SeisSol

The ADER-DG Method for Seismic Wave Propagation

USER MANUAL - PARAMETER FILES

Amaryillis Nerger, Christian Pelties & Johannes Klicpera

GEOPHYSICS
DEPARTMENT OF EARTH AND ENVIRONMENT
LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Contents

1 How to use this Manual

The actual manuscript guides through the various options to set up a proper parameter file for a SeisSol simulation. The parameter file must be specified for a correct job submission as

```
mpirun.openmpi -n NRprocs seissolxx PARAMETERS.par
```

that has usually the extension `*.par` by convention. SeisSol reads the file during initialization enabling run-time settings of various model parameters. The routine `/src/Reader/readpar.f90` handles the reading and can be modified for the introduction of further parameters and options. The parameter file must be located in the current working directory or preceded by a link to its location. We use Fortran's namelist approach which is structured by namelist blocks indicated by an ampersand & followed by the namelist's name and ends by a common slash /. In such blocks parameter=option pairs can be set in arbitrary order. Certain parameters require additional information in form of further input files, which may be limited to static order. Default options (as specified later in this document) are opt by leaving the namelist blocks empty or individual parameters are dropped. However, the namelist blocks **Equations, IniCondition, Boundaries, SourceType, SpongeLayer, MeshNml, Discretization, Output, AbortCriteria, Analysis, Debugging**
have be included even if not used as:

```
&IniCondition
/

&SpongeLayer
/

    ⋮
```

This manual is structured according to the available namelist blocks. The tables will guide you in the following way to change the option of a parameter.
Black: The parameter has a default value. You only have to add this parameter to your namelist if you want to change its option.
Green: If changing a option requires setting other parameters, you have to add the green parameters.
Red: If changing a option requires setting other parameters in a new namelist, you have to add the red namelist with the green options.
Blue: For some special cases you can find more information in this manual which is linked to the according blue words.

1.1 Examples

Documented example `*.par` and input/output files are located in `/Documentation/Example_IO_Files/`.

A complete small example is provided in `/examples/small_example/`.

2 Equations

This block handles the information related to the equations to be solved, as well as all the material properties and similar basic information. In the underlying ADER-DG method, this is equivalent to specifying the shape and size of the Jacobian and reaction matrices to be used. The size of the Jacobian (square) matrices is the same as the size of the vectors of unknowns, and thus the larger they are the more computational cost is involved in their solution. The standard for a SeisSol simulation is the elastic isotropic case, in which the Jacobian matrices have a size of 5 (in 2D) and 9 (in 3D) and no reaction terms. In this case, the flags for advection, anisotropy, anelasticity, poroelasticity and the material file are set equal to 0 and the materials are described by only three material reference values, namely ρ (material density), μ and λ (Lamé parameters). This is fully described in [8] and [4].

In the following, we list all parameters with their possible and default values to be set in the namelist **Equations** (if changing a value results in setting a new parameter in a different namelist, this namelist is written in red). A blue keyword indicates a link to the according section:

Table 1: Parameters and their default values as well as type for the namelist **Equations**. changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green). If changing the default value requires an additional setting of a parameter in a different namelist, the namelist is written in red. The blue words represent links for navigating to a detailed description of the parameter.

parameter	Default Value	Type	Short Description
rho mu lambda	1. 1. 1.	real real real	Material density Lamé Parameter Lamé Parameter
Anisotropy	0	0/1	Different wave velocities in different directions.
Anisotropy = 1 Namelist 'Output': iOutputMask(11)		0/1	Constants for Jacobian. Change this parameter in the namelist Output, which already exists.
Anelasticity	0	0/1	Viscoelastic dissipation or attenuation.
Plasticity	0	0/1	Plastic material behaviour for the whole domain.
Plasticity = 1 PlastCo BulkFriction Tv MaterialType		 real real real character	 plastic cohesion bulk friction relaxation coefficient choose a material type to assign the initial background stress for the whole domain
Adjoint	0	0/1	Generating an adjoint field simultaneously.
Adjoint = 1 AdjFileName		 character	 read the adjoint wavefield in
MaterialType	0		Indicates source and behaviour of material properties.
MaterialType = 1,3-9, 11, 101 MaterialFileName		 character	 Reads material parameters from file

MaterialType = 2 and Anelasticity = 1			
nMechanisms		integer	# attenuation mechanisms
FreqCentral		real	Central Frequency
FreqRatio		real	Frequency ratio
RandomField_Flag	0	integer	Material random fields
RandomField $\neq 0$			
New Namelist 'RFFile': RF_Files		character	According to the number of random field flags the names of the RF_Files have to be set.

ANISOTROPY: This option allows to use materials with different wave velocities in different directions, usually described by the 21 independent coefficients of the general Hooke's tensor. Isotropic materials are just particular cases of the anisotropic framework. The size of the Jacobian matrices remains unchanged, and so anisotropy has little impact in the computational costs. Setting the value of anisotropy equal to 1 requires also setting the material file option to 1. Additional information can be found in [3].

ANELASTICITY: This option allows to include viscoelastic dissipation or attenuation using the Generalized Maxwell Body theory. The attenuation is measured in terms of two quality factors $\mathcal{Q}$, one for the P- and one for the S-waves. This value will then be approximated in a given frequency band by a series of attenuating mechanisms in order to reproduce an almost frequency independent attenuation. The use of this option enlarges the size of the Jacobian matrices by 3 (in 2D) or 6 (in 3D) per attenuating mechanism used and causes the apparition of a reaction matrix. As a consequence the computational costs increase and the user should choose carefully the number of mechanisms required for his particular case. A number between 2 and 5 mechanisms is considered acceptable for attenuating bandwidths of around 100Hz. Setting the value of anelasticity to 1 requires setting the material file option to 1. Additional information can be found in [9].

PLASTICITY: This option enables the use of a Drucker-Prager yield criterion that is used to simulate plastic material behaviour.
Specifying 1 as plasticity option requires three additional variables providing the parameters for plastic cohesion c, the bulk friction v and a relaxation coefficient T_v. T_v is used for the adjustment factor for the stresses when the material behaves plastically. For the relaxation coefficent, we recommend using a guiding value of $T_v \approx \frac{\Delta x/(\#BF)}{V_p}$ where Δx is the smallest distance between two nodes, $\#BF$ is the number of basis functions (depending on the approximation order) and V_s is

the S-wave velocity.
When using the plasticity option, the user has to specify initial background stresses for every element in the domain. For assigning an initial background stress, please write an additional material case in the file
`/trunk/src/Physics/ini_ model.f90`
by adding another case to the variable `EQN%LinType`. This additional case is read in in 'MaterialType' inside the Equations block. Further, the user needs to add this material case in the file readpar.f90.
The Drucker-Prager yield criterion:
In every time step and for every element the values for normal and shear stresses are first calculated elastically. Then the yield criterion and the maximized shear stress over all orientations are calculated for the elastic values (added the initial background stresses):

$$\begin{aligned} \tau_c &= c \cdot \cos\phi - \sigma_m \cdot \sin\phi \\ \tau_{max} &= \sqrt{\sigma_{xy}^2 + (\sigma_{xx} - \sigma_{yy})/2)^2}, \end{aligned}$$

where c is the plastic cohesion, $\phi = \arctan(v)$ the internal coeffcient of friction for the bulk friction v and σ_m the mean stress.
If the yield criterion($\tau_c < \tau_{max}$) is violated, the deviatoric stresses are adjusted by the factor $1 - (1 - \tau_c/\tau_{max})(1 - \exp(-\Delta t/T_v))$, for the timestep Δt and T_v as stated above. The initial background stress is then subtracted such that it is only used to check the yield criterion. For further information see /trunk/src/Solver/-plasticity.f90 or [1].

2.1 Material File

A material file is required in the following cases: either the user wants to treat heterogeneous materials or at least one of the anisotropy, anelasticity or poroelasticity options is activated. When option 1 is selected for the material file, the name of the material file, which has a `.def` extension by convention, has to be given. Additionally, the three values set as material reference values will become now dummy values now.
A general material file will always begin by defining the number of material zones N existing in our simulation. This number must be the same as the zones defined in our mesh. Then, for each of this zones 1,...,N we must first give the zone index and then a series of m material values, depending on the rheology type the user has chosen. Material properties can also be individually applied to each element. In this case, each element can be considered and is treated as a zone without specifying any zones in the mesh generator. In this particular case of element-wise material properties the number of zones at the beginning of the material definition file has to equal the total number of elements.

Elastic Isotropic ($m = 3$): ρ, μ and λ.

Elastic Anisotropic ($m = 31$): ρ, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{22}, c_{23}, c_{24}, c_{25}, c_{26}, c_{33}, c_{34}, c_{35}, c_{36}, c_{44}, c_{45}, c_{46}, c_{55}, c_{56}, c_{66}, n_x, n_y, n_z, s_x, s_y, s_z, t_x, t_y, t_z.

Poroelastic ($m = 42$):ρ_S, c_{11}, c_{12}, c_{13}, c_{14}, c_{15}, c_{16}, c_{22}, c_{23}, c_{24}, c_{25}, c_{26}, c_{33}, c_{34}, c_{35}, c_{36}, c_{44}, c_{45}, c_{46}, c_{55}, c_{56}, c_{66}, ρ_F, K_F, ν, K_S, ϕ, κ_1, κ_2, κ_3, T_1, T_2, T_3, n_x, n_y, n_z, s_x, s_y, s_z, t_x, t_y, t_z.

where the $\vec{n}$, $\vec{s}$ and $\vec{t}$ entries are the directions of the three coordinate axis in which is defined the material described in the material file. The reader is addressed to the bibliography for the description of each of these parameters.

In the anelastic case, we must add three lines after the first, which define the number of attenuating mechanisms, central frequency and bandwidth of the attenuating mechanisms given by the ratio of the maximum to minimum frequency over which the Q-law is approximated by the attenuation mechanisms. Those apply to all material zones. Then, attenuation will be described individually for each zone by adding two additional values to the material properties (that means $m_{new} = m_{old} + 2$) which are the $\mathcal{Q}$ factors for the P- and S-waves at that particular zone. Those values have to be typed after all the other parameter values but before the $\vec{n}$, $\vec{s}$ and $\vec{t}$ entries, if present. A value of $\mathcal{Q}$ higher than 9999 in any of those implies that no attenuation is considered in that particular layer. Note, that in the anelastic material case the material parameters λ and μ are frequency dependent and therefore have to be specified for the central frequency as given in line 2 of

Table 2: Structure of material file

Values				Description
#I				Number of material zones (referred to as N)
#I				Number of attenuating mechanisms
#R				Central frequency of attenuation bandwidth
#R				Bandwidth of attenuation
1	$\#R_1$	$\cdots$	$\#R_m$	Index of zone and m material parameters for zone 1
$\vdots$				
N	$\#R_1$	$\cdots$	$\#R_m$	Index of zone and m material parameters for zone N

Table 3: Structure of random field file

Values				Description
#I	#I			Zone number Number of affected material parameters
#I	#I	#I	$\#I_1 \ldots \#I_m$	Number of samples (NX NY NZ) Material parameter indices
#R	#R	#R	$\#R_1 \ldots \#R_m$	Location (XYZ) Perturbation for 1st ... mth material parameter

the material definition file.
Notice, that acoustic wave propagation can be simulated by setting $\mu = 0$. This way the code understands that the value set for the λ parameter corresponds to the acoustic bulk modulus K. It should also be remarked that values higher than 1 are possible for the material file value, but those correspond to specific cases not covered in the present general manual.

2.2 Random Field File

When random distribution material properties are used we set this flag equal to the number of zones in the model affected by the randomness. This imply that we set also the Material File flag to 1. In the next lines we introduce the names of the files that describe the random properties for each zone affected. The perturbation values (e.g. $\Delta\rho, \Delta\mu, \Delta\lambda$) are added to the background material parameters (e.g. ρ, μ, λ). The structure of the random field file and an example are given in Tab. 3 and Tab. 4 respectively.

Table 4: Example of random field file

1	3				
10	20	30	1	2	3
X_1	Y_1	Z_1	$\Delta\rho_1$	$\Delta\mu_1$	$\Delta\lambda_1$
X_1	Y_1	Z_2	$\Delta\rho_2$	$\Delta\mu_2$	$\Delta\lambda_2$
X_1	Y_1	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_1	Y_1	Z_{30}	$\Delta\rho_{30}$	$\Delta\mu_{30}$	$\Delta\lambda_{30}$
X_1	Y_2	Z_1	$\Delta\rho_{31}$	$\Delta\mu_{31}$	$\Delta\lambda_{31}$
X_1	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_1	Y_{20}	Z_{30}	$\Delta\rho_{600}$	$\Delta\mu_{600}$	$\Delta\lambda_{600}$
X_2	Y_{20}	Z_{30}	$\Delta\rho_{601}$	$\Delta\mu_{601}$	$\Delta\lambda_{601}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
X_{10}	Y_{20}	Z_{30}	$\Delta\rho_{6000}$	$\Delta\mu_{6000}$	$\Delta\lambda_{6000}$

3 Initial Condition

In the following, we list all parameters with their possible and default values to be set in the namelist **IniCondition** If changing a value requires setting a new parameter it is indicated in green after the dashed line:

Table 5: Parameters and their default values as well as type for the namelist **IniCondition**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

parameter	Default Value	Type	Short Description
variable	1	integer	
xc(:)	0.0	real	
amplitude	0.0	real	
hwidth(:)	5.0e3	real	
cICType	Gauss_Puls_Rad	character	Defines the type of initial condition.
cICType = Var_Gauss_Puls, Char_Gauss_Puls, Char_Ricker_Puls, Planarwave_Gauss_Puls, Planarwave_Ricker_Puls, Planarwave, PlanarwaveAnel, PlanarwaveAn, PlanarwaveAniso			
IniConditionFile		character	Reads variables from file.

3.1 IniConditionFile

3.1.1 'Var_Gauss_Puls', 'Char_Gauss_Puls', 'Char_Ricker_Puls'

Table 6: Structure of the initial condition file in the cases of the Var_Gauss_Puls', 'Char_Gauss_Puls' and 'Char_Ricker_Puls'.

Homogeneous background; 2D: vector(5), 3D: vector(9)
setvar
variables
amplitudes
center coordinates (x, y, z)
Halfwidths (x, y, z)
Read normal direction of GP coord. system
Read tangent 1 direction of GP coord. system

3.1.2 'Planarwave_Gauss_Puls','Planarwave_Ricker_Puls'

Table 7: Structure of the initial condition file in the cases of the 'Planarwave_Gauss_Puls' and 'Planarwave_Ricker_Puls'.

Homogeneous background; 2D: vector(5), 3D: vector(9)
setvar
variables
amplitudes
center coordinates (x, y, z)
Halfwidths (x, y, z)
Read normal direction of GP coord. system
Read tangent 1 direction of GP coord. system
File Name for data for eigenvectors and eigenvalues

EigenVecValName

Table 8: Structure of the eigenvector/value file in the cases of the 'Planarwave_Gauss_Puls' and 'Planarwave_Ricker_Puls'.

cdummy
Eigenvalues
cdummy
Read eigenvalues (insert one line for each eigenvalue)
Read eigenvectors (insert one line for each eigenvector)

3.1.3 'Planarwave'

Table 9: Structure of the initial condition file in the cases of the 'Planarwave'.

Homogeneous background; 2D: vector(5), 3D: vector(9)
setvar
variables
amplitudes
iLambda ('0' or default)
case lambda = 0: Read directly the 3D wavenumber vector; case default: Read 3D wavelength vector

3.1.4 'PlanarwaveAnel','PlanarwaveAn', 'PlanarwaveAniso'

Table 10: Structure of the initial condition file in the cases of the 'PlanarwaveAnel', 'PlanarwaveAn' and 'PlanarwaveAniso'.

Homogeneous background; 2D: vector(5), 3D: vector(9)
setvar
variables
amplitudes
File Name for data for eigenvectors and eigenvalues

EigenVecValName

Table 11: Structure of the eigenvector/value file in the cases of the 'PlanarwaveAnel', 'PlanarwaveAn' and 'PlanarwaveAniso'.

cdummy
Eigenvalues
cdummy
read wavenumbers
cdummy
Read eigenvalues (insert one line for each eigenvalue)
Read eigenvectors (insert one line for each eigenvector)

4 Boundaries

In the following, we list all parameters with their possible and default values to be set in the namelist **Boundaries**. If changing a value requires setting a new parameter it is indicated in green after the dashed line (if changing a value results in setting a new parameter in a different namelist, this namelist is written in red). A blue keyword indicates a link to the according section:

Table 12: Parameters and their default values as well as type for the namelist **Boundaries**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green). If changing the default value requires an additional setting of a parameter in a different namelist, the namelist is written in red. The blue words represent links for navigating to a detailed description of the parameter.

Parameter	Default Value	Type	Short Description
BC_fs	0	integer	# free surfaces boundaries.
BC_nc	0	integer	# non-conforming boundaries.
BC_dr	0	integer	# fault inner boundaries.
BC_dr $\neq 0$ New namelist: DynamicRupture			
BC_if	0	integer	# inflow boundaries.
BC_if $\neq 0$ New namelist: InflowBounduin u0_in + New namelist: InflowBound setvar	0	real integer	dimension of varfield vector.
char_option		character	Inflow boundary conditions (different possibilities): Char_Gauss_Puls, Char_Ricker_Puls, Custom_PlaneWave_File.
char_option = Custom_PlaneWave_File PWFileName New namelist: InflowBoundPWFile	0	integer	Wave time histories read from file.
varfield	0	integer	# periodic boundaries.
BC_of	0	integer	# outflow boundaries.
BC_pe	0	integer	# periodic boundaries.

5 Dynamic Rupture

If dynamic rupture is simulated set rupture inner boundaries to 1 and add the new namelist **DynamicRupture**. Dynamic rupture is treated in SeisSol as a boundary condition and, thus, is not listed in the source section. For details about its implementation and the theory we refer to [10] and [11].

Since the namelist **DynamicRupture** requires many additional parameters, the structure of the tabulars differs slightly from the rest of the tabulars in this documentation. For changing the three parameters Backgroundtype, Friction Law and Nucleation you can find three additional tabulars below (14, 15, 16).

Table 13: Parameters and their default values as well as type for the namelist **DynamicRupture**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green). If changing the default value requires an additional setting of a parameter in a different namelist, the namelist is written in red. If you change the value for *BackgroundType*, *FL* or *Nucelation*, add the table 14, 15, 16, respectively.

Parameter	Default Value	Type	Short Description
BackgroundType	0	integer	Type of the stress background. For changing the parameter go to 14.
FL	0	integer	Type of the Friction Law. For changing the parameter go to 15.
Nucleation	0	integer	Characteristics of the Nucleation Patch. For changing the parameter go to 16.
RF_output_on	0	0/1	Rupture front output off/on
magnitude_output_on	0	0/1	output scalar seismic moment off/on
cohesion_0	0	real	Default cohesion value
read_fault_file	0	0/1	Input of fault parameters from Par_file_faults off/on (see section 5.1)
OutputPointType	3	integer	Output is Pickpointwise.
OutputPointType = 3 New namelist: Pickpoint printtimeinterval OutputMask nOutPoints PPFileName	 2 (1, 1, 1, 0)	 integer integer integer character	 Timeinterval # Output points File where Pickpoints are listed; x, y, z next to each other separated by a space;
OutputPointType = 4 New namelist: Elementwise printtimeinterval OutputMask refinement_strategy	 2 (1, 1, 1, 0, 1) 2	 integer integer integer	 Timeinterval

refinement	2	integer	
BinaryOutput	0	integer	0: ASCII; 1: binary float; 2: binary double
OutputPointType = 5			
Add both namelists:			
Pickpoint and Elementwise			

Table 14: Parameters and their default values as well as type for the namelist **DynamicRupture**, if you have set *BackgoundType* unequal to 0. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

Parameter	Default Value	Type	Short Description
BackgroundType	0	integer	Type of the stress background.
BackgroundType = 0-5, 7, 10, 11, 14, 15, 50, 100, 101, 103			
Bulk_xx_0		real	
Bulk_yy_0		real	
Bulk_zz_0		real	
ShearXY_0		real	
ShearYZ_0		real	
ShearXZ_0		real	
RS_sv0		real	
XRef		real	
YRef		real	
ZRef		real	
BackgroundType = 16, 17			
FileName_BackgroundStress		character	Background stress read from file.
GPwise		0/1	0 for elementwise, 1 for grid-pointwise.
XRef		real	reference point
YRef		real	reference point
ZRef		real	reference point

Table 15: Parameters and their default values as well as type for the namelist **DynamicRupture**, if you have set *FL* unequal to 0. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

Parameter	Default Value	Type	Short Description
FL	0	integer	Type of the friction Law
FL = 1			
Rupspeed		real	Rupture Speed.
Mu_D_ini		real	
Mu_S_ini		real	
H_Length		real	
FL = 2			
Mu_D_ini		real	
Mu_S_ini		real	
D_C_ini		real	
inst_healing		0/1	Instantaneous healing switch.
FL = 3, 4 ,7, 101		real	
RS_f0		real	
RS_sr0		real	
RS_a		real	
RS_b		real	
RS_sl0		real	
RS_iniSlipRate1		real	
RS_iniSlipRate2		real	
FL = 6			
Mu_D_ini		real	
Mu_S_ini		real	
D_C_ini		real	
v_star		real	
L		real	
FL = 16, 17			Parameters are already defined in the input file
FL = 103			
RS_f0		real	
RS_sr0		real	
RS_a		real	
RS_b		real	
RS_sl0		real	
Mu_W		real	
RS_srW		real	
RS_iniSlipRate1		real	
RS_iniSlipRate2		real	

Table 16: Parameters and their default values as well as type for the namelist **DynamicRupture**, if you have set *Nucleation* unequal to zero. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

Parameter	Default Value	Type	Short Description
Nucleation	0	integer	Characteristics of the nucleation patch type.
Nucleation = 1			square patch
NucDirX		real	
NucXmin		real	
NucXmax		real	
NucDirY		real	
NucYmin		real	
NucYmax		real	
NucBulk_xx_0		real	
NucBulk_yy_0		real	
NucBulk_zz_0		real	
NucShearXY_0		real	
NucShearYZ_0		real	
NucShearXZ_0		real	
NucRS_sv0		real	
Nucleation = 2,3		real	smooth (2) or discontinuous (3) elliptical patch
NucDirX		real	
NucXmin		real	
NucXmax		real	
NucDirY		real	
NucYmin		real	
NucYmax		real	
r_s		real	
NucBulk_xx_0		real	
NucBulk_yy_0		real	
NucBulk_zz_0		real	
NucShearXY_0		real	
NucShearYZ_0		real	
NucShearXZ_0		real	
NucRS_sv0		real	

5.1 Fault input using Par_file_faults

Apart from defining basic values in the namelist `DynamicRupture` and coding most of the background parameters in `Physics/ini_model_DR.f90`, fault parameters can also be read in from a text file called `Par_file_faults`. This feature is enabled by setting `read_fault_file` in the `DynamicRupture` namelist to 1. Values set in this file have priority over values set in the PARAMETER-file and over those set in the subroutines in `Physics/ini_model_DR.f90`. For sign and vector conventions see `RD_sign_conventions.pdf`. These conventions are compatible with SPECFEM3D.

The file uses namelists defining different geometric shapes to specify the parameters. The stress field and slip-weakening friction input can be written in a SPECFEM3D-compatible style. The `Par_file_faults` file from SPECFEM3D can simply be copied into the working directory and SeisSol will interpret these parts correctly (see section 5.1.2).

5.1.1 Heterogeneous distribution blocks (DIST2D-blocks)

DIST2D-blocks use different geometric shapes for specifying heterogeneous distributions of different parameters. The parameter on all boundary Gaussian points inside of a specified shape is set to the given value. This means that for the specified parameter p at the boundary Gaussian point with coordinates $\vec{x}$

$$p(\vec{x}) = \begin{cases} a, & \vec{x} \in V \\ p_{\text{prev}}(\vec{x}), & \text{otherwise} \end{cases} \tag{1}$$

with a being the value specified in the distribution block, V the volume defined by the block's shape and $p_{\text{prev}}(\vec{x})$ the value the parameter previously had.

The values specified in these blocks are exclusive for each parameter, they are not added up. The last block in which a point lies specifies the value of the specified parameter. The stress fields in local and global coordinates are added up.

For example, to set up a homogeneous stress field with a nucleation patch, a global shape for the homogeneous stress field and afterwards a rectangular shape for the nucleation patch could be specified.

If SPECFEM3D namelists are present, the first blocks are used for the ones specified in these namelists (see section 5.1.2). The values of `param` and `dir` are ignored for these blocks.

Table 17: Namelist **DIST2D**, used for setting the value of a parameter inside a geometric shape to a given value.

Parameter	Default value	Type	Short description
param	-	string	String defining the parameter on which this block acts
param = 'globalstress'			Stress field using global xyz-coordinates
dir	-	string	Stress field direction
param = 'localstress'			Stress field using fault-local coordinates
dir	-	string	Stress field direction
param = 'cohesion'			Fault cohesion (negative, since compression is negative normal stress)
Rate-and-state friction			
param = 'd_c'			Critical slip
param = 'inistatevar'			Initial state variable used for rate-and-state friction
param = 'rs_a'			RS constitutive parameter "a"
param = 'rs_srw'			Velocity weakening scale
param = 'inimu'			Initial friction coefficient
Linear slip-weakening friction			$\mu = \begin{cases} \mu_\mathrm{s} - \frac{\mu_\mathrm{s}-\mu_\mathrm{d}}{D_\mathrm{c}}\delta, & \delta < D_\mathrm{c} \\ \mu_\mathrm{d}, & \delta \geq D_\mathrm{c} \end{cases}$
param = 'd_c'			Critical slip-weakening distance
param = 'mu_s'			Static friction coefficient
param = 'mu_d'			Dynamic friction coefficient
param = 'strength'			Strength parameter $S = \frac{\tau_\mathrm{s}-\tau_0}{\tau_0-\tau_\mathrm{d}}$ (static and dynamic tractions τ_s and τ_d, shear traction τ_0). Use only if no initial stress field is given.
dir	-	string	String defining the stress field direction on which this block acts
param = 'globalstress'			
dir = 'xx'			Bulk stress xx-component
dir = 'yy'			Bulk stress yy-component
dir = 'zz'			Bulk stress zz-component

dir = 'xy'			Shear stress xy-component
dir = 'yz'			Shear stress yz-component
dir = 'xz'			Shear stress xz-component
param = 'localstress'			
dir = 'strike'			Along-strike shear stress
dir = 'dip'			Along-dip shear stress
dir = 'normal'			fault-normal stress (negative is compression)
shapeval	-	string	String defining the shape of this block
shapeval = 'global'			The parameter at all points is assigned to this value
val	0	real	Value of the parameter
shapeval = 'square'			Cube
val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
l	0	real	Side length of the cube
shapeval = 'rectangle'			Cuboid
val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
lx	0	real	length of the cuboid along x
ly	0	real	length of the cuboid along y
lz	0	real	length of the cuboid along z
shapeval = 'rectangle-taper'			Cuboid with a linear variation of the value along the z-direction
val	0	real	Value of the parameter at $z = z_c - l_z/2$
valh	0	real	Value of the parameter at $z = z_c + l_z/2$
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
lx	0	real	length of the cuboid along x
ly	0	real	length of the cuboid along y
lz	0	real	length of the cuboid along z
shapeval = 'circle'			Sphere

val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
r	0	real	Radius of the sphere
shapeval = 'ellipse'			Ellipsoid with semiaxes along x-, y- and z-direction
val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
lx	0	real	Length of semiaxis along x
ly	0	real	Length of semiaxis along y
lz	0	real	Length of semiaxis along z
shapeval = 'x-cylinder'			Cylinder with central axis along x-direction
val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
r	0	real	Radius of the base circle
lz	0	real	Height of the cylinder
shapeval = 'y-cylinder'			Cylinder with central axis along y-direction
val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
r	0	real	Radius of the base circle
lz	0	real	Height of the cylinder
shapeval = 'z-cylinder'			Cylinder with central axis along z-direction
val	0	real	Value of the parameter
xc	0	real	x-coordinate of the center
yc	0	real	y-coordinate of the center
zc	0	real	z-coordinate of the center
r	0	real	Radius of the base circle
lz	0	real	Height of the cylinder

5.1.2 SPECFEM3D namelists

The file can optionally start with a part in the syntax of the SPECFEM3D `Par_file_faults` file. The purpose of this section is primarily offering compatibility with this format. The functionality offered by directly using heterogeneous blocks with a value for `param` (DIST2D-blocks, described in section 5.1.1) is more powerful.

In addition to the DIST2D-blocks, SPECFEM3D files use three different namelists for specifying the stress field and the slip-weakening friction parameters: `stress_tensor`, `init_stress` and `SWF`. The first DIST2D-blocks in the file are interpreted as those specified in `init_stress` (in the order 1. along-strike, 2. along-dip, 3. normal). The blocks after these are interpreted as those specified in `SWF` (in the order 1. static friction coefficient, 2. dynamic friction coefficient, 3. critical slip-weakening distance). All the following blocks are interpreted using the value in `param`.

Table 18: Namelist **init_stress**, used for defining a homogeneous stress field in fault-local coordinates and specifying a number of heterogeneous blocks (DIST2D-blocks) for a heterogeneous stress field in fault-local coordinates

Parameter	Default value	Type	Short description
S1	0	real	Initial homogeneous along-strike shear stress
S2	0	real	Initial homogeneous along-dip shear stress
S3	0	real	Initial homogeneous fault-normal stress (negative is compression)
n1	0	integer	Number of heterogeneous blocks (DIST2D-blocks) for along-strike shear stress
n2	0	integer	Number of heterogeneous blocks (DIST2D-blocks) for along-dip shear stress
n3	0	integer	Number of heterogeneous blocks (DIST2D-blocks) for fault-normal stress

Table 19: Namelist **stress_tensor**, used for defining a homogeneous stress field in global XYZ-coordinates

Parameter	Default value	Type	Short description
Sigma	0 0 0 0 0 0	real	Initial homogeneous stress field, components: XX, YY, ZZ, XY, YZ, XZ

Table 20: Namelist **SWF**, used for defining homogeneous slip-weakening friction parameters and specifying a number of DIST2D-blocks for heterogeneous parameters

Parameter	Default value	Type	Short description
mus	0	real	Homogeneous static friction coefficient
mud	0	real	Homogeneous dynamic friction coefficient
dc	0	real	Homogeneous critical slip-weakening distance
nmus	0	integer	Number of heterogeneous blocks (DIST2D-blocks) for the static friction coefficient
nmud	0	integer	Number of heterogeneous blocks (DIST2D-blocks) for the dynamic friction coefficient
ndc	0	integer	Number of heterogeneous blocks (DIST2D-blocks) for the critical slip-weakening distance

5.1.3 Example

An example of a `Par_file_faults` file with different shapes and parameters is given below. It contains a section with SPECFEM3D-compatible definitions for a fault-local stress field.

```
! ----------------- SPECFEM3D-compatible part -----------------
&INIT_STRESS S1=0.0, S2=0.0, S3=0.0, n1=0, n2=1, n3=1 /

! ------ along-dip ------
&DIST2D shapeval='x-cylinder', val=-80e6, xc=0, yc=0, zc=-15e3,
    lz=3e3, r=2.0e3 /

! ----- fault-normal -----
! Stress: 1020 · |z|
&DIST2D shapeval='rectangle-taper', val = -20.4e6, valh = 0, xc
    =0, yc=-3.75e3, zc=-10e3, lx=32.0e3, ly=9.0e3, lz=20.0e3 /

! ------------------- SeisSol-exclusive part -------------------
&DIST2D param='localstress', dir='dip', shapeval='square', val =
     -100e6, xc=0, yc=0, zc=-5e3, l=2.0e3 /

&DIST2D param='globalstress', dir='yz', shapeval='circle', val
    =60e6, xc=0, yc=0, zc=-10e3, r=3.0e3 /

&DIST2D param='cohesion', shapeval='ellipse', val=40e6, xc=10e3,
     yc=0, zc=-10e3, lx=4e3, ly=1e3, lz=2e3 /
```

6 Source Terms

In the following, we list all parameters with their possible and default values to be set in the namelist **SourceType**. If changing a value requires setting a new parameter it is indicated in green after the dashed line (if changing a value results in setting a new parameter in a different namelist, this namelist is written in red):

Table 21: Parameters and their default values as well as type for the namelist **SourceType**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green). If changing the default value requires an additional setting of a parameter in a different namelist, the namelist is written in red. The blue words represent links for navigating to a detailed description of the parameter.

Parameter	Default Value	Type	Short Description
Type	0	integer	Source Type
Type = 1			
New namelist:			
Source110			
U0		real	
l1		real	
Type = 16, 18			
nRicker		integer	# Ricker Sources in case 16. # Gaussian Sources in case 18.
New namelist:			
Source1618			
SpacePositionx		real	x-line of Spaceposition-matrix.
SpacePositiony		real	y-line of Spaceposition-matrix.
SpacePositionz		real	z-line of Spaceposition-matrix.
Delay		real	
a1		real	
f		real	
EqnNr		real	
Type = 20, 30, 50			
FileName		character	Read parameters from file.

6.1 Source%Type 16 = RICKER

Like source type 18 but other wavelet!. Dirac sources in space and Ricker wavelet in time.

$$g''(\tau) = a_1 \left(1/2 + a_2\tau^2\right) \cdot \exp(a_2\tau^2) \tag{2}$$

6.2 Source%Type 18 = GAUSS

Dirac sources in space and Gaussian wavelet in time. Single point source described by

$$g(\tau) = a_1 \cdot \exp(-\tau/f^2) \tag{3}$$

6.3 SOURCE%Type = 20

A single point force is an important source type for the computation of Green's functions. Computing the solution for the three different point forces in x-, y- and z-direction allows us to synthesize seismograms that would result from an arbitrary moment tensor afterwards. Each single force can have an individual source time function for the slip. As an example, Tab. 22 shows the buildup of such a source file.
However, the synthesized seismograms will result in an equivalent moment tensor source which uses the slip rate as source time function. If the slip rate function is normalized to 1, the single force seismograms also have to be scaled by multiplying with the corresponding scalar seismic moment of the moment tensor source and the shear modulus, i.e.

$$s(t)_{MomentTensor} = s(t)_{SingleForce} \cdot M_0 \cdot \mu\,. \tag{4}$$

Furthermore, in combination with the single force source, it is useful to use a specific output option given in Section 10 and denoted as Moment Tensor Contribution.

6.4 SOURCE%Type = 50

A very important source type is given by the SOURCE%Type=50. It offers extended finite sources with a double couple moment tensor with individual slip rate history for each subfault. All information is comprised in a separate file, which we usually call `FSRM-filename.dat` as it is a Finite Source Rupture Model. As an example, Tab. 23 shows the buildup of such a source file. Lines 2 - 4 include the components of the seismic moment tensor. In this case we have chosen 2 Pa for the $xz-$ component.

The number of subfaults is given in line 6. In the example of Table 23 line 8 and 9 (the number of lines has to coincide with the number of subfaults!) give

Table 22: Structure of Single Force source file

Number of Sources
1
Single Force on Variable Nr.
7
x y z
10. 0. 20.
Source Time Functions
0.001 3
Samples
0.
1.3
0.

Table 23: Structure of Double Couple source file

Seismic Moment Tensor							
0.	0.	2.					
0.	0.	0.					
2.	0.	0.					
Number of Subfaults							
2							
x	y	z	strike	dip	rake	area·μ	onset time
30.	0.	90.	0.	1.57	0.	1.	0.123
12.	8.	10.	0.	1.57	0.	1.	0.234
Source Time Functions							
0.001	3						
Samples							
0.							
1.3							
0.							
0.							
1.6							
0.							

information about the location and the properties of the rupture. Angles of strike, dip, and rake have to be specified in rad. Afterwards the shape of the impulse time function itself is specified by the sampling interval (e.g. 0.001 sec) and the number of samples (e.g. 3). Then, the 3 values of these samples are given line by line for subfault 1. Then the values of subfault 2 are listed and so on, such that there are $n =$ number-of-subfaults $\times$ number-of-samples lines.

7 Sponge Layer

The sponge consists of one (or more) cohesive layer(s) around the computational domain. In contrast to common absorbing boundary conditions it uses not just the outermost boundary of the grid but has a thickness of several elements. As already indicated, the sponge layer is a technical construct which absorbs the outgoing waves in order to minimize unwanted reflections due to the artificial boundaries.
The current parameter file allows 3 different values for the type of sponge: Option 0 means that no sponge is used. Option 1 activates one (or more) sponge layer(s), in which the outgoing waves are attenuated uniformly. The dumping factor depends on the distance to the outermost boundary, where the waves should finally be absorbed completely.
Several studies of the last few years showed that these sponge layers do not work satisfying. Reflections into the computational domain are still observed. Therefore, another type of layer (option 2) was introduced. The perfectly matched layer (PML) differentiates between the spatial components of the outgoing waves and therefore improves the nullification of reflections. As in our case the PML is still under construction, option 1 is kept running.

In the following, we list all parameters with their possible and default values to be set in the namelist **SpongeLayer**. If changing a value requires setting a new parameter it is indicated in green after the dashed line (if changing a value results in setting a new parameter in a different namelist, this namelist is written in red):

Table 24: Parameters and their default values as well as type for the namelist **SpongeLayer**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green). If changing the default value requires an additional setting of a parameter in a different namelist, the namelist is written in red.

Parameter	Default Value	Type	Short Description
enabled	0	0, 1, 3	Sponge on/off or usage of PML.
enabled = 1			
DGSpongeTol		real	
intDummy		integer	
nDGSponge		integer	# Sponge boundaries.
New namelist:			
Sponges			
SpongeDelta		real	
SpongePower		real	
SigmaMax		real	
enabled = 3			
PMLDelta		real	
Refl_Coeff		real	
PMLPrefactor		real	
PMLFrequency		real	

8 Mesh

In the following, we list all parameters with their possible and default values to be set in the namelist **MeshNml**.

Table 25: Parameters and their default values as well as type for the namelist **MeshNml**.

Parameter	Default Value	Type	Short Description
MeshFile	LOH1	character	Name of mesh file.
mesh generator	Gambit3D	character	Meshgenerator.
displacement	(0.0, 0.0, 0.0)	real	
ScalingMatrixX	(1.0, 0.0., 0.0)	real	x-component of scaling matrix.
ScalingMatrixY	(0.0, 1.0., 0.0)	real	y-component of scaling matrix.
ScalingMatrixZ	(0.0, 0.0., 1.0)	real	z-component of scaling matrix.
periodic	0	0/1	Periodic boundary conditions off/on.
periodic_direction	(0, 0, 0)	0/1	Vector with 3 switches, indicating if periodic boundary exists in x-, y- and/or z-direction.

9 Discretization

This block collects the information on the polynomial approximation used for the numerical simulation as well as the particularities of the solver chosen.
The SeisSol variants are, at the present moment, either 1 or 3 (option 2 is disabled). Option 1 means a global time stepping (GTS) computation and option 3 a local time stepping (LTS) computation. The difference between both is that GTS updates all the cells from time step to time step at once. Therefore the minimum stable time increment has to be used for all cells. LTS computations, on the other hand, iterate the updates for each cell independently, minimizing the total number of operations required. In general, for very uneven meshes, LTS should be faster. Further information can be found in [5]. The time integration variant includes two possible types. First is the standard ADER based upon a Cauchy-Kovalewski procedure (value 0), as used in most practical problems and described in [8] and [4]. The other type is the Space-Time Discontinuous ADER time integration (value 1) which is just as accurate but more robust for solving stiff problems (see [2]).
For LTS simulation, we must specify what we call a computing *cycle*. Using a value 1 means that one cycle is fulfilled each time we loop over all elements, independently on whether we update their values or not. Using a value 2 means that one cycle is fulfilled once each and every element has been at least updated once.
Next, we can adapt the order of the scheme $\mathcal{O}$. This value is linked to the polyomial degree of the basis functions p by $\mathcal{O} = p - 1$. An order of accuracy must be assigned, ranging from 1 to 10 (in 2D) or from 1 to 7 (in 3D).
Finally, the Courant number C can be adjusted for stability reasons. This number is the fraction of the CFL stability limit, which for our DG schemes is

$$\Delta t \leq C \frac{1}{2p+1} \frac{\Delta x}{\alpha}\,, \tag{5}$$

being α the maximum wave velocity allowed, usually the P-wave velocity, Δx a measure of the size of the element (3D tetrahedral elements: in-sphere diameter) and p is the polynomial degree of the approximation. Notice, that equation (5) must be fulfilled by all elements, hence the advantage of the above mentioned LTS schemes over GTS schemes in highly uneven meshes. The value of C is bound by the upper value $C = 1$, although for ADER-DG the maximum is $C = 0.7$. Often a safer value of $C = 0.5$ is used.
The last important parameter is the fixed timestep, a value Δt at which we want to impose the scheme to iterate. In general, we do not want to impose such number, but let the code choose the optimal following equation (5). Then it is recommended to use a very large value here, much larger that the time scale of the problem to be solved.

In the following, we list all parameters with their possible and default values to be set in the namelist **Discretization**. If changing a value requires setting a new

parameter it is indicated in green after the dashed line:

Table 26: Parameters and their default values as well as type for the namelist **Discretization**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

Parameter	Default Value	Type	Short Description
CFL	0.5	real	
FixTimeStep	5000	integer	
CKMethod	0	0/1	0: standard CK procedure; 1: Using space-time DG approach.
FluxMethod	0	0/1	0: Gudunov flux; 1: Rusanov flux.
DGMethod	0	integer	Possiblity to choose time stepping.
DGMethod = 1			Global time stepping
Order		integer	
nPolyMap	0	integer	
Material	0	integer	Material basis functions degree.
DGMethod = 3			Local time stepping
Order		integer	
nPolyMap	0	integer	
Material	0	integer	Material basis functions degree.
IterationCriterion		1/2	1: Iteration is defined as one cycle; 2: Iteration is defined by update of all elements.

10 Output

This block includes all the information on the outputting of data from the SeisSol simulations. The data output is about the two types: *snapshot* and *receiver* output. The snapshot output is volumetric information on the variables at some particular instant of the simulation. The receiver output is a time continuous recording of the variables at a particular point in space. In general, the user has to specify the output variables as well as when and where these variables are going to be recorded.

A generic filename provided by the user will be created of different character strings to produce unique files for the various outputs. The structure and content of these files will be treated in a Section of its own.

When the rotational output option 1 is selected, the code will output rotation rates around the axes x, y and z. Then, the user is required to specify which of this 3 rotation rates he wants to output with 0-1 switches.

When the moment tensor contribution output option 2 is selected, the code will output the six further quantities, which are the moment tensor contributions of a single force in x-, y- or z-direction, depending on the Single Force Variable Nr. specified in Source&Type $= 20$. For further information see [7].

The number of variables (stresses and velocities) that can be outputted depends on the equations being solved and the dimension. A 0-1 switch has to be specified for each variable, meaning 1 that the user wants to output that particular variable and 0 that he does not want. The following variables' switches have to be specified:
2D, 5 switches: σ_{xx}, σ_{yy}, σ_{xy}, u, v.
3D, 9 switches: σ_{xx}, σ_{yy}, σ_{zz}, σ_{xy}, σ_{yz}, σ_{xz}, u, v, w.

For the poroelastic case, one additional parameter has to be provided specifying the output switches for the fluid variables (pressure and fluid velocities) as follows:
2D, 3 switches: p, u_f, v_f.
3D, 4 switches: p, u_f, v_f, w_f.

In the following, we list all parameters with their possible and default values to be set in the namelist **Output**. If changing a value requires setting a new parameter it is indicated in green after the dashed line:

Table 27: Parameters and their default values as well as type for the namelist **Output**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

Parameter	Default Value	Type	Short Description
OutputFile	data	character	Root name of output file.
iOutputMaskMaterial	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	integer	
iOutputMaskMaterial	(0, 0, 0)	integer	
Rotation	0	integer	
Format = 1,5,10	1	integer	1 Tecplot, 5 HDF5/XDMF, 10 3D output off
printIntervalCriterion		integer	Criterion about time printed time step.
printIntervalCriterion = 1,3			
Interval		real	
printIntervalCriterion = 2,4			
TimeInterval		real	
pickdt	0.1	real	
pickDtType	1	integer	
pickDtType	1	integer	
nRecordPoints		integer	
RFileName	RecordPoints	character	
PGMFlag	0	0/1	Peak ground motion output at given receivers.
PGMFlag = 1			Number of receivers for peak ground motion output, followed by their 3D coordinates.
PGMFile		character	
FaultOutputFlag	0	0/1	Switch fault output off/on.
checkPointInterval	0	integer	Interval for writing checkpoints (0 = disable checkpoints)
checkPointFile		character	Checkpoint filename

10.1 Snapshot Output

For the snapshot output, the material parameters to be outputted have to be defined by switches in a similar way as the variables just shown in the last paragraphs. The material values depend on the rheology type chosen. They are described in the Section 2.1 in the proper order. All the material parameters can be outputted with the sole exception of the $\mathcal{Q}$ values and the $\vec{n}$, $\vec{s}$ and $\vec{t}$ entries. The switches have to be typed in order and not more than 11 switches per line. If the number of material parameters is larger than 11, a second line will have to be added. In case more than 22 parameters are required, a third line is needed.
SeisSol currently only accepts 2 output format for snapshots and requires TECPLOT or paraview (use postprocessing conversion script tecp2vtk!) to be visualized:

1: TECPLOT format
5: HDF5/XDMF parallel binary output for paraview

There are 3 possible criteria: Output every given number of timesteps (option 1), every given multiples of a specified time (option 2) or when any of both criteria just mentioned are fulfilled (option 3).

10.2 Receiver Output

For the receiver output, three parameters are required: the time sampling rate, the number of receiver stations and their locations. The code will automatically interpolate the variables' values at the desired times.

11 Abort Criteria

In the following, we list all parameters with their possible and default values to be set in the namelist **AbortCriteria**. If changing a value requires setting a new parameter it is indicated in green after the dashed line:

Table 28: Parameters and their default values as well as type for the namelist **AbortCriteria**.

Parameter	Default Value	Type	Short Description
EndTime	15.0	real	
MaxIteration	10000000	integer	
WallTime_h	1e20	real	
Delay_h	0.0	real	

12 Analysis of Data

This option is only important for convergence tests, to analyze the numerical error with respect to an analytical solution and to output directly the errors in the L_1, L_2, or L_∞-norm on screen or in the log files.
In all other cases (e.g. simulations for applications), which are not convergence studies this option has to be 0.

In the following, we list all parameters with their possible and default values to be set in the namelist **Analysis**. If changing a value requires setting a new parameter it is indicated in green after the dashed line (if changing a value results in setting a new parameter in a different namelist, this namelist is written in red):

Table 29: Parameters and their default values as well as type for the namelist **Analysis**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green). If changing the default value requires an additional setting of a parameter in a different namelist, the namelist is written in red.

Parameter	Default Value	Type	Short Description
typ	0	integer	Choice of analysis type.
variables	(0, 0, 0, 0, 0, 0, 0, 0, 0, 0)	0/1	Analysis variables.
typ = 14			
setvar		integer	
EigenVecValName		character	
New namelist:			
AnalysisFields			
varfield		real	
ampfield		real	

13 Debugging Modus

At the moment, to be set to 0 as other options are not possible. This option is a default for debugging purposes.

In the following, we list all parameters with their possible and default values to be set in the namelist **Debugging**. If changing a value requires setting a new parameter it is indicated in green after the dashed line:

Table 30: Parameters and their default values as well as type for the namelist **Debugging**. Changing the parameter from the default value often results in setting additional parameters in the corresponding namelist (indicated in green).

Parameter	Default Value	Type	Short Description
debug_flag	0	0/1	Debug mode off/on.
debug_flag = 1			
level	0	integer	

14 DISCLAIMER

Note that the here documented features are not all compatible with each other. Moreover, no guarantee is given that some features might be outdated or their functionality changed as the SeisSol development is always in progress. We highly recommend to test thoroughly each functionality for your individual problem.
Furthermore, our general DISCLAIMER holds:
THIS SOFTWARE OR DOCUMENTATION IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS ”AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

References

[1] D.J. Andrews, 2005. *Rupture dynamics with energy loss outside the slip zone*, Journal of Geophysical Research.

[2] de la Puente, J., Dumbser,M., Käser, M. & Igel, H.: Discontinuous Galerkin Methods for Wave Propagation in Poroelastic Media, *Geophysics*. 73(5), T77-T97, 2008.

[3] de la Puente, J., Käser, M., Dumbser, M. & Igel, H., 2007. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes - IV. Anisotropy. *Geophys. J. Int.* **169 (3)**, 1210-1228.

[4] Dumbser, M. & Käser, M., 2006. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes - II. The Three-Dimensional Isotropic Case. *Geophys. J. Int.* **167 (1)**, 319-336.

[5] Dumbser, M., Käser, M. & Toro, E.F., 2007. An arbitrary high-order Discontinuous Galerkin method for elastic waves on unstructured meshes V. Local time stepping and p-adaptivity. *Geophys. J. Int.*

[6] Dumbser, M., Käser, M. & de la Puente, J., 2007. Arbitrary High Order Finite Volume Schemes for Seismic Wave Propagation on Unstructured Meshes in 2D and 3D, *Geophys. J. Int.*

[7] Graves, W.R. & Wald, D.J., 2001. Resolution analysis of finite fault source inversion using one- and three-dimensional Green's functions 1. Strong motions, *J. Geophys.Res.*, **106 (B5)**, 8745-8766.

[8] Käser, M. & Dumbser, M., 2006. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes - I. The Two-Dimensional Isotropic Case with External Source Terms. *Geophys. J. Int.*, **166 (2)**, 855-877.

[9] Käser, M., Dumbser, M., de la Puente, J. & Igel, H., 2007. An Arbitrary High Order Discontinuous Galerkin Method for Elastic Waves on Unstructured Meshes - III. Viscoelastic Attenuation. *Geophys. J. Int.*, **168 (1)**, 224-242.

[10] Pelties, C., de la Puente, P., Ampuero, J.-P., Brietzke, G. B., and Käser, M.: Three-dimensional dynamic rupture simulation with a high-order discontinuous Galerkin method on unstructured tetrahedral meshes, J. Geophys. Res., 117, B02309, http://dx.doi.org/10.1029/2011JB008857, 2012.

[11] Pelties, C., Gabriel, A.-A., and Ampuero, J.-P.: Verification of an ADER-DG method for complex dynamic rupture problems, Geosci. Model Dev., 7, 847-866, 2014.