QDYN

Quasi-DYNamic earthquake simulator

User's Manual

V 1.3.0

code.google.com/p/qdyn/

Developers

QuakeID Group, Caltech Seismo Lab:

Yingdi Luo (http://www.seismolab.caltech.edu/luo_y.html)

Jean-Paul Ampuero (http://www.seismolab.caltech.edu/ampuero_jp.html)

Bryan Riel (http://www.seismolab.caltech.edu/riel_b.html)

Last modified on 10/29/2014

Click here to access the most recent version

Summary

QDYN is a boundary element software to simulate earthquake cycles (seismic and aseismic slip on tectonic faults) under the quasi-dynamic approximation (quasi-static elasticity combined with radiation damping) on faults governed by rate-and-state friction and embedded in elastic media.

QDYN includes various forms of rate-and-state friction and state evolution laws, and handles non-planar fault geometry in 3D and 2D media, as well as spring-block simulations. Loading is controlled by remote displacement, steady creep or oscillatory load. In 3D it handles free surface effects in a half-space, including normal stress coupling. The medium surrounding the fault is uniform, linear, isotropic and elastic.

QDYN implements adaptive time stepping, shared-memory parallelization, and can deal with multi-scale earthquake cycle simulations with fine details in both time and space. It is equipped with a user-friendly Matlab interface and graphical output utilities.

Features

- Rate-and-state friction, with velocity cut-offs, aging and slip laws
- Arbitrarily heterogeneous frictional properties
- Slow and fast, aseismic and seismic slip transients (adaptive timestep)
- Non-planar faults (currently limited to variable dip, rectangular elements)
- 3D, 2D and 1D (spring-block)
- Steady and oscillatory loads
- Matlab wrapper and graphic output display utilities
- Parallelized for shared memory systems (OpenMP)
- Normal stress coupling

Table of contents

1. Installation

1.1 Download QDYN

QDYN is managed under the Subversion version control system. To get the most current version of QDYN execute the following command on Linux (first time only):

```
svn checkout http://qdyn.googlecode.com/svn/trunk/ qdyn-read-only
```

This creates a directory 'qdyn-read-only' which contains the whole QDYN package. You can create a directory with a different name.

You can also manually download the QDYN source files and pre-complied executables for Windows or Mac OS directly from this link. However, note that these may not be most up-to-date versions, and that pre-compiled executables may run slower than if you compile the code by yourself.

1.2 Install QDYN

1. Move to the qdyn-read-only/src directory
2. Modify the section "User Settings" of file Makefile, following the instructions and examples therein:
    a. In section 1, set the executable file name and its path: set the variable EXEC = [*your executable file name, with its full path*]. By default, the executable file is named ~/bin/qdyn. Make sure the path points to an existing directory. For instance, if the ~/bin/ directory does not exist, create it.
    b. In section 2, adjust your Fortran compiler settings: set the variables F90 = [*your compiler name*] and OPT = [*your compiler optimisation flags*]. Intel Fortran (ifort) is the default compiler, but settings for several commonly used compilers are provided.
3. Update the path to file *src/kernel*tab* in subroutine *init_kernel_2D* of src/fault_stress.f90
4. Run 'make'
5. If in step 2 you changed the path, modify accordingly the line status = system('~/bin/qdyn') of file qdyn.m

1.3 Update QDYN

After the first-time checkout, you can update the package by executing the following command in your qdyn-read-only directory:

```
svn update
```

and then following the steps in section 1.2 again.

2. Running a simulation

2.1 The Matlab wrapper

The core of QDYN is a Fortran code. While the format of its input and output files is well defined, we find it more convenient to set up the input parameters, perform simulations and read the output data within the Matlab environment through the Matlab wrapper qdyn.m. Its general usage syntax is:

```
[pars,ot,ox] = qdyn(mode,[parsin],['Property',Value,...])
```

The default input values can be examined by executing:

```
pars = qdyn('set')
```

The input parameters are:

mode	One of the following execution modes: 'set' outputs the default parameter structure (*pars*) or overrides it with fields present in the structure *parsin* or with *Property/Value* pairs 'write' sets parameters and writes the qdyn input file 'run' sets parameters, writes the input file and runs a simulation 'read' reads parameters and outputs from a previous simulation
parsin	parameter structure to override the default parameters (see 2.2)
'Property'	name of a field to be set in the parameter structure (see 2.2)
Value	value to override the default value and the value in *parsin*

The outputs are (see section 2.3 for more details):

pars structure containing the parameters
ot structure containing time series outputs, global or at selected points
ox structure containing snapshot outputs, i.e. quantities over the whole fault, output at selected times

2.2 Simulation parameters structure (pars)

The parameters in the structure *pars*, that can be set through 'parsin' or 'Prop/Value' pairs are:

Parameters defining the geometry of the problem and loading:

MESHDIM dimension of the problem:
0 = spring-block model
1 = 1D fault in a 2D elastic medium
2 = 2D fault in a 3D elastic medium
4 = same as 2 but fault stresses computed via 2D-FFT (works only if the grid spacings and dip angle are uniform)

MU shear modulus (Pa)
LAM elastic modulus lambda for 3D simulations (Pa)
VS shear wave velocity (m/s). If VS=0 radiation damping is turned off.

L fault length if MESHDIM=1
stiffness is *MU/L* if MESHDIM=0

FINITE boundary conditions for case MESHDIM=1
0 = fault is infinitely long but slip is spatially periodic with period L, loaded by steady displacement at distance W from the fault
1 = fault is infinitely long but only a segment of length L has rate-and-state friction, the rest has steady slip. If running the code with this option gives the error message “finite kernel is too small”, you should create a larger “kernel file” with the matlab function *TabKernelFiniteFlt.m*, update the file name in subroutine *init_kernel_2D* of *src/fault_stress.f90*, and re-compile the code.

W distance between displacement loading and fault, only if MESHDIM=1 and FINITE=0

DIP_W dipping angle (degree). If depth-dependent, values must be given from deeper to shallower depth.

Z_CORNER fault bottom depth (km, negative down)

SIGMA_CPL normal stress coupling
0 = disable
1 = enable

APER amplitude of additional time-dependent oscillatory shear stress loading (Pa)
TPER period of oscillatory loading (s)

Rate-and-state friction parameters:

A amplitude of direct effect
B amplitude of evolution effect
DC characteristic slip distance (m)
MU_SS reference steady-state friction coefficient
V_SS reference steady-state slip velocity (m/s)
TH_SS reference steady-state state (the default is TH_SS=DC/V_SS)
RNS_LAW type of rate-and-state friction law:
0 = original
1 = with cut-off velocities
V1 cut-off velocity of direct effect (m/s)
V2 cut-off velocity of evolution effect (m/s), controls the transition from weakening to strengthening when a<b. V2 should be ≤ V1
THETA_LAW type of evolution law for the state variable:
0 = ageing law in the "no-healing" approximation
1 = ageing law
2 = slip law

Initial conditions:
SIGMA initial effective normal stress (Pa).
Remains constant unless *SIGMA_CPL =1.*
V_0 initial slip velocity
TH_0 initial state

Discretization and accuracy parameters:

N number of fault nodes (elements) if MESHDIM=1
NX number of fault nodes along-strike, in 3D
NW number of fault nodes along-dip, in 3D
DW along-dip length (km) of each node along-dip, from deep to shallow

TMAX total simulation time (s)
NSTOP stopping criterion

0 = at t=TMAX
1 = at end of slip localization phase
2 = at first slip rate peak

DTTRY first trial timestep (s)
DTMAX maximum timestep (0=unrestricted)

ACC solver accuracy

Output control parameters:

OX_SEQ type of snapshot outputs
0 = all snapshots in a single output file (fort.19)
1 = one output file per snapshot (fort.1001, ...)
NXOUT spatial interval for snapshot outputs (in number of nodes)
NTOUT temporal interval (in number of time steps) for snapshot outputs

OX_DYN output specific snapshots of dynamic events defined by thresholds on peak slip velocity DYN_TH_ON and DYN_TH_OFF (see below)
0 = disable
1 = enable outputs for event #i:
event start: fort.19998+3i
event end: fort.19999+3i
rupture time: fort.20000+3i
NXOUT_DYN spatial interval (in number of nodes) for dynamic snapshot outputs
DYN_TH_ON peak slip rate threshold to define the beginning of a dynamic event
DYN_TH_OFF peak slip rate threshold to define the end of a dynamic event

IC index of node for time series outputs

Parameters for integration with dynamic code:

DYN_FLAG integration with dynamic code
0 = disable
1 = enable: stop QDYN at the DYN_SKIP+1-th event with seismic moment > DYN_M
DYN_M target seismic moment of a dynamic event
DYN_SKIP number of dynamic events to skip (warm up cycles)

2.3 Output structures (ot, ox)

The outputs are:

pars structure containing the parameters described above and:
X,Y,Z fault coordinates

ot structure of time series outputs, with the following fields:

t output times
locl localization length (distance between stressing rate maxima)
cl crack length (distance between slip rate maxima)
p seismic potency
pdot seismic potency rate

Outputs at the fault location with maximum slip rate:
xm location
v slip rate
th state variable theta
om slip_rate*theta/dc
tau shear stress
d slip

Outputs at selected fault node with index pars.ic
vc slip rate
thc state variable
omc (slip rate)*theta/dc
tauc shear stress
dc slip

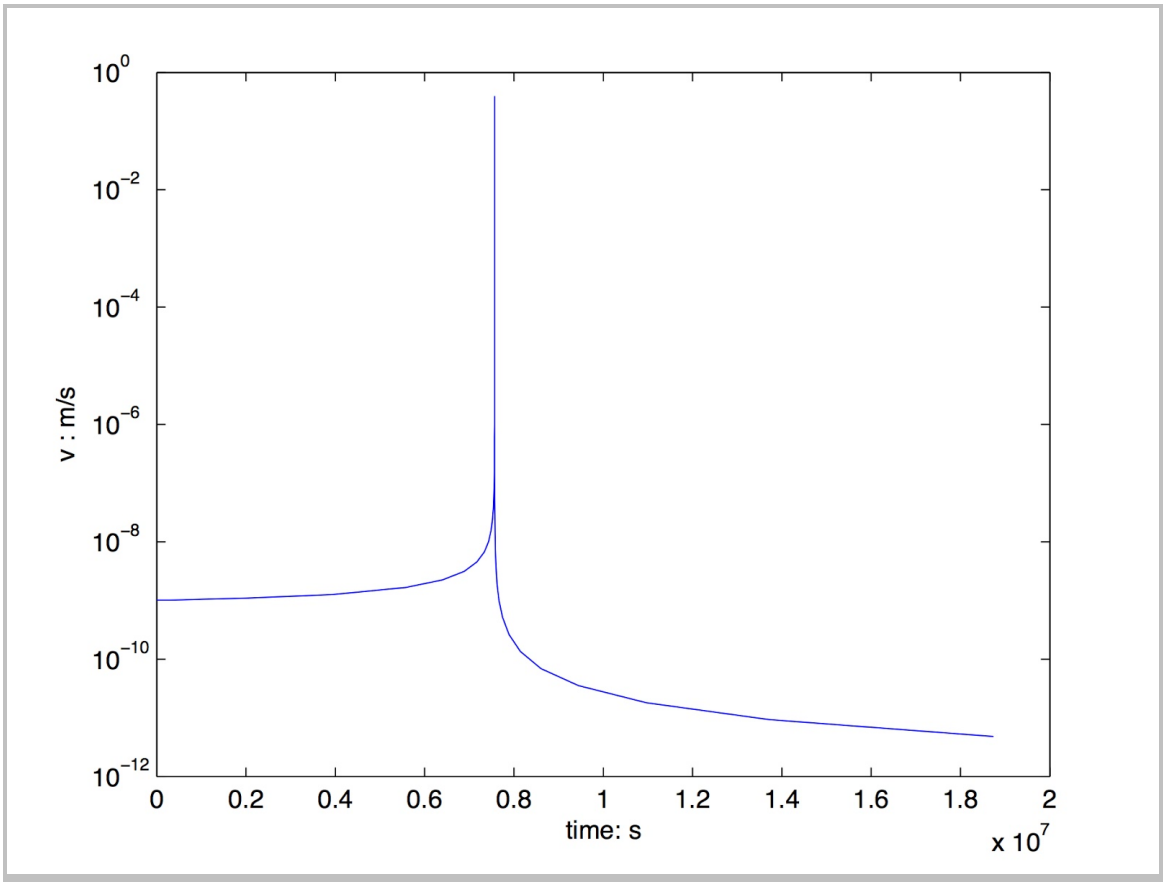
ox structure of snapshot outputs, with the following fields:
x fault coordinates (for 2D)
t output times
v slip rate
th state variable theta
vd slip acceleration
dtau shear stress relative to initial stress
dtaud shear stress rate
d slip
sigma effective normal stress

2.4 A simple 2D example

A 2D run with initial velocity slightly above the default steady state.
In Matlab:

```
p = qdyn('set');
[p,ot,ox] = qdyn('run','V_0',1.01*p.V_SS);
semilogy(ot.t,ot.v); xlabel('time (s)');ylabel('v (m/s)');
```

Estimated simulation time <10s on a single thread machine.

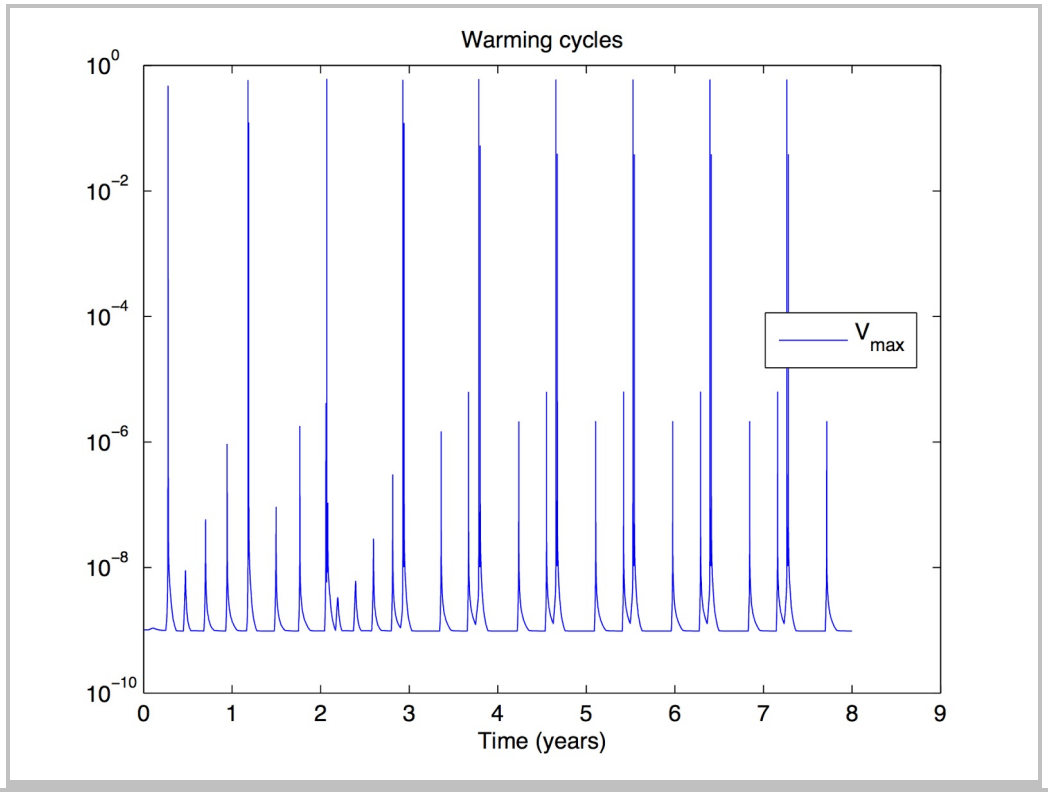


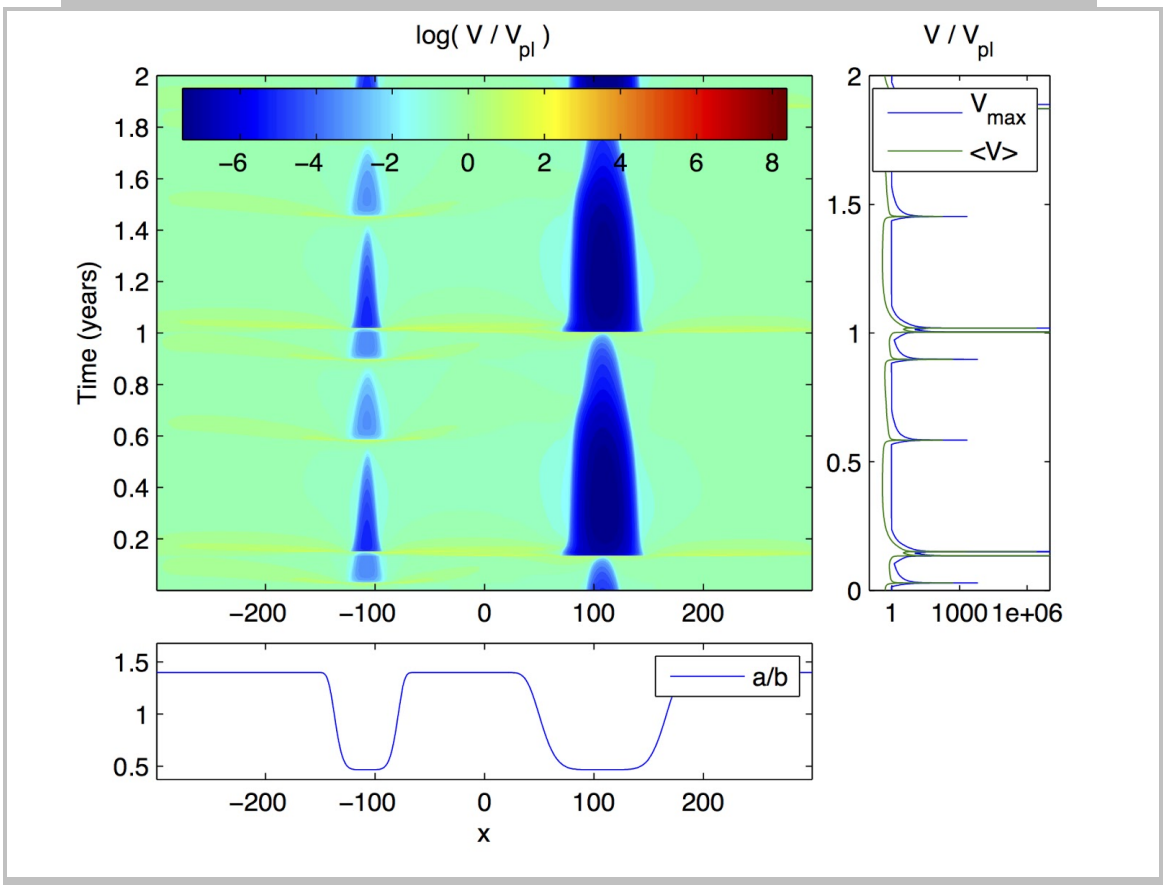
2.3 Two asperities interacting

A large asperity (size/Lc = 4) with long recurrence interval interacts with a small asperity (size/Lc = 2) with short recurrence interval. When the large asperity breaks, its post-seismic slip propagates bi-laterally and triggers rupture of the small asperity. During the interseismic period of the large asperity, the smaller asperity breaks twice with a decreasing recurrence interval.

'working directory'/ Examples/double_asperities

The estimated simulation time is about 6 mins on a single thread machine.

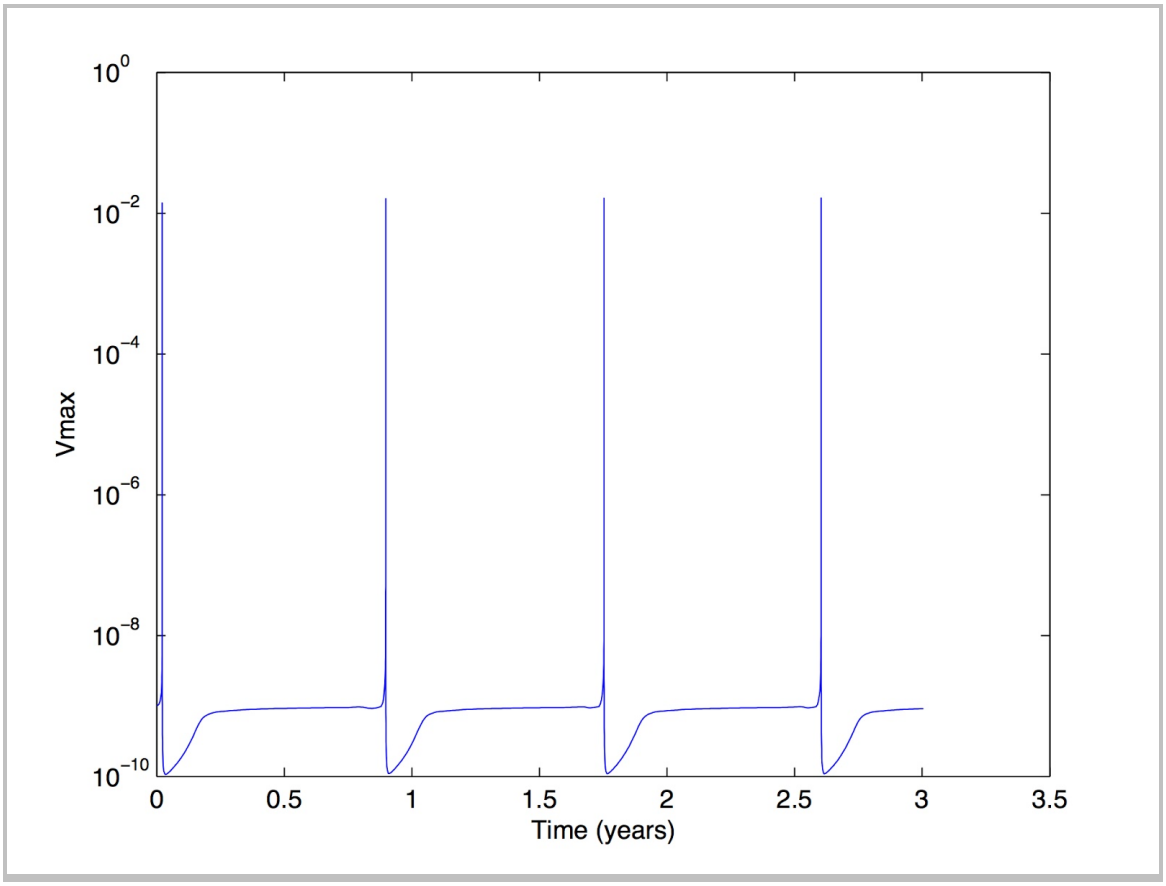

Warming cycles
10^{0}
10^{-2}
10^{-4}
10^{-6}
10^{-8}
10^{-10}
0 1 2 3 4 5 6 7 8 9
Time (years)
V_{max}

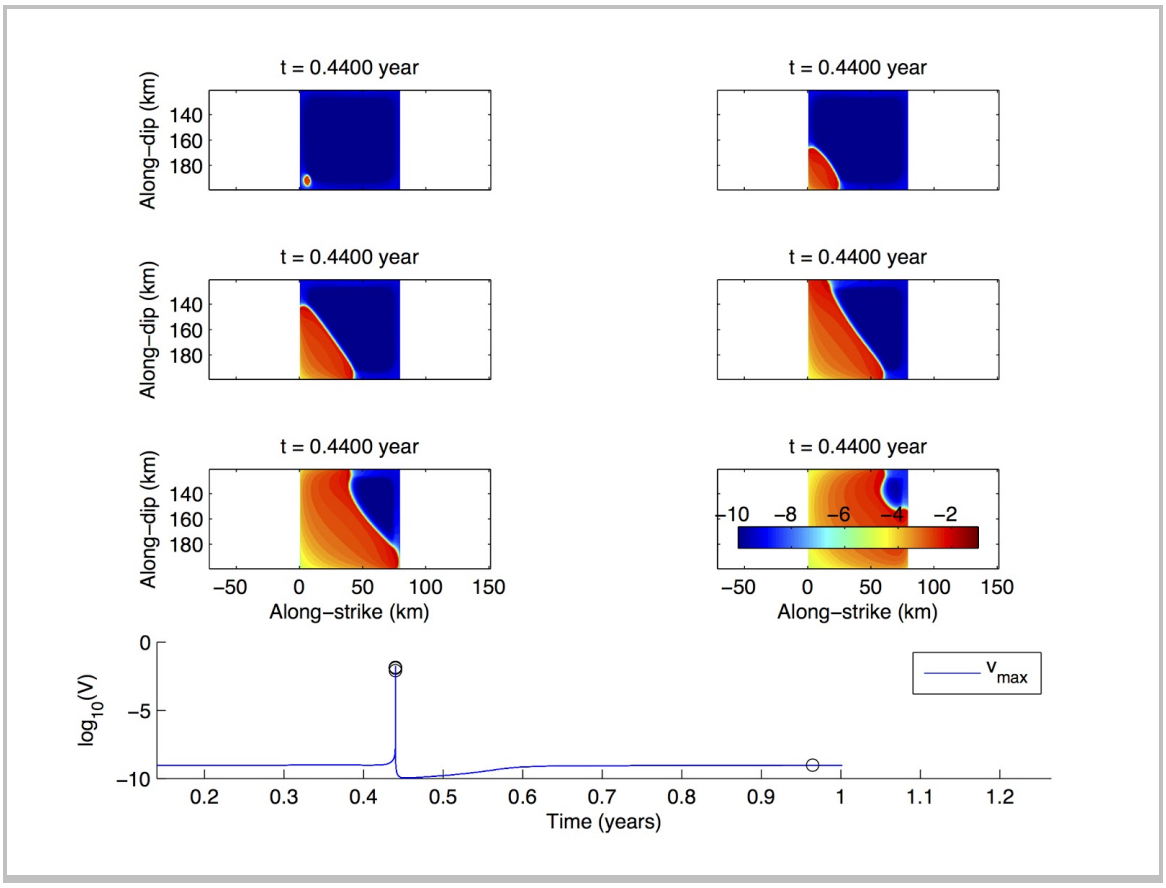

log(V / V_{pl})
V / V_{pl}
Time (years)
−6 −4 −2 0 2 4 6 8
−200 −100 0 100 200
1 1000 1e+06
V_{max}
<V>
a/b
x

2.4 Complicated simulations

An upper-layer Matlab wrapper for the base-layer Matlab wrapper qdyn.m is recommended for complicated simulations. We have included some examples for reference:

a) A simple 3D simulation:
'working directory'/src/test3dfft.m
The estimated simulation time is about 10 mins on a single thread machine with one-cycle simulation.
Sample figures shown here is a complete multi-cycle simulation (4 cycles warming-up and one-cycle output)





b) A simplified model for the Tohoku earthquake (2D along-dip and 3D simulations):

'working directory'/ Examples/Tohoku

For more examples and real-world implications please refer to the wiki page on the website here.

3.Optimizing Performance

3.1 Running simulations outside the Matlab environment

Under certain circumstances, you may want to perform simulations outside the Matlab environment (e.g. while computing on a HPC). In this case, use Matlab wrapper separately first to generate an input file (qdyn.in), then run the qdyn executable directly.

3.2 Managing parallel computing

3.2.1 OpenMP

QDYN is efficiently parallelized with OpenMP (shared memory parallelization). You should set the following environment variable before performing parallel simulations:

```
setenv OMP_NUM_THREADS 8
```

This command allows QDYN to run on 8 threads, which will roughly speed up calculations by a factor of 8. The number of threads should be set according to demand. In general, set this value to the maximum number of threads available on your shared memory system.

3.2.2 MPI

Features will be available in the next major update.

3.3 Managing outputs of complicated simulations

QDYN by default outputs results as a single ASCII file (fort.19), that may become overwhelming while performing large simulations with heavy outputs, (In practical, most multi-cycle 3D simulations). In these cases, it will be helpful to switch output-mode to separate mode,
set `OX_SEQ = 1` in `qdyn.m` , outputs will be generated as separate, sequential ox file outputs (fort.1001, ...)
set `OX_DYN = 1` in `qdyn.m` , qdyn will automatically detect seismic events (`DYN_th_on and DYN_th_on`), and generate 3 snapshots of the i-th event (event start fort.19998+3i, end fort.19999+3i, rupture time fort.20000+3i)

4.Visualizing simulation results:

The QDYN software package includes several Matlab scripts for visualizing simulation results at ‘working directory’/src/
These scripts are all self-documented:

- plot_default.m : plots slip rate of a 2D problem (along-strike);
- plot2d_slip.m : plots slip of a 2D problem (along-dip);
- plot3d_m.m: plots a sequence of snapshots of slip rate for 3D simulation;
- plot3d_faultview3.m: plotting several snapshots of slip rate for 3D simulation in one figure;

Suggested References:

Y. Luo, J. P. Ampuero (2011), *Numerical Simulation of Tremor Migration Triggered by Slow Slip and Rapid Tremor Reversals*, AGU Fall Meeting 2011 Abstract S33C-02
Y. Luo, J. P. Ampuero (2012), *Simulation of Complex Tremor Migration Patterns*, AGU Fall Meeting 2012 Abstract S44B-02