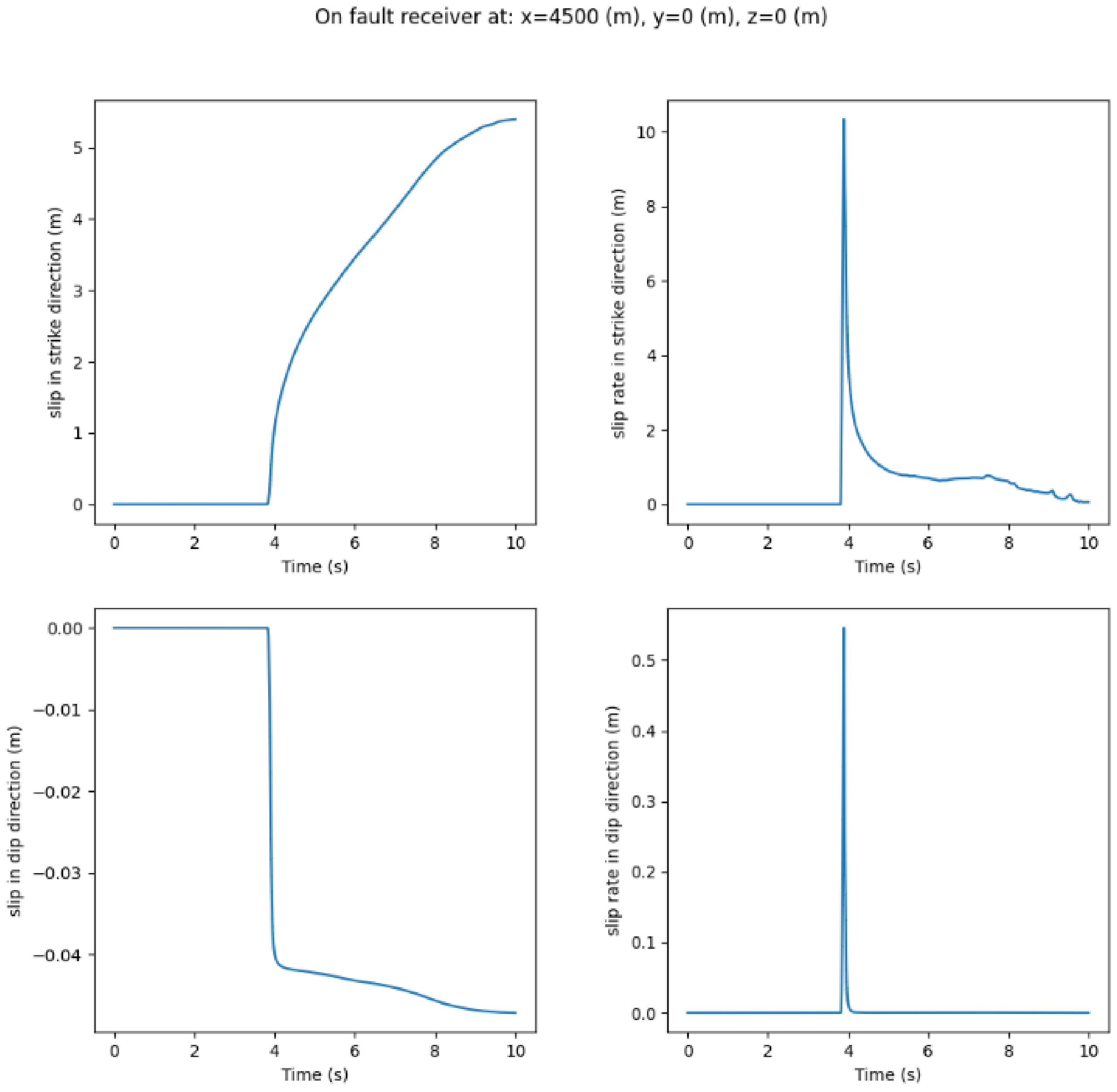


Figure 18. Results from on fault receivers. Plots show slip and slip rate in strike and dip directions.

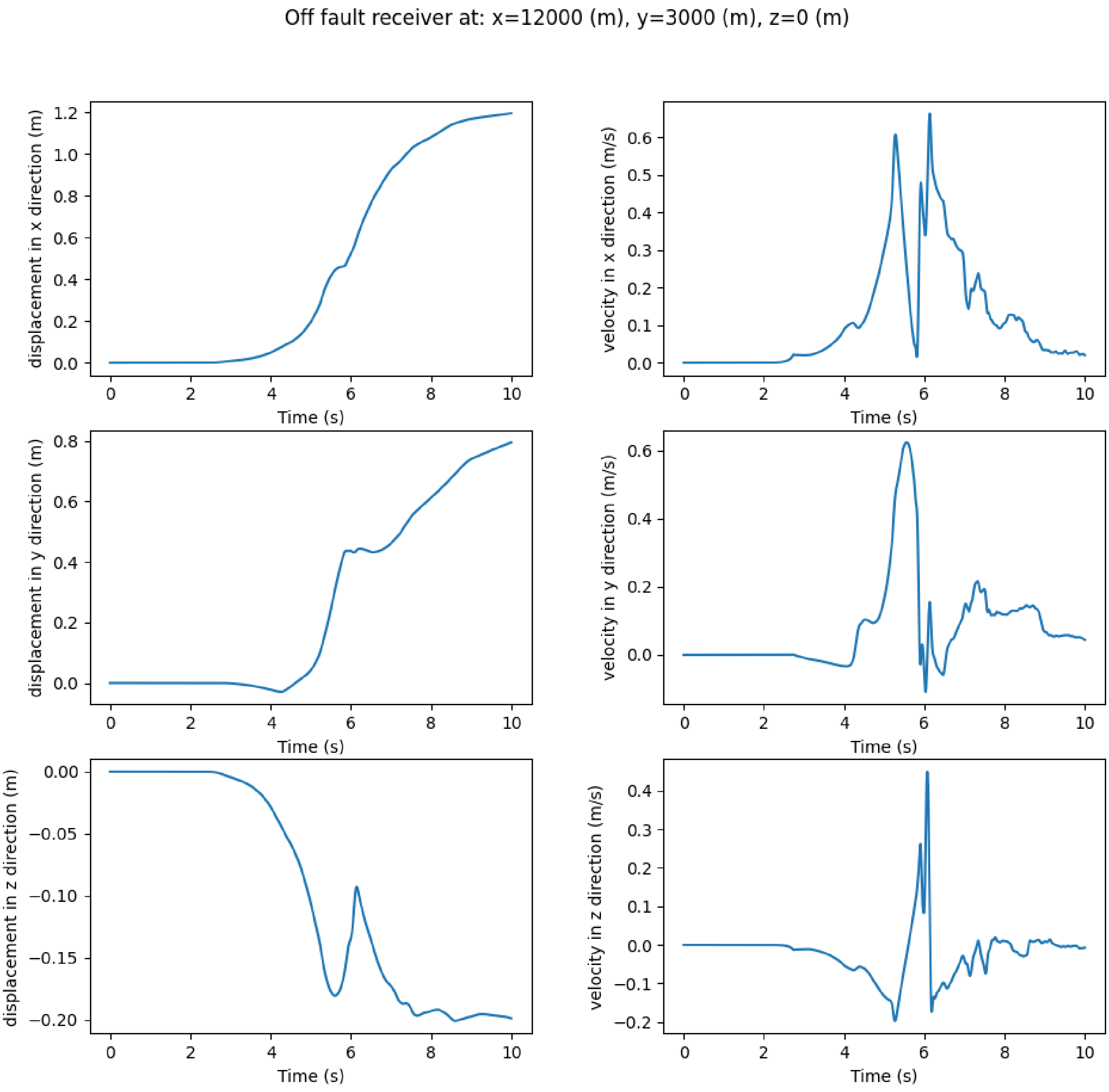


Figure 19. Ground motions, displacement, and velocity in strike (x), normal (y) and dip (z) directions.

7. Discussion and conclusions

In FLAC3D, the computation time of the dynamic simulation was significantly decreased from 3DEC simulations. Utilizing the initially prestressed model, simulating several dynamic runs with varying strength parameters would be feasible in short time. Short simulation times however required decreasing mesh density which caused numerical oscillations in the model. The numerical noise could be decreased by using uniformly meshed model, where element size differences would not affect the travel paths of seismic waves at any frequencies. However, such dense model would significantly increase computation time on the other hand. The FLAC3D code has in-built additional damping options such as Rayleigh, local and combined damping which are originally purposed to model anelastic attenuation of material.

The response of the simulation with FLAC3D corresponds rather well with the ones reported in the exercise. The vertical displacement components at fault differed relatively more from the TVP5 benchmark results than the horizontal components (Figure 13). However, the difference is insignificant if regarding the absolute values of the displacements, as the horizontal components are orders of magnitudes larger than the vertical ones. We preliminarily attribute this behaviour to issues with numerical precision of the model, which appear in these near-zero quantities, and keep investigating the exact source. It was noticed also that some numerical oscillations were generated due to the degrading element size of the model. The oscillations are caused by the fact that seismic waves at higher frequencies are not able to propagate into larger elements due to the inertia and dimensions of the larger elements. Thus, the energy at high frequencies is reflected from the element boundaries within the model. The numerical noise is seen especially at the off-fault monitoring point, where the element size is larger than the 100m-size used closer to the fault.

SeisSol is versatile solver, which has formulation providing great scalability in terms of computer cores. TPV5 model had almost two million elements resulting simulation time of two days and eleven hours. As a reference NPP model in Jussila (2016) had only ~151 200 elements. In this perspective SeisSol's performance is very good. Number of elements and their density can be adjusted to reduce simulation wall clock time if needed. SeisSol is still developing phase, and provides install instructions only for Linux based operating system, which can be a disadvantage. Most of the SeisSol dependencies have Windows versions.

Once the installation process is successful running model with SeisSol is intuitive, although input parameters are given with text files. YAML scripting language provides great flexibility for complex models. Generating geometrical models is fairly easy but it requires additional steps until the model interpretable for SeisSol. Simulations can be automized, however user must write a small program to do it. In Jussila et al. (2021) all simulations were conducted in one session with a Python script. Similar approach could be used with SeisSol.

The usage of SeisSol requires programming skills and may require slightly deeper understanding of operating systems than average users. Knowledge of mechanics of earthquakes and faulting is mandatory and experience in vibrational mechanics, signal processing and measurements with accelerometer or seismometer is significant asset. Once these requirements are met SeisSol is valuable package for simulating fault ruptures and ground motions.

Regarding the objectives of the study:

- Both simulation methodologies used in this study are faster from the computational point of view, and they are both calculating comparable results with the TVP5 benchmark case used.
- In terms of agility (i.e. learning time, preparation of inputs, extraction of results etc.) they present advantages over the use of 3DEC (Table 5). FLAC3D is a general-purpose software tool, and more experts are familiar with it. It is also a commercial tool, with well-structured documentation and support. Hence, the learning time for a new user can be expected to be faster. The advantage of SeisSol is that it is a freeware with an active user community. The flexibility of scripting interfaces

is also an asset. However, SeisSol is a specialised software, which means that the user must be a dedicated expert. In this respect SeisSol is well suited for more scientific studies.

Table 5 summarises the characteristics of the different modelling paths. The costs of the commercial software options are high, but not unmatchable considering that FLAC3D is a multipurpose toll. The analysis time of the benchmarked model with FLAC3D was shorter, but the computer running SeisSol had much worse performance. It can be concluded that the project achieved its goal of identifying suitable, accurate, and agile modelling methods to replace 3DEC for fault rupture simulations.

The next step is to clarify which modelling path is more suited for the future project requirements in Finland. Are the near-term computational needs sufficient a specialized software like SeisSol, and train a few experts in its use? Or are the computational needs periodical, in which case, the use of general purpose commercial software could be advantageous.

Table 7. Comparisons of the examined modelling techniques.

Features of the technique	3DEC	FLAC3D	SeisSol
Modelling of the fault rupture	Ok	Ok	Ok
Ground motions (GM)	Ok	Ok	Ok
High frequency (e.g., 25Hz)	Probably too time expensive	Yes	Yes
License fees (kEuro)	11.5 k€ (annual lease + dynamic option)	8.5 k€ (annual lease + dynamic option)	0 k€
Estimated learning time for new user (days)			
Support	Yes, Itasca support	Yes, Itasca support	Yes, developers, SeisSol forum in GitHub
Parallel computing	Yes, CPU	Yes, CPU	Yes, CPU or GPU
Operating system	Windows 10, 11, Linux Ubuntu 20.04	Windows 10, 11, Linux Ubuntu 20.04	Tested with Ubuntu 22.04 LTS
Minimum system requirements (CPU, RAM, etc)	No. ITASCA recommends single physical processor with high core count (e.g., 32), multi-threaded 64Gb RAM (e.g., 4x16Gb), both Intel and AMD CPU	No. ITASCA recommends single physical processor with high core count (e.g., 32), multi-threaded 64Gb RAM (e.g., 4x16Gb), both Intel and AMD CPU	Intel, 6 performance cores, RAM 64Gb
Run time of the models in this report		40 min	2 days, 11 hours and 22 minutes
Preparation time of models in this report	Couple of hours	Couple of hours	Couple of hours
Plug-in interface for automating runs	Automatic runs possible if scripted in the model	Automatic runs possible if scripted in the model	No plug-in but automatic runs are possible for instance with Python
Number of elements		5 187 322	1 947 942

References

Blanksma, D., J. Hazzard, B. Damjanac, T. Lam, and H.A. Kasani, Effect of fault reactivation on deformation of off-fault fractures near a generic deep geological repository in crystalline rock in Canada. J Seismol (2022). https://doi.org/10.1007/s10950-022-10096-7

SCEC, 2009, https://strike.scec.org/cvws/tpv5docs.html (Accessed: 07.03.2023).

Darcel, C., Dedecker, F., Emam, S., Tran, M-H., 2019. Numerical Investigation of Earthquake Rupture and Off-Fault Fracture Response with a Coupled PFC3D/FLAC3D Environment. Working report WR 2019-03, Posiva Oy, Eurajoki.

Fälth, B., and H. Hökmark, 2011, Modelling End-Glacial Earthquakes at Olkiluoto, Working Report 2011–13, Posiva Oy.

Fälth, B., H. Hökmark, B. Lund, P. M. Mai, R. Roberts, and R. Munier, 2015, Simulating Earthquake Rupture and Off-Fault Fracture Response: Application to the Safety Assessment of the Swedish Nuclear Waste Repository, Bull. Seism. Soc. of Am., 105, no. 1, 134–151, doi: 10.1785/0120140090.

FLAC3D, (n.d.). https://www.itascacg.com/software/flac3d (accessed March 17, 2023).

Fülöp, L., V. Jussila, B. Lund, B. Fälth, P. Voss, J. Puttonen, J. Saari, 2016, Modelling as a tool to augment ground motion data in regions of diffuse seismicity - Progress 2015, NKS Nordic Nuclear Safety Research.

Fülöp, L., V. Jussila, B. Lund, B. Fälth, P. Voss, J. Puttonen, J. Saari, P. Heikkinen, 2017, Modelling as a tool to augment ground motion data in regions of diffuse seismicity - Final report, NKS Nordic Nuclear Safety Research, Roskilde, Denmark.

Fülöp, L., V. Jussila, B. Fälth, P. Voss, B. Lund, 2019, Synthetic ground motions to support the Fennoscandian GMPEs, NKS Nordic Nuclear Safety Research, Roskilde, Denmark, 2019.

Fülöp, L., V. Jussila, R. Aapasuo, T. Vuorinen, P. Mäntyniemi, 2020, A Ground-Motion Prediction Equation for Fennoscandian Nuclear Installations, Bull. Seism. Soc. of Am. 110 (2020) 1211–1230. https://doi.org/10.1785/0120190230.

Fülöp, L., P. Mäntyniemi, M. Malm, G. Toro, M. J. Crespo, T. Schmitt, S. Burck, and P. Välikangas, 2022, Probabilistic seismic hazard analysis in low-seismicity regions: an investigation of sensitivity with a focus on Finland, Nat Hazards, doi: 10.1007/s11069-022-05666-4.

Griddle, (n.d.). https://www.itascacg.com/software/Griddle (accessed March 17, 2023).

Harris, R.A., M. Barall, R. Archuleta, B. Aagaard, J.-P. Ampuero, H. Bhat, V. Cruz-Atienza, L. Dalguer, P. Dawson, S. Day, B. Duan, E. Dunham, G. Ely, Y. Kaneko, Y. Kase, N. Lapusta, Y. Liu, S. Ma, D. Oglesby, K. Olsen, A. Pitarka, S. Song, and E. Templeton, 2009, The SCEC/USGS Dynamic Earthquake Rupture Code Verification Exercise, Seis. Res. Lett., vol. 80, no. 1, pages 119-126, doi:10.1785/gssrl.80.1.119.

Harris, R. A. et al., 2018, A Suite of Exercises for Verifying Dynamic Earthquake Rupture Codes, Seism. Res. Lett., 89, no. 3, 1146–1162, doi: 10.1785/0220170222.

Hökmark, H., B. Fälth, M. Lönnqvist, and R. Munier, 2019, Earthquake simulations performed to assess the long-term safety of a KBS-3 repository - Overview and evaluation of results produced after SR-Site, Technical Report TR-19-19, Swedish Nuclear Fuel and Waste Management Co., Solna, 104 p.

IAEA, 2018. Best practices in physics-based fault rupture models for seismic hazard assessment of nuclear installations, International Atomic Energy Agency, Wien, Austria.

Igel, H., 2017, Computational Seismology a practical Introduction. Oxford University Press, ISBN 978–0–19–871740–9.

Itasca Consulting Group, Inc., (n.d.). https://www.itascacg.com/ (accessed March 17, 2023).

Jussila, V., Y. Li, L. Fülöp, 2016 Statistical analysis of the variation of floor vibrations in nuclear power plants subject to seismic loads, Nuclear Engineering and Design. 309 84-96. doi: https://doi.org/10.1016/j.nucengdes.2016.09.005

Jussila, V., B. Fälth, P. Mäntyniemi, P.H. Voss, B. Lund, L. Fülöp, 2021, Application of a Hybrid Modeling Method for Generating Synthetic Ground Motions in Fennoscandia, Northern Europe, Bull. Seism. Soc. of Am. https://doi.org/10.1785/0120210081.

Leonard, M., 2010, Earthquake fault scaling: Self-consistent relating of rupture length, width, average displacement, and moment release, Bull. Seism. Soc. of Am. 100 1971-1988. doi: 10.1785/0120090189

Leonard, M., 2014, Self-consistent earthquake fault-scaling relations: Update and extension to stable continental strike-slip faults. Bull. Seism. Soc. of Am., 104, 2953-2965.

Ojala, A., J. Mattila, T. Ruskeeniemi, M. Markovaara-Koivisto, J.-P. Palmu, N. Nordbäck, A. Lindberg, I. Aaltonen, J. Savunen, and R. Sutinen, 2019, Postglacial faults in Finland – a review of PGSdyn – project results, 2019–1, Posiva Oy, Eurajoki.

PFC, (n.d.). https://www.itascacg.com/software/pfc (accessed March 17, 2023).POSIVA, (2019), Co-Seismic Secondary Fracture Displacements Under Different Stress Conditions, Fälth, B., Lönnqvist, M., Hökmark, H., Working report 2019-10, 2019 SeisSol, 2023, https://seissol.readthedocs.io/en/latest/index.html.

SeisSol, (n.d.). https://seissol.org/ (accessed March 17, 2023).

SKB, (2011) Long-term safety for the final repository for spent nuclear fuel at Forsmark. Main report of the SR-Site project. SKB TR-11−01. Svensk Kärnbränslehantering AB (SKB), Stockholm, SwedenVarpasuo, P., J. Saari, and Y. Nikkari, 2001, Seismic Hazard and Ground Motion for Leningrad NPP site, in Transactions of the 16th International Conference on Structural Mechanics in Reactor Technology (SMiRT16), August 12-17, 2001, Washington DC, USA.

3DEC, (n.d.). https://www.itascacg.com/software/3dec (accessed March 17, 2023).

DocuSign Envelope ID: F90AD6A6-9F1D-411F-82E4-0B573C18F616

Appendix A

Installing SeisSol

A finite element model consists of geometry, mesh, material model and solver. Finite element analysis suite (FEA) typically includes a tool for each of the features required for a complete model. Typically, FEA is best in solving the problem while other software is better for creating a geometry and meshing. A user controls the workflow by using graphical user interface. Such a software is typically proprietary and favored by industry. They are easy to install with few clicks usually requiring installation path and license server. Technical support is typically provided, and annual fees are starting from 10 k€. Usually, proprietary software has specific task to do, and their capabilities cannot be extended by a user. Abaqus is an exception to this rule.

Open-source software are typically more flexible than proprietary versions, but installation may be overwhelming and technical support is typically located on forums such as Stack Overflow. Lack of graphical user interface is not uncommon implying that parameters of the model are given with input text files and commands are executed in command-line.

SeisSol is a solver, which saves analysis results to ASCII or binary files. Creation of a geometry and meshing are the main tasks in pre-processing, which could be done with proprietary or open-source software. Herein, we focus on open-source version. Similar approach applies to analysis of the result. Often open-source software could be compiled by a user or install developer compiled version of it. The SeisSol development team does not provide latter, and the guideline of the installation is somewhat incomplete, therefore the installation process is described herein in detail.

Requirements for computer

SeisSol is very demanding software, which scales from powerful laptops to super-computers. In this project SeisSol was installed to Lenovo's Thinkpad X1 Extreme 3rd generation. It has Intel Core i7-10750H processor, 64Gb RAM and 2T hard-drive. The operating system was Ubuntu 22.04 LTS. This configuration could be considered bare minimum for SeisSol.

In theory SeisSol could be installed to Windows or Mac but in practice it is most convenient install to Debian GNU/Linux or its derivatives such as Ubuntu. Currently SeisSol supports only Intel's processors, which are listed in Table 1. Prior installation user must check computers core architecture, since it is required information later in the installation.

Table 8. Supported core architectures and their short names for installation.

Westmere	wsm
Sandy Bridge	snb
Knights Corner (Xeon Phi)	knc
Haswell	hsw
Knight Landing (Xeon Phi)	knl

Installation prerequisites

Ubuntu offers several methods to install applications. A popular method is Advanced Package Tool (apt) used on the command-line. Users preferring graphical user interface should install Synaptic Package Manager, which is a front-end to apt. However, major portions of commands for the installation of SeisSol is given on command-line.

SeisSol dependencies can be categorized based on installation method. The first group of libraries were installed on system level with Synaptic Package Manager, which are Git, GMSH, virtualenv, gcc (C), g++ (C++), gfortran (FORTRAN) and OpenMPI. Header files of C, C++, FORTRAN and OpenMPI are mandatory for the installation of SeisSol but they are not included to the standard libraries. Required header files are in development files, which can be identified with suffix -dev. The second group of libraries are installed locally on command-line. They are CMAKE, SCONS, HDF5, NETCDF, LIBXSMM, PSPAMM, GEMMFORGE, PARMETIS, YAML-CPP, IMPALAJIT, EASI, EIGEN, PUMI and PUMI. CMAKE and HDF5 could be installed globally with Synaptic Package Manager, but we follow suggestions of SeisSol development team by installing them locally for SeisSol. Python is only dependency, which does not require installation since it is included to the Ubuntu, however, it is convenient create a virtual environment isolating SeisSol's Python dependency from the rest of the operating system.

Installing virtual environment

The virtual environment for Python can created with virtualenv. Easiest method for installation is usage of Synaptic Package Manager. The syntax indicating terminal command begins with $ character and followed by white space. These characters are omitted when installation commands are given.

Next open terminal (ctrl+alt+T). Current path is $HOME. Following command will install virtual environment for Python to $HOME.

```
$ virtualenv -p python3 pyseissol
```

If user wish installed to another location then the format of command would be following:

```
$ virtualenv -p python3 path/pyseissol
```

To launch Python in virtual environment following text files with gedit or alternative text editor should be created. Create empty text document, and copy following to it:

```
#!/bin/sh
export VIRTUAL_ENV="$HOME/pyseissol"
export PATH="$VIRTUAL_ENV/bin:$PATH"
unset PYTHON_HOME
echo "Use 'exit' command to close down virtual environment"
python –version
cd $HOME
exec "${@:-$SHELL}"
```

Name it as "launch_python", and move it to pyseissol/bin/. Remember change the VIRTUAL_ENV path if pyseissol was installed other location than $HOME. In order to add pyseissol to applications list a desktop file is created as follows:

```
        [Desktop Entry]
        Version=1.0
        Name=launch
        Exec=bash -c 'cd $HOME/pyseissol/bin; ./launch_python;$SHELL'
        Icon=/usr/share/pixmaps/python3.xpm
        Terminal=true
        Type=Application
        Categories=Application
        Name[en_US]=Python
```

Name it as “launch_python.desktop”, and assuming the current path in the terminal session is equal to location of “launch_python.desktop” then give following command:

```
        $ sudo cp launch_python.desktop /usr/share/applications
```

The Python icon appears to the application launcher. All previous tasks were given in the general terminal session, which can be closed. Following commands are provided in a terminal session, which applies just created Python virtual environment, which implies locally installed libraries.

Installing first group dependencies

Synaptic Package Manager is easy and intuitive to use, therefore we do not provide additional instructions for installing Git, GMSH, virtualenv, gcc, g++, gfortran and OpenMPI. Make sure following development libraries libgcc-XX-dev, libstdc++-XX-dev, libgfortran-XX-dev and libopenmpi-dev are installed as well. Capital letters XX implies version of the library.

Installing second group dependencies

Prior installation user must edit temporarily environmental variables by opening bashrc with gedit. The command is given in a terminal session, which applies Python virtual environment. Thus, click Python icon in applications, and type following command:

```
        $ gedit ~/.bashrc
```

Add following lines in the end of bashrc:

```
        export PATH=$HOME/App/SeisSol/bin:$PATH
        export LIBRARY_PATH=$HOME/App/SeisSol/lib:$LIBRARY_PATH
        export LD_LIBRARY_PATH=$HOME/App/SeisSol/lib:$LD_LIBRARY_PATH
        export
PKG_CONFIG_PATH=$HOME/App/SeisSol/lib/pkgconfig:$PKG_CONFIG_PATH
        export CMAKE_PREFIX_PATH=$HOME/App/SeisSol
        export CPATH=$HOME/App/SeisSol/include:$CPATH
        export
CPLUS_INCLUDE_PATH=$HOME/App/SeisSol/include:$CPLUS_INCLUDE_PATH
        export C_INCLUDE_PATH=$HOME/App/SeisSol/include:$C_INCLUDE_PATH
```

Next load the temporary environmental variables with command:

```
        $ source ~/.bashrc
```

Now the terminal session is ready for installation of SeisSol. Make sure that current path is $HOME. If not, then type:

```
        $ cd
```

Next, we create two folders, which are App and SeisSol. It can be done in Files or typing:

```
        $ mkdir App && cd App && mkdir SeisSol && cd SeisSol
```

The current path is $HOME/App/SeisSol. If user wish install SeisSol into other location path related variables are to be change accordingly. Then several dependencies are installed prior downloading SeisSol with Git. We begin with creating folders bin, lib and include.

```
        $ mkdir bin && mkdir lib && mkdir include
```

In the installation process we used wget to download files from specified url, and git clone to clone a git repository to computer.

CMAKE is used to control compilation process of SeisSol and its dependencies. It is independent of operating system. The downloaded version of CMAKE includes compiled binaries, which is retrieved with a command:

```
        $ wget
https://github.com/Kitware/CMake/releases/download/v3.21.0/cmake-3.21.0-
linux-x86_64.tar.gz && mkdir cmake && tar -xvf cmake-3.21.0-linux-
x86_64.tar.gz -C cmake --strip-components 1
```

A symbolic link is created to SeisSol/bin otherwise CMAKE binaries are not found.

```
        $ ln -s ${HOME}/App/SeisSol/cmake/bin/cmake
${HOME}/App/SeisSol/bin/cmake
```

HDF5 is software for storing data. It is installed with following commands:

```
        $ wget https://support.hdfgroup.org/ftp/HDF5/releases/hdf5-1.12/hdf5-
1.12.2/src/hdf5-1.12.2.tar.bz2
$ tar -xaf hdf5-1.12.2.tar.bz2

        $ cd hdf5-1.12.2

        $ CPPFLAGS="-fPIC ${CPPFLAGS}" CC=mpicc FC=mpif90 ./configure --
enable-parallel --prefix=$HOME/App/SeisSol --with-zlib --disable-shared --
enable-fortran

        $ make -j8

        $ make install

        $ cd ..
```

NETCDF is set of software libraries and data format for creating, accessing and sharing of array-oriented data.

```
       $ wget https://github.com/Unidata/netcdf-
c/archive/refs/tags/v4.9.0.tar.gz
$ tar -xaf v4.9.0.tar.gz
       $ cd netcdf-c-4.9.0
       $ CFLAGS="-fPIC ${CFLAGS}" CC=h5pcc ./configure --enable-shared=no
--prefix=$HOME/App/SeisSol --disable-dap
       $ make check
       $ make -j8
       $ make install
       $ cd ..
```

LIBXSMM is a library for dense and sparse matrix operations. Use following commands for installation:

```
       $ git clone --branch 1.17 https://github.com/hfp/libxsmm
       $ cd libxsmm
       $ make generator
       $ cp bin/libxsmm_gemm_generator $HOME/App/SeisSol/bin
       $ cd ..
```

PSPAMM is a library for sparse matrix multiplication.

```
       $ git clone https://github.com/peterwauligmann/PSpaMM.git
       $ ln -s $(pwd)/PSpaMM/pspamm.py $HOME/App/SeisSol/bin
```

GEMMFORGE is a library for general matrix multiplication utilizing graphics processing unit (GPU). We installed it to test GPU version of SeisSol, however the installation failed. The library is optional, and not required in TPV5.

```
       $ pip install git+https://github.com/ravil-mobile/gemmforge.git
```

PARMETIS is an MPI-based library for partitioning graphs, partitioning finite element meshes, and producing fill reducing orderings for sparse matrices.

```
       $ wget https://ftp.mcs.anl.gov/pub/pdetools/spack-pkgs/parmetis-
4.0.3.tar.gz
       $ tar -xvf parmetis-4.0.3.tar.gz
       $ cd   parmetis-4.0.3
```

In December 2022 metis was shipped with parmetis. With following commands it is configured:

Go to the folder ...parmetis-4.0.3 /metis/include, and open header file metis.h, and change values of IDXTYPEWIDTH and REALTYPEWIDTH on lines 33 and 43 to 64.

Now it is time to install ParMetis:

```
        $ make config cc=mpicc cxx=mpic++ prefix=$HOME/App/SeisSol
        $ make install
        $ cp build/Linux-x86_64/libmetis/libmetis.a $HOME/App/SeisSol/lib
        $ cp metis/include/metis.h $HOME/App/SeisSol/include
        $ cd ..
```

Now the SeisSol is downloaded, and several sub-modules installed prior installation of SeisSol.

```
        $ git clone https://github.com/SeisSol/SeisSol.git
        $ cd SeisSol
        $ git submodule update --init
        $ cd submodules
```

YAML-CPP is parser and emitter written in C++ of YAML 1.2 data language. It’s primarily application is inputting for instance fault’s initial stress state to SeisSol.

```
        $ git clone https://github.com/jbeder/yaml-cpp.git
        $ cd yaml-cpp
        $ git checkout yaml-cpp-0.6.3
        $ mkdir build && cd build
        $ cmake ..  -DCMAKE_INSTALL_PREFIX=$HOME/App/SeisSol
        $ make -j4
        $ make install
        $ cd ../..
```

IMPALAJIT is a JIT compiler for accessing data in simulation applications.

```
        $ git clone https://github.com/Manuel1605/ImpalaJIT.git
        $ cd ImpalaJIT
        $ mkdir build && cd build
        $ cmake ..  -DCMAKE_INSTALL_PREFIX=$HOME/App/SeisSol
        $ make -j4
        $ make install
        $ cd ../..
```

EASI is a library for initializing models.

```
        $ git clone https://github.com/SeisSol/easi.git
        $ cd easi
        $ mkdir build && cd build
        $ cmake .. -DCMAKE_PREFIX_PATH=$HOME/App/SeisSol/SeisSol/submodules –
DCMAKE_INSTALL_PREFIX=$HOME/App/SeisSol  -DASAGI=OFF -DIMPALAJIT=ON $easi
```

```
       $ make -j4 install
       $ cd ../..
```

EIGEN is a template library written in C++ for linear algebra.

```
       $ wget https://gitlab.com/libeigen/eigen/-/archive/3.4.0/eigen-
3.4.0.tar.gz
       $ tar -xf eigen-3.4.0.tar.gz
       $ cd eigen-3.4.0
       $ mkdir build && cd build
       $ cmake .. -DCMAKE_INSTALL_PREFIX=$HOME/App/SeisSol
       $ make install
       $ cd ../..
```

SeisSol

Prior installation of SeisSol the core architecture is to be defined. Supported versions are given in Table 2. Intel Core i7-10750H processor uses Haswell architecture, which is given with option -DHOST_ARCH=hsw. Once the core architecture is set, we can continue to SeisSol installation with following command:

```
       $ mkdir build-release && cd build-release
       $ CC=mpicc CXX=mpic++ FC=mpif90
CMAKE_PREFIX_PATH=$HOME/App/Seissol:$CMAKE_PREFIX_PATH
PKG_CONFIG_PATH=$HOME/App/Seissol/lib/pkgconfig/:$PKG_CONFIG_PATH cmake -
DNETCDF=ON -DMETIS=ON -DCOMMTHREAD=ON -DASAGI=OFF -DHDF5=ON -
DCMAKE_BUILD_TYPE=Release -DTESTING=OFF  -DLOG_LEVEL=warning -
DLOG_LEVEL_MASTER=info -DHOST_ARCH=hsw -DPRECISION=double ..
       $ make -j48
       $ cd .. / ..
```

Next, we will create a folder where SeisSol simulations are launched.

```
       $ mkdir launch_SeisSol
       $ cd launch_SeisSol
       $ cp $HOME/App/SeisSol/SeisSol/build-release/SeisSol*
$HOME/App/SeisSol/launch_SeisSol/
```

PUMI is a parallel unstructured mesh infrastructure API.

```
       $ git clone https://github.com/SCOREC/core.git
       $ cd core
       $ git submodule update --init
       $ mkdir build && cd build
       $ cmake .. \
         -DCMAKE_C_COMPILER="mpicc" \
         -DCMAKE_CXX_COMPILER="mpicxx" \
         -DCMAKE_C_FLAGS="-O2 -g -Wall" \
         -DCMAKE_CXX_FLAGS="-O2 -g -Wall" \
         -DCMAKE_INSTALL_PREFIX=$HOME/App/SeisSol
       $ make -j $ncpu
       $ make install
       $ cd ../..
```

PUMGen is a library for generating mesh for SeisSol.

```
        $ git clone https://github.com/SeisSol/PUMGen.git
        $ cd PUMGen
        $ git submodule update --init
        $ mkdir build && cd build
        $ cmake .. -DCMAKE_INSTALL_PREFIX="$HOME/App/SeisSol" -DSIMMETRIX=OFF
\
            -DCMAKE_BUILD_TYPE=Release\
            -DCMAKE_C_COMPILER="mpicc" -DCMAKE_CXX_COMPILER="mpicxx"
        $ make -j $ncpu
        $ make install
        $ cd ../..
```

Should problems occur

The most typical errors are result of incorrect paths, missing dependency, invalid command or incorrect order of installation of dependencies. The incorrect path is the most common, and invalid command is most obvious in the installation of SeisSol since user must define core architecture. If it is not correct the installation fails since the instruction provided by SeisSol cannot be interpreted correctly by core. Another pitfall is installing ParMetis. There are two sources Karypislab and the path provided by SeisSol team for the package. The version provided by Karypislab can be installed but then SeisSol fails. Last possible source for the errors are the development of SeisSol. It simply may contain bugs. They are discussed in the Github page of SeisSol and in the SeisSol documentation. Should problems occur read error log carefully since they are providing valuable hints what went wrong.

We have tested installation with instructions given herein several times. Despite of it, the instructions may contain typos or the installation of SeisSol has changed.

Glossary

Table 9. List of dependencies and software utilized in TPV5 simulation.

RAM, random access memory	Memory in computer for storing working data and machine code
SWAP	A partition on hard drive used as virtual memory when a system runs out of physical memory
CPU, central processing unit	Performs arithmetic, logic, controlling and input/output instructed by a program
Ubuntu	Open-source Debian based Linux distribution
gedit	Text editor
apt	Advanced package tool, the main command-line package manager
Synaptic package manager	Package management software using apt.
Python	High level interprets programming language
virtualenv	Python virtual environment creator
pip	Package installer for Python
gmsh	A three-dimensional finite element mesh generator with built-in pre- and post-processing facilities
Paraview	Open-source post-processing visualization engine
git	Open-source distributed version control system
CMAKE	Open source and platform independent tool for building, testing and packing software
SCONS	Alternative option to CMAKE for constructing software
OPENMPI	Open-source message passing interface. It can be used for distributing parallel computational tasks to CPUs.
gcc	GNU C compiler
gfortan	GNU fortran compiler
g++	GNU C++ compiler
HDF5	Open-source file format supporting large, complex and heterogeneous data
NETCDF	Network common data form, set of software libraries and data format for creating, accessing and sharing of array-oriented data
LIBXSMM	Library for dense and sparse matrix operations
PSPAMM	A library for sparse matrix multiplication
GEMMFORGE	A library for general matrix multiplication (SeisSol with GPU capabilities)

RAM, random access memory	Memory in computer for storing working data and machine code
PARMETIS	An MPI-based library for partitioning graphs, partitioning finite element meshes, and producing fill reducing orderings for sparse matrices.
YAML-CPP	A parser and emitter of input data
IMPALAJIT	A JIT compiler for accessing data
EASI	A library for initializing models
EIGEN	A template library for linear algebra
PUMI	A parallel unstructured mesh infrastructure API
PUMgen	A library for generating mesh for SeisSol

DocuSign

Certificate Of Completion

Envelope Id: F90AD6A69F1D411F82E40B573C18F616		Status: Completed
Subject: Complete with DocuSign: VTT_R_00102_23		
Source Envelope:		
Document Pages: 49	Signatures: 1	Envelope Originator:
Certificate Pages: 1	Initials: 0	Erja Schlesier
AutoNav: Enabled		Vuorimiehentie 3, Espoo,
EnvelopeId Stamping: Enabled		., . P.O Box1000,FI-02044
Time Zone: (UTC+02:00) Helsinki, Kyiv, Riga, Sofia, Tallinn, Vilnius		Erja.Schlesier@vtt.fi
		IP Address: 86.60.173.229

Record Tracking

Status: Original 05 April 2023 | 13:08	Holder: Erja Schlesier Erja.Schlesier@vtt.fi	Location: DocuSign

Signer Events	Signature	Timestamp
Marko Mäkipää Marko.Makipaa@vtt.fi Research Team Leader Teknologian tutkimuskeskus VTT Oy Security Level: Email, Account Authentication (None), Authentication	DocuSigned by: Marko Mäkipää EF95EB85E9804AC... Signature Adoption: Pre-selected Style Using IP Address: 86.115.67.231	Sent: 05 April 2023 | 13:18 Viewed: 06 April 2023 | 11:54 Signed: 06 April 2023 | 12:07

Authentication Details
SMS Auth:
Transaction: 8ac72acd-b9d3-4031-9bae-6293b851c43e
Result: passed
Vendor ID: TeleSign
Type: SMSAuth
Performed: 06 April 2023 | 11:53
Phone: +358 45 6788723

Electronic Record and Signature Disclosure:
Not Offered via DocuSign

In Person Signer Events	Signature	Timestamp

Editor Delivery Events	Status	Timestamp

Agent Delivery Events	Status	Timestamp

Intermediary Delivery Events	Status	Timestamp

Certified Delivery Events	Status	Timestamp

Carbon Copy Events	Status	Timestamp

Witness Events	Signature	Timestamp

Notary Events	Signature	Timestamp

Envelope Summary Events	Status	Timestamps
Envelope Sent	Hashed/Encrypted	05 April 2023 | 13:18
Certified Delivered	Security Checked	06 April 2023 | 11:54
Signing Complete	Security Checked	06 April 2023 | 12:07
Completed	Security Checked	06 April 2023 | 12:07

Payment Events	Status	Timestamps