scaling_lab_events

August 24, 2018

In [1]:
```
# Stefan Nielsen 2018
## the inline option is necessary for Latex export of figures:
%matplotlib inline
import os
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from pylab import plot, xlabel, ylabel
import pandas as pd
from xlrd import open_workbook
## This sets PDF format for export to LaTeX,
## while allowing inline SVG in the notebook:
from IPython.display import set_matplotlib_formats
set_matplotlib_formats('svg', 'pdf')
```

In [2]:
```
## Alternative options:
#%matplotlib notebook
#%config InlineBackend.figure_format = 'pdf'
#%config InlineBackend.figure_format = 'png'
#%config InlineBackend.figure_format = 'svg'
#from matplotlib import animation, rc, interactive
#import matplotlib.ticker as ticker
#matplotlib.interactive(True)
#from pylab import *
#from scipy import arange
#from IPython import display
#plt.rcParams.update({'figure.figsize': (10,7)})
# LaTeX support, with pslatex package :
#plt.rc('text', usetex=True);plt.rc('font', family='serif')
#matplotlib.rcParams['text.latex.preamble'] = [r'\usepackage{amsmath}',r'\usepackage{pslatex}']
```

1 Parameters of rock samples:

$$\mu' = 24.3 \text{ GPa}$$
$$\lambda = 39.1 \text{ GPa}$$
$$\rho = 2700 \text{ kg m}^{-3}$$
$$V_p = 5699 \text{ m/s}$$
$$V_s = 3000 \text{ m/s}$$

1.1 import xlsx file, show contents, use contents

In [3]:
```
mu=24.3e9
df=pd.read_excel("event_params.xlsx")
df
```

Out[3]:

	Event	t0/2e-7	t1/2e-7	tw	rise time (s)	tc	Sn	mu0	\
0	157_28c	1927	3000	2243	0.000215	0.000063	75000000	0.5	
1	157_68e	2003	3038	2277	0.000207	0.000055	75000000	0.5	
2	159_184	1984	2725	2539	0.000148	0.000111	80000000	0.5	
3	159_237	1984	2772	2298	0.000158	0.000063	100000000	0.5	
4	159_240	2000	2777	2589	0.000155	0.000118	100000000	0.5	
5	160_27	2000	3981	2329	0.000396	0.000066	58000000	0.5	
6	160_79	1981	4191	2352	0.000442	0.000074	73000000	0.5	
7	160_124	1955	4230	2390	0.000455	0.000087	75000000	0.5	
8	160_130	1867	4300	2398	0.000487	0.000106	76000000	0.5	

	muR	Δmu	Δtau	Vr	Vmax	U	Dc	Dc ida
0	0.4354	0.0646	4845000.0	1100	0.1856	0.000013	0.000005	0.000002
1	0.4284	0.0716	5370000.0	1000	0.1417	0.000009	0.000004	0.000002
2	0.3557	0.1443	11544000.0	2500	0.1792	0.000012	0.000011	0.000023
3	0.3055	0.1945	19450000.0	2190	0.1467	0.000013	0.000006	0.000019
4	0.2902	0.2098	20980000.0	2560	0.1632	0.000014	0.000013	0.000045
5	0.2253	0.2747	15932600.0	2166	0.3548	0.000083	0.000015	0.000016
6	0.1703	0.3297	24068100.0	1460	0.3455	0.000085	0.000016	0.000018
7	0.1018	0.3982	29865000.0	1797	0.5300	0.000129	0.000027	0.000033
8	0.0968	0.4032	30643200.0	1800	0.5199	0.000127	0.000017	0.000041

In [4]:
```
Vmean=df["U"]/df['rise time (s)']
nor=df['Sn']/mu
```

2 Make graphics using xlsx contents:

In [5]:
```
## This sets the dpi resolution for screen and png files:
#plt.rcParams['figure.dpi'] = 120;plt.figure(dpi=120);
# for some reason, it does not work if declared in the initial cell of notebook.
# note that PDF format still takes precedence during export to LaTeX||
```

In [6]:
```
fig1, ax1 = plt.subplots()
xa="$V_\mathrm{max}\ \mathrm{  (m/s)}$";ya='$\Delta \mu$'
ax1.plot(df["Vmax"],(df["mu0"]-df["muR"]),'ro')
xlabel(xa);ylabel(ya)
ax1.set_xlim(left=0);ax1.set_ylim(bottom=0)
ax1.set_xlim(right=2);ax1.set_ylim(top=.55)
x=np.linspace(0.01,2,100)
y=0.5*(1-.12/x)
ax1.plot(x,y);
y=.5+0*x
ax1.plot(x,y);
```

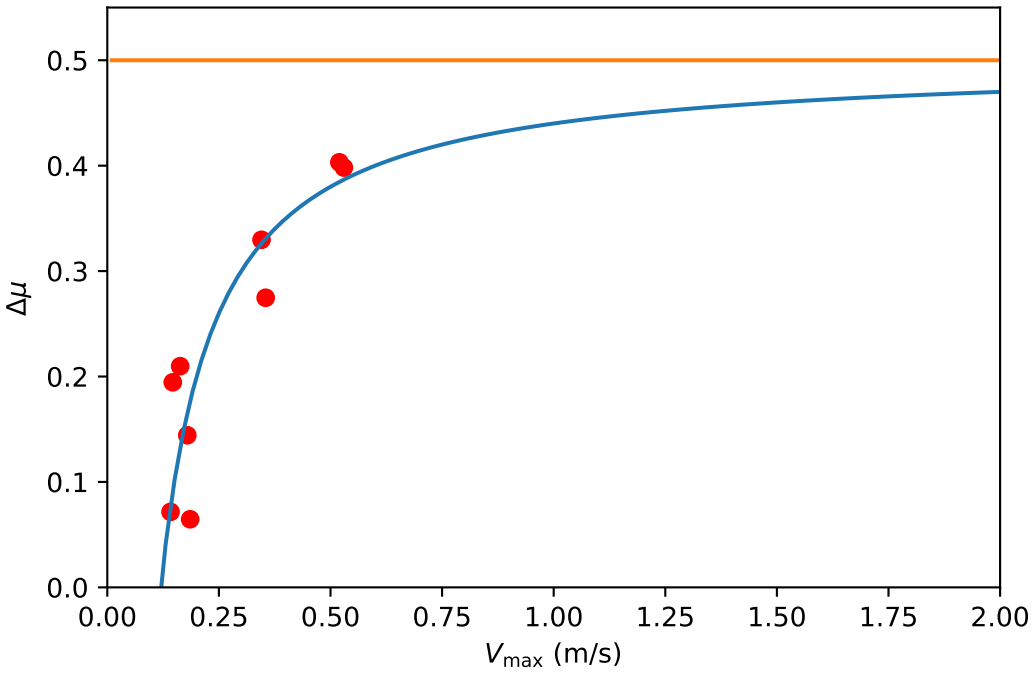

Fig. 1. Stress drop $\Delta\mu = \mu_0 - \mu_{dyn}$ as a function of maximum slip rate $V_{\max}$ in different experimental microearthquakes (red dots). The theoretical fit (blue curve) uses $\Delta\mu = \mu_0(1 - V_w/V)$, which results from $\mu_{dyn} = \mu_0\, V_0/V$, a high-velocity ($V \gg V_w$) approximation of the flash weakening law. Here $V_w = 0.12\,\mathrm{m/s}$ and $\mu_0 = 0.5$ (orange line, or total stress drop, reached asymtotically for $V \to \infty$).

```
In [7]: fig4, ax1 = plt.subplots()
        xa='rise time (s)';ya='U'
        ax1.plot(df[[xa]],df[[ya]],'ro');
        xlabel(xa);ylabel(ya)
        ax1.set_xlim(left=0);ax1.set_ylim(bottom=0)
        #ax1.set_xlim(left=0,right=1e-3);ax1.set_ylim(bottom=0,top=.3e-3)
        x=np.linspace(0,2e-3,100)
        y=5e2*x**2
        vvmax=.12+(x/8e-4)**2
        y=.36*x*(1-0.12/vvmax)
        ax1.plot(x,y)
        y=.36*(x-.0e-4)
        ax1.plot(x,y,linestyle='dotted');
```

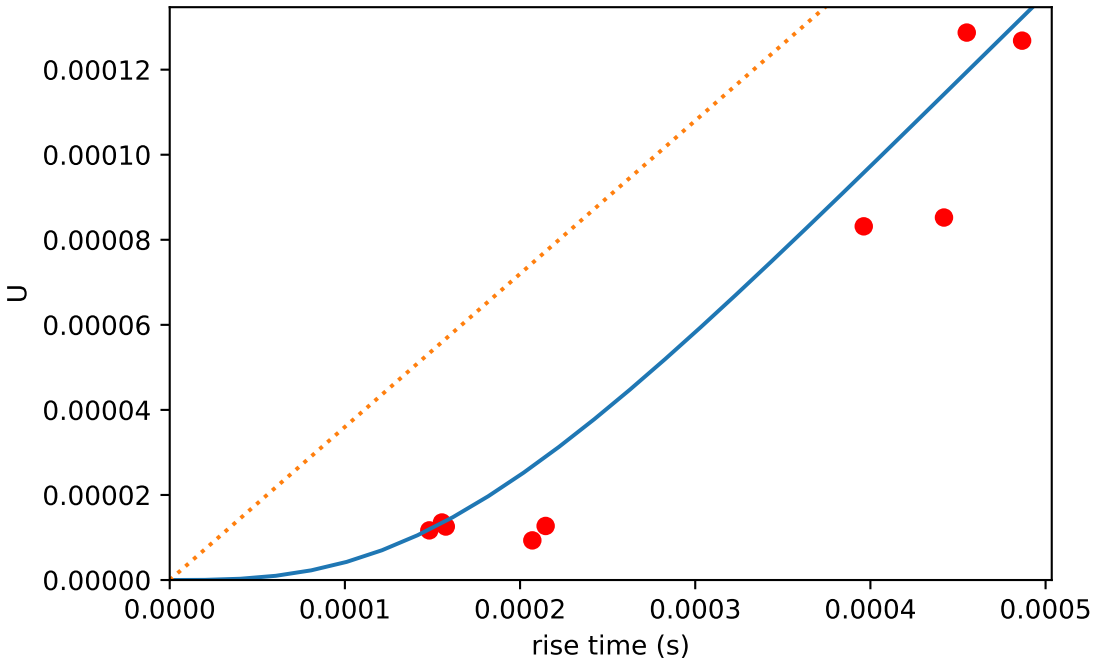

Fig. 4. Total slip for each rupture event as a function of rise time (red dots). The theoretical fit (blue curve) is shown assuming the classic scaling relation

$$U = C\,\frac{\sigma \Delta\mu}{\mu'}\,\Gamma$$

where Γ is the length of the rupture, μ' is the shear stiffness, σ is the normal stress and C is a geometrical constant of the order of 1. Furthermore, we may use

$$\Gamma \approx T\,V_r$$

where T is rise time and V_r is rupture velocity. According to approximate relation of stress drop to maximum slip velocity as discussed in Figure 3, we have:

$$\Delta\mu = \mu_0(1 - V_w/V_{max})$$

And according to the fit of Figure 5, we may replace $V_{max} = 0.12 + (\frac{T}{8\ 10^{-4}})^2$. As a result we obtain the relation:

$$\begin{aligned} U &= \frac{C\,\sigma\,\mu_0\,V_r}{\mu'}\left(1 - \frac{V_w}{0.12 + (\frac{T}{8\ 10^{-4}})^2}\right)\,T \\ &= 0.36\,\left(1 - \frac{V_w}{0.12 + (\frac{T}{8\ 10^{-4}})^2}\right)\,T \end{aligned}$$

where we have used the indicative values, compatibly with the experimental conditions, of $\sigma \approx$ 70MPa, $V_r \approx 1000(\mathrm{m/s})$, $\mu' = 50$ GPa, $\mu_0 = 0.5$, and set the constant $C = 0.6$ to obtain the fit of the experimental points. The asymptotic value $U = 0.36\ T$ at large T is shown as a dotted line. The linear asymptote corresponds to the maximum possible friction drop (possibly close to total drop or $\Delta\mu \approx \mu_0 = 0.5$) which is achieved at large V_{max} (and large T), and whereby self-similar scaling is retrieved.

3 Export to LaTeX (without code cells)

```
In [74]: import os
         os.system('mkdir build');
         os.system('jupyter nbconvert --to=latex --template=latex_nocode.tplx scaling_lab_events.ipynb ')
         #the latex_nocode.tplx will eliminate the code cells - the file latex_article.tplx is also needed
```

```
Out[74]: 0
```

4 compile pdflatex and visualise the result:

```
In [108]: os.system('pdflatex -output-directory=build scaling_lab_events')
          os.system('open build/scaling_lab_events.pdf')
```

```
Out[108]: 0
```

5 Export to LaTeX (including code cells)

```
In [107]: import os
          os.system('mkdir build')
          os.system('jupyter nbconvert --to=latex --template=article_plus.tplx scaling_lab_events.ipynb ')
          #the latex_nocode.tplx will eleiminate the code cells - the file latex_article.tplx is also needed
```

```
Out[107]: 0
```