Modeling the Shape and Evolution of Normal-Fault Facets: Supplemental Information

This document contains supplemental information for the manuscript “Modeling the Shape and Evolution of Normal-Fault Facets” by G.E. Tucker, D.E.J. Hobley, S.W. McCoy, and W.T. Struble. The supplemental information includes analysis of the Grain Facet model’s sensitivity to grain-motion rules and surface roughness, a derivation of the method to calculate vertical erosion rate from a model snapshot, and the procedures used to generate several of the figures in the manuscript. This material is provided mainly in the form of PDF representations of a collection of Jupyter Notebooks, which combine text, mathematics, figures, and Python code.

Contents

S1. Sensitivity to Motion Rules and Rates

The Lattice Grain model of grain motion (Tucker et al., 2016) uses a set of lattice transition rules to represent, in a heuristic way, the mechanics of a granular medium in a gravitational field. One of the assumptions underlying its application to hillslope evolution is that the details of grain motion have little influence on long-term behavior, provided that there exists a clear scale separation between the time scale of grain movement (which is expected to be on the order of seconds) and the time scale of morphologic evolution (10^4 years or more) (Tucker et al., 2018).

This document presents two brief tests of that assumption. The first test changes the gravitational parameter. The second compares model behavior with and without a component of elastic rebound in grain collisions.

Varying gravitational parameter

In the Lattice Grain framework, the rates of motion and collision use a rate parameter, γ (with dimensions of 1/time) that derives from gravitational acceleration. As described by Tucker et al. (2016), the probability distribution for time intervals between (for example) episodes of grain motion derives from treating motion as a Poisson process. Recall that movement of regolith is represented by a cell-pair transition in which the moving-regolith and air cells swap states, indicating that the regolith has moved by one lattice unit. The movement transition is taken to have a fixed probability per unit time of occurring, such that the probability distribution of the time interval between successive motion events is

$$p(t) = \gamma e^{-\gamma t}$$

(note that this probability distribution is conditional on no other transitions occurring that would change the state of either cell in the pair). The same rate parameter γ is used for all of the motion, gravitational, and collision transitions. The time scale $T = 1/\gamma$ is take to be the time required for a grain to fall a distance of one cell unit, δ, assuming that it starts from rest and that there is no air resistance. In this case,

$$T = 1/\gamma = \sqrt{\frac{2\delta}{g}},$$

where g is gravitational acceleration.

In the sensitivity experiment below, we run the model with two different values of g: one for earth (9.8 m/s^2) and one for mars (3.7 m/s^2). The other parameters are configured to represent $w' = 5$ and $d' = 0.5$.

Control run

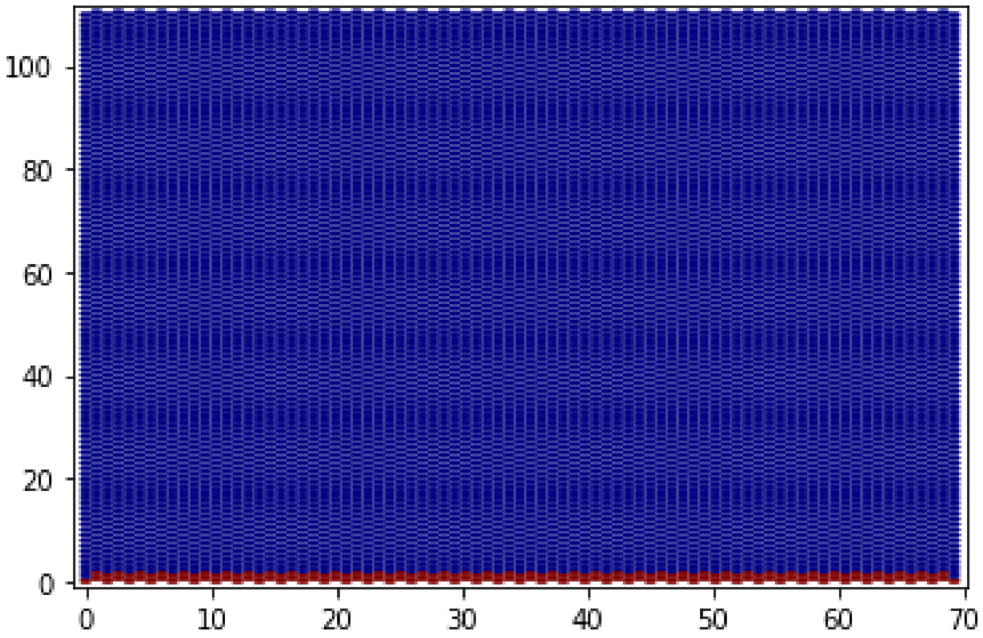
The code in the next three cells imports the necessary packages, sets the input parameters, and runs the model for 30,000 years.

[1]:

```
import numpy as np
import datetime
from grainhill import GrainFacetSimulator
from grainhill import SlopeMeasurer
import landlab
from landlab.io.native_landlab import save_grid
import os
```

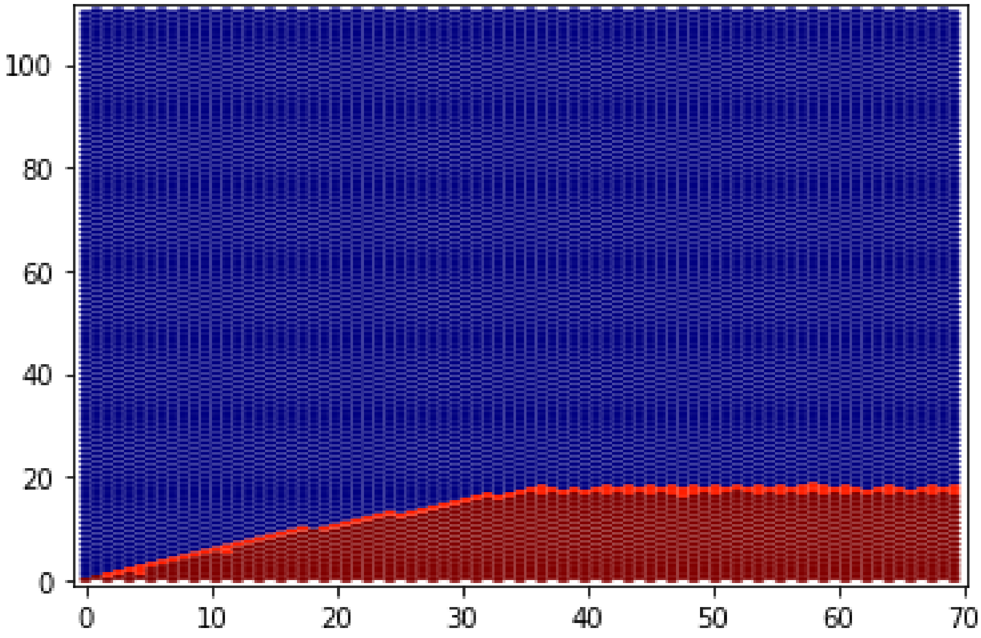
[2]:
```
params = {
    'grid_size' : (111, 81),
    'report_interval' : 5.0,
    'run_duration' : 30000.0,
    'output_interval' : 1.0e99,
    'disturbance_rate' : 0.001,
    'weathering_rate' : 0.01,
    'dissolution_rate': 0.0,
    'uplift_interval' : 866.0,
    'plot_interval' : 10000.0,
    'friction_coef' : 1.0,
    'fault_x' : -0.01,
    'cell_width' : 0.5,
    'grav_accel' : 9.8,
    }
```

[3]:
```
gfs = GrainFacetSimulator(**params)
gfs.run()
```

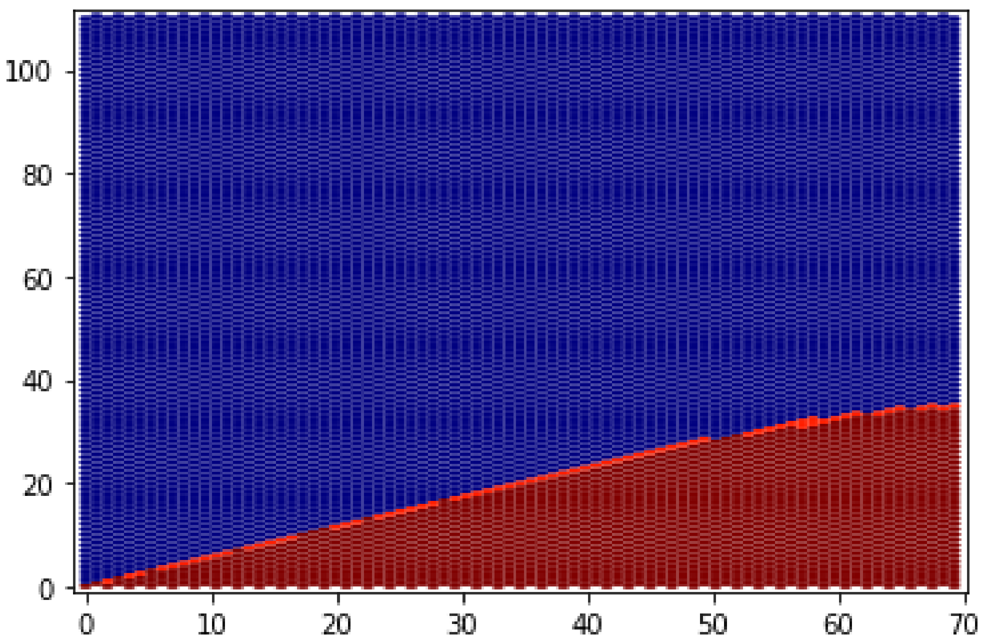


```
Current sim time0.0(0.0%)
Running to...866.0
Running to...1732.0
Running to...2598.0
Running to...3464.0
Running to...4330.0
Running to...5196.0
```

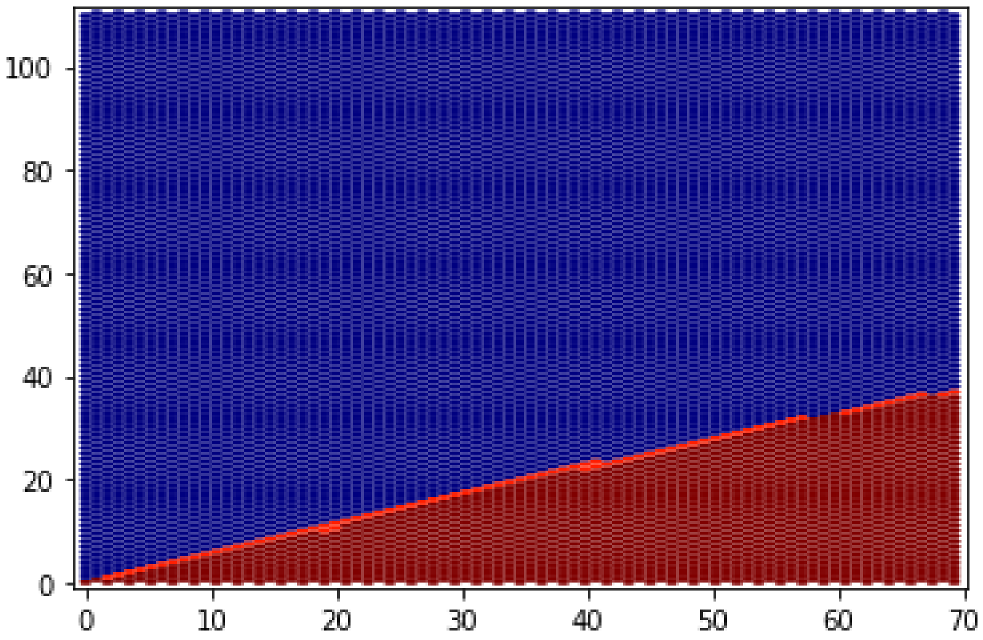
```
Running to...6062.0
Running to...6928.0
Running to...7794.0
Running to...8660.0
Running to...9526.0
Running to...10000.0
```



```
Running to...10392.0
Current sim time10392.0(34.64%)
Running to...11258.0
Running to...12124.0
Running to...12990.0
Running to...13856.0
Running to...14722.0
Running to...15588.0
Running to...16454.0
Current sim time16454.0(54.846666666666664%)
Running to...17320.0
Running to...18186.0
Running to...19052.0
Running to...19918.0
Current sim time19918.0(66.39333333333333%)
Running to...20000.0
```



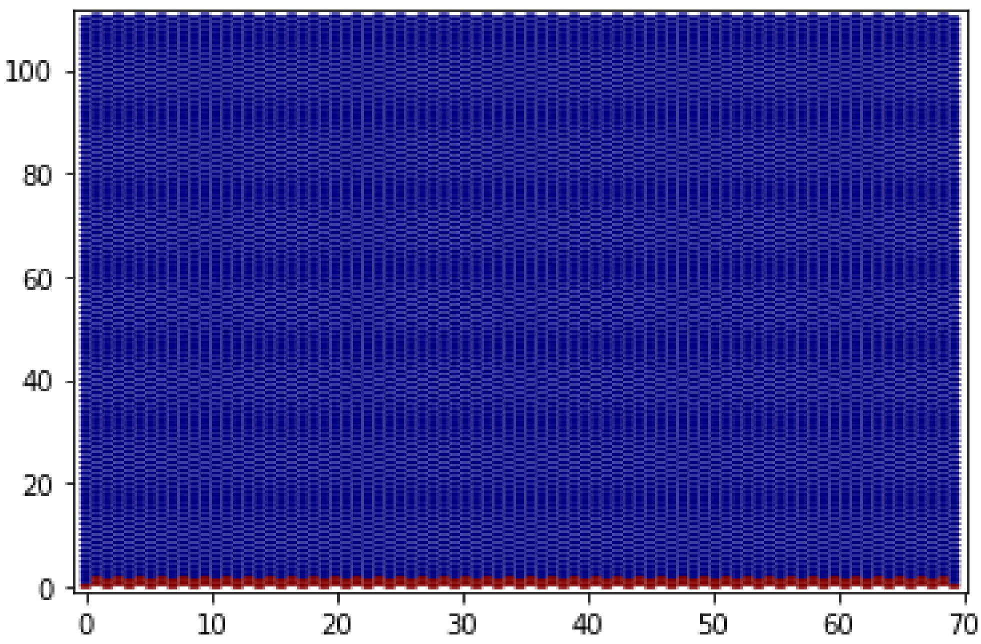
```
Running to...20784.0
Running to...21650.0
Running to...22516.0
Current sim time22516.0(75.05333333333333%)
Running to...23382.0
Running to...24248.0
Running to...25114.0
Current sim time25114.0(83.71333333333334%)
Running to...25980.0
Running to...26846.0
Running to...27712.0
Current sim time27712.0(92.37333333333333%)
Running to...28578.0
Running to...29444.0
Running to...30000.0
```



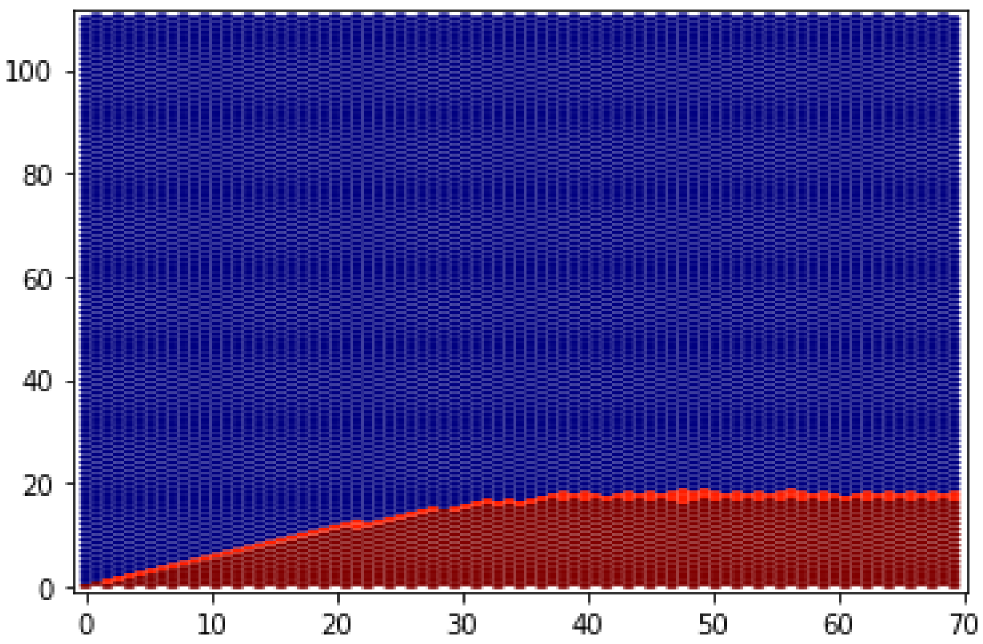
Run with reduced gravitational parameter

Below the gravitational parameter is changed, a new model object is instantiated, and the model is run. All other parameters are the same as those in the control run.

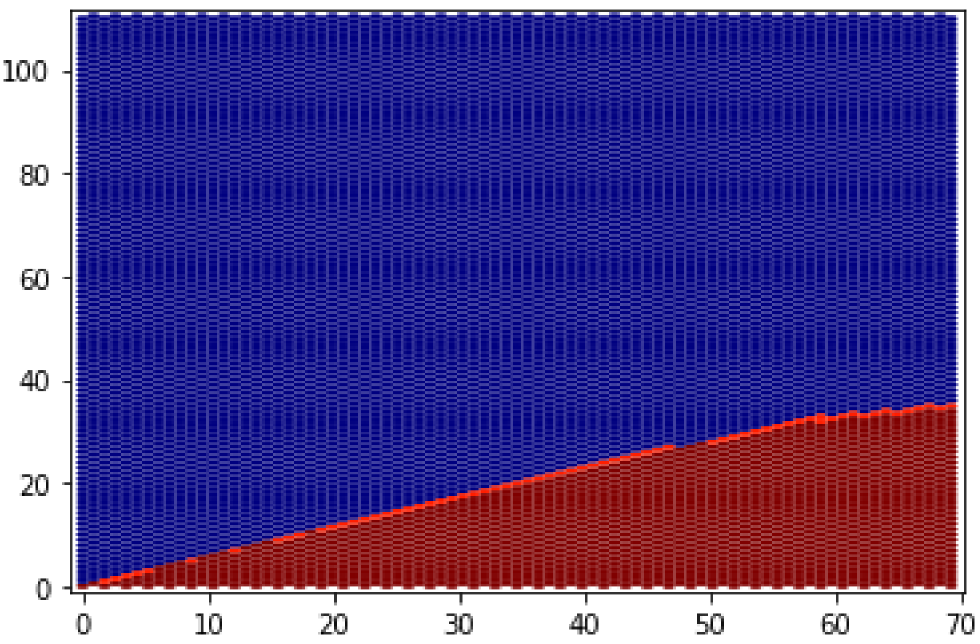
[4]:
```
params['grav_accel'] = 3.7
gfs = GrainFacetSimulator(**params)
gfs.run()
```



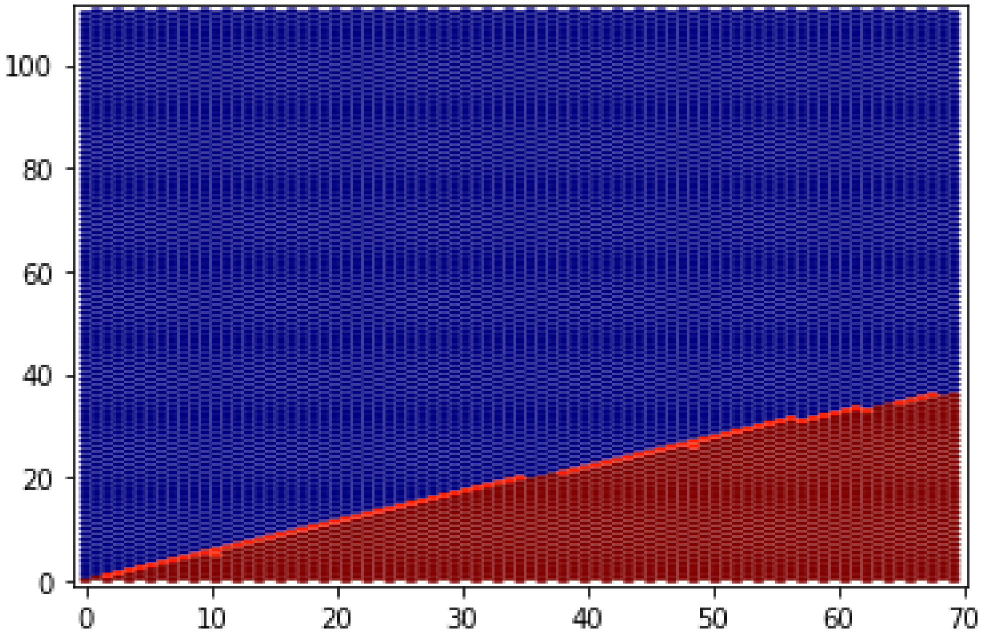
```
Current sim time0.0(0.0%)
Running to...866.0
Running to...1732.0
Running to...2598.0
Running to...3464.0
Running to...4330.0
Running to...5196.0
Running to...6062.0
Running to...6928.0
Running to...7794.0
Running to...8660.0
Running to...9526.0
Running to...10000.0
```



```
Running to...10392.0
Running to...11258.0
Current sim time11258.0(37.526666666666664%)
Running to...12124.0
Running to...12990.0
Running to...13856.0
Running to...14722.0
Running to...15588.0
Running to...16454.0
Current sim time16454.0(54.846666666666664%)
Running to...17320.0
Running to...18186.0
Running to...19052.0
Running to...19918.0
Current sim time19918.0(66.39333333333333%)
Running to...20000.0
```



```
Running to...20784.0
Running to...21650.0
Running to...22516.0
Current sim time22516.0(75.05333333333333%)
Running to...23382.0
Running to...24248.0
Running to...25114.0
Current sim time25114.0(83.71333333333334%)
Running to...25980.0
Running to...26846.0
Running to...27712.0
Current sim time27712.0(92.37333333333333%)
Running to...28578.0
Running to...29444.0
Running to...30000.0
```



The runs with higher and lower gravitational acceleration are nearly identical.

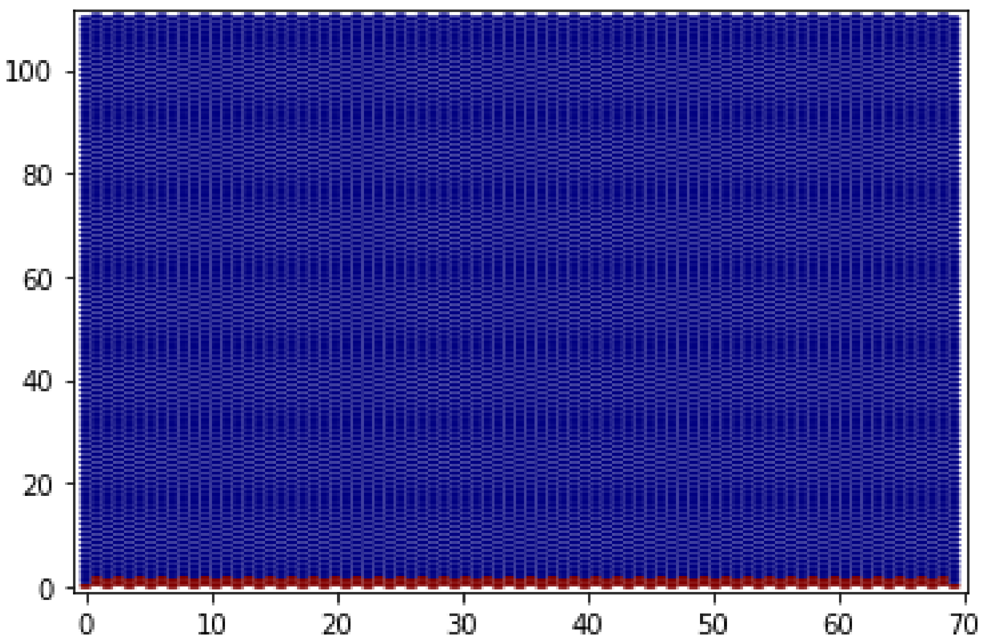
Frictional versus partly elastic behavior

The Lattice Grain cellular representation of grain dynamics includes a frictional parameter f that is analogous to a restitution coefficient. The parameter expresses the relative probability that a collision event (whether head-on or angular) will be frictional, as opposed to elastic. In frictional collisions, the impacting grain(s) cease motion, whereas with elastic collisions, there is a change in motion direction(s).

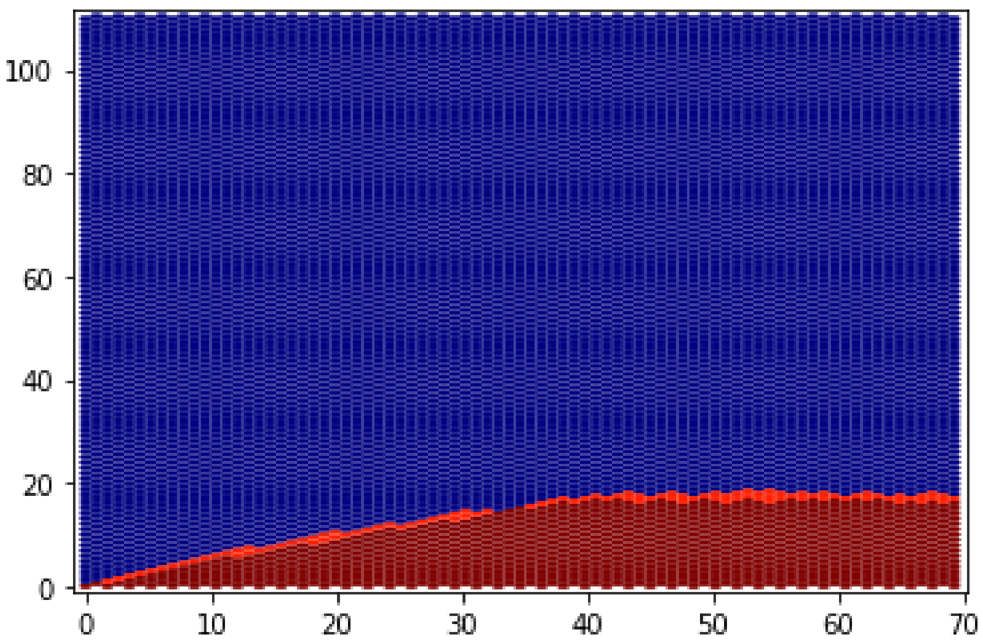
All of the runs explored in the paper use $f = 1$, with no possibility of elastic rebound. This treatment is motivated by two considerations: first, that a moving-regolith cell contains an amalgam of moving particles (as for example would be the case with a handful of grains tossed upward by a burrowing animal), and second that the surfaces that come into contact during a collision event are likely to be rough and, in the case of soil, deformable. Nonetheless, it is worth considering what would happen if elastic rebound did commonly occur during collisions between particles or between particles and rock.

Nino et al. (1994) reported restitution coefficients ranging from 0 to 0.5 from gravel saltation experiments. In the experiment below, a friction coefficient of 0.5 is used, meaning that on average half of collisions will result in rebound, while half will result in the moving material coming to rest. The cell below sets the friction coefficient to 0.5, resets the gravitational acceleration back to 9.8 m/s^2, and runs the model.

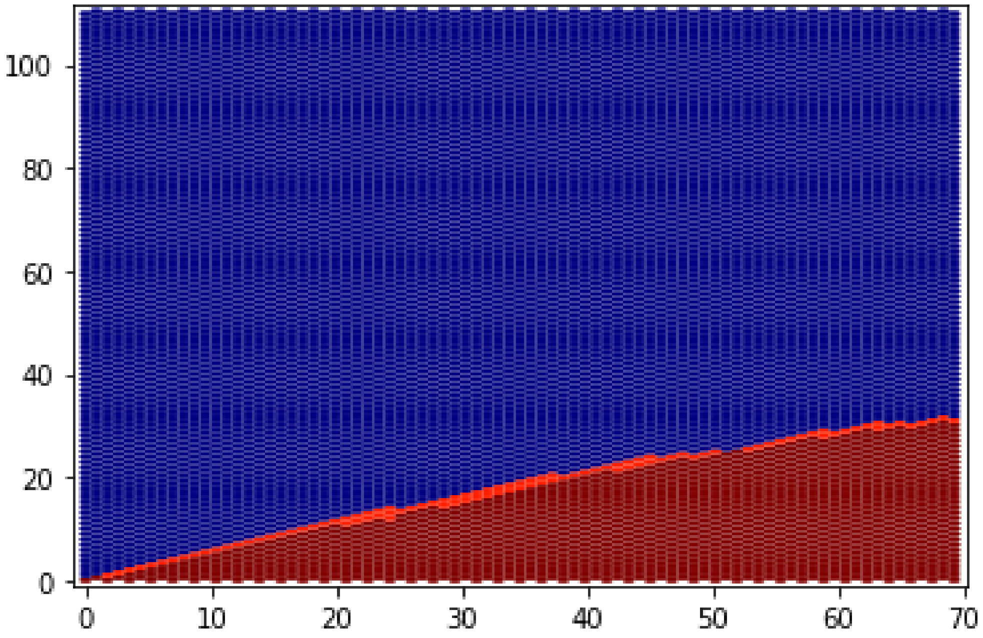
[5]:
```
params['friction_coef'] = 0.5
params['grav_accel'] = 9.8
gfs = GrainFacetSimulator(**params)
gfs.run()
```



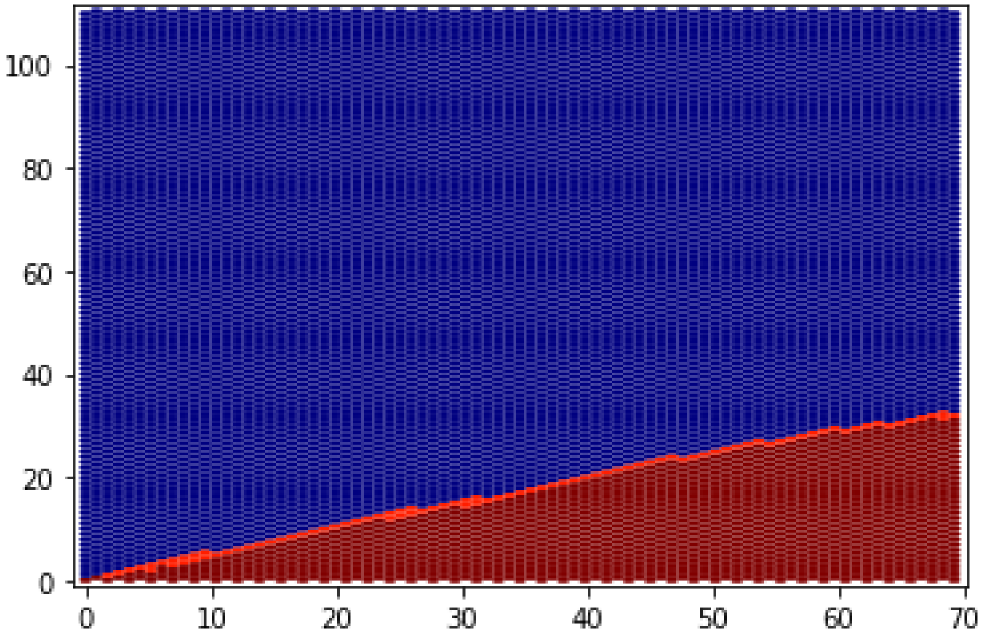
```
Current sim time0.0(0.0%)
Running to...866.0
Running to...1732.0
Running to...2598.0
Running to...3464.0
Running to...4330.0
Running to...5196.0
Running to...6062.0
Running to...6928.0
Running to...7794.0
Running to...8660.0
Current sim time8660.0(28.866666666666667%)
Running to...9526.0
Running to...10000.0
```



```
Running to...10392.0
Running to...11258.0
Running to...12124.0
Running to...12990.0
Current sim time12990.0(43.3%)
Running to...13856.0
Running to...14722.0
Running to...15588.0
Running to...16454.0
Current sim time16454.0(54.846666666666664%)
Running to...17320.0
Running to...18186.0
Running to...19052.0
Current sim time19052.0(63.50666666666667%)
Running to...19918.0
Running to...20000.0
```



```
Running to...20784.0
Current sim time20784.0(69.28%)
Running to...21650.0
Running to...22516.0
Running to...23382.0
Current sim time23382.0(77.94%)
Running to...24248.0
Running to...25114.0
Running to...25980.0
Current sim time25980.0(86.6%)
Running to...26846.0
Running to...27712.0
Current sim time27712.0(92.37333333333333%)
Running to...28578.0
Running to...29444.0
Running to...30000.0
```



The slope angle is slightly lower than in the control run, reflecting the increased mobility of regolith when elastic collisions are frequent. The effect of an increase in the frequency of elastic collisions is similar to the effect of a small increase in the disturbance-frequency parameter d. On the whole, however, the resulting facet slope is quite similar to the control case.

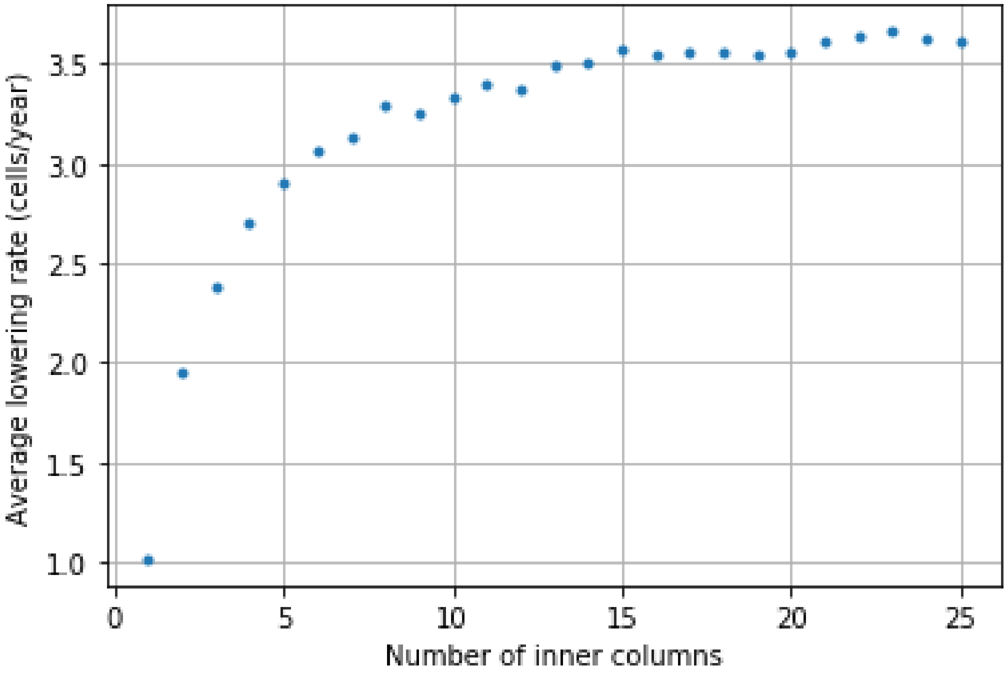
S2. Relationship between domain length, surface roughness, and weathering rate

This section reports a set of tests that examine the predicted relationship between the surface roughness and the rate of descent of a weathering front. The test consists of running the model in "dissolution model" (all rock weathering results in removal of mass, rather than conversion to regolith), with a starting condition consisting of a column of rock that fills the domain except for a single interior row of air cells at the top. The question to be answered is: how does the lowering rate of the surface vary according to the width of the domain?

If there is only one interior column, the surface available for weathering is one cell width. In this case, the expected average descent rate is equal to the given dissolution rate parameter, s, times the cell width, δ. For these experiments, both parameters are set to unity, and the initial rock column height is 1000 cells. Thus, the expected lowering rate is 1.

With more than one interior column, the lowering rate should be faster because the surface area exposed to weathering is greater. These experiments are designed to answer the questions: how much faster, and is there an apparent asymptote as the number of columns increases? To answer these questions, experiments were run with a range of 1 to 25 interior columns, with three replicate runs (using different random seed values) for each. The results are plotted below.

[1]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

x = np.loadtxt('dissolution_width_vs_ero_rate.csv', delimiter=',')
plt.plot(x[:,0], x[:,1], '.')
plt.xlabel('Number of inner columns')
plt.ylabel('Average lowering rate (cells/year)')
plt.grid(True)
```



The results show that, as expected, the lowering rate for a model with a single column is unity, and the lowering rate increases as the number of columns increases. The data show an asymptotic curve, with an asymptotic value of about 3.6.

As discussed in the main text, if the surface were purely level, the exposure of (on average) two faces per grid cell should produce a lowering rate of

$$\dot{\epsilon} = 2\delta w.$$

Surface roughness increases the rate by a factor a, such that the actual rate is

$$\dot{\epsilon} = 2a\delta w.$$

The column-dissolution experiments imply that, as the width of the domain grows much larger than the characteristic scale of disturbance (δ), $2a\delta w \approx 3.6\delta w$. Hence, $a \approx 1.8$.

S3. Obtaining erosion rate from a snapshot of model topography

Building on Tucker et al. (2011), consider the cross-sectional profile of a facet with a uniform dip angle θ. The dip angle of the fault is α and its slip rate is V. We want to know what is the implied *vertical* erosion rate for a given θ, α, and V.

Consider a location in the footwall at a horizontal distance x from the fault trace. The time t taken for rock to reach that location from the fault trace is equal to the distance, x, divided by the horizontal component of slip rate, v_h. This in turn implies that

$$x = v_h t.$$

By simple geometry, the horizontal component of slip rate is

$$v_h = V \cos \alpha.$$

If $\alpha = 60^\circ$, as we'll generally assume here, then $v_h = V/2$.

We can imagine a triangle formed by the cumulative fault slip Vt (the hypotenuse), the horizontal component $v_h = V \cos \alpha$, and the vertical component $v_v = V \sin \alpha$. Thus, the three sides of the triangle are Vt (hypotenuse), $Vt \cos \alpha$ (base), and $Vt \sin \alpha$ (height). If $\alpha = 60^\circ$, the sides of the triangle are Vt, $Vt/2$, and $\sqrt{3}Vt/2$, respectively.

Now consider the case when the facet dip is less than that of the fault dip. This implies that at horizontal position $x = v_h t$, there is an uneroded vertical column of rock of height H, where $\tan \theta = H/x$, and therefore

$$\begin{aligned} H = x \tan \theta &= v_h t \frac{\sin \theta}{\cos \theta} \\ &= Vt \frac{\cos \alpha \sin \theta}{\cos \theta}. \end{aligned}$$

As a check on this math, note that if $\theta = 0$, then $H = 0$ (the surface is flat), whereas if $\theta = \alpha$, then we simply have the height of original (uneroded) triangle $Vt \sin \alpha$.

The cumulative erosion depth h_e is the difference between the total column height if there were no erosion, and the actual column height:

$$\begin{aligned} h_e = Vt \sin \alpha - H &= Vt \sin \alpha - Vt \frac{\cos \alpha \sin \theta}{\cos \theta} \\ &= Vt \left(\sin \alpha - \frac{\cos \alpha \sin \theta}{\cos \theta} \right). \end{aligned}$$

The cumulative vertical erosion h_e at position x is also equal to the vertical erosion rate E_v times the elapsed time, t, which implies

$$E_v = V \left(\sin \alpha - \frac{\cos \alpha \sin \theta}{\cos \theta} \right).$$

We can do a basic sanity check. If $\theta = \alpha$ then the erosion rate should be zero, as it is above (the quantity in parentheses is zero). If $\theta = 0$, then the vertical erosion rate should equal the vertical component of slip. And indeed because the right-hand term in parentheses is zero when $\theta = 0$, we have $E_v = V \sin \alpha$, which is the vertical component of slip rate.

Below, we plot this relationship, and look at some example scenarios.

[1]:

```
def normalized_ero_rate(theta, alpha=60.0):
    """Calculate and return erosion rate over slip rate for given theta and␣
 ↪alpha."""
    alphar = np.radians(alpha)
    thetar = np.radians(theta)
    return np.sin(alphar) - np.cos(alphar) * (np.sin(thetar)
                                              / np.cos(thetar))
```

[2]:

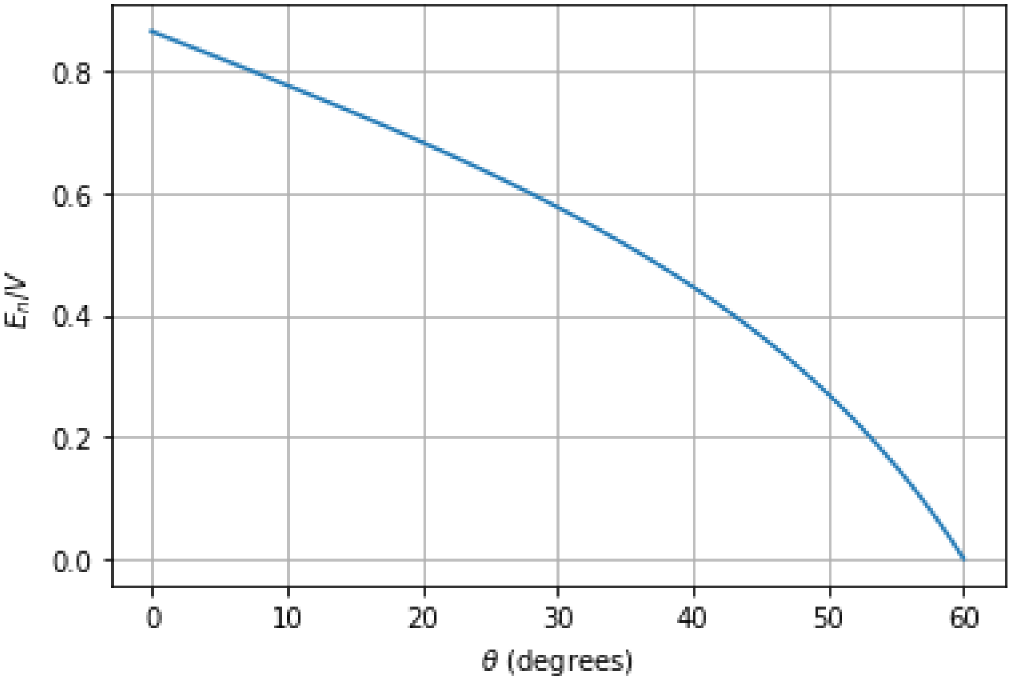
```
import matplotlib.pyplot as plt
import numpy as np
```

[3]:

```
alpha = 60.0
theta = np.arange(0, 61)  # range of facet angles from 0 to 60 degrees
EvV = normalized_ero_rate(theta, alpha)  # corresponding normalized erosion rate
```

[4]:

```
plt.plot(theta, EvV)
plt.xlabel(r'$\theta$ (degrees)')
plt.ylabel(r'$E_n / V$')
plt.grid(True)
```



As expected, when the facet and fault dip are the same, the erosion rate is zero. When the facet angle is zero, meaning the surface is flat, erosion is vertical and its rate is equal to the vertical component of the slip rate. That vertical component is:

$$v_v = V \sin \alpha,$$

which for $\alpha = 60^\circ$ again is $\sqrt{3}/2)V$:

[5]:
```
normalized_ero_rate(0.0)
```

[5]:
```
0.8660254037844386
```

And for $\alpha = 30°$:

[6]:
```
normalized_ero_rate(30.0)
```

[6]:
```
0.5773502691896257
```

Suppose, for example, the slip rate is $\sqrt{3}$ mm/yr. (Because of the grid geometry in the GrainFacet model, this is rate one gets using $\delta = 1$ m and $\tau = 1,000$ yr.) Then for a flat slope, the vertical component of slip rate, v_v, and the vertical erosion rate, E_v, should be identical. Using the result above, we can calculate v_v as:

[7]:
```
V = np.sqrt(3)
V * np.sin(np.radians(60.0))
```

[7]:
```
1.4999999999999998
```

And we can calculate E_v as:

[8]:
```
V * normalized_ero_rate(0.0, 60.0)
```

[8]:
```
1.4999999999999998
```

Indeed we see they are the same, and the maximum erosion rate for this particular scenario is 1.5 mm/yr.

Consider next the example of a 30° facet. Here the vertical erosion rate corresponding to the above slip rate would be:

[9]:
```
V * normalized_ero_rate(30.0, 60.0)
```

[9]:
```
0.9999999999999999
```

So, for a 30° facet, the erosion rate is 2/3 of the theoretical maximum (i.e., what it would be for a flat "facet", where any uplifted material is immediately shaved off).

Just for completeness, we can look at the cases of 15° …

[10]:
```
V * normalized_ero_rate(15.0, 60.0)
```

[10]:
```
1.2679491924311224
```

… and 45° …

[11]:
```
V * normalized_ero_rate(45.0, 60.0)
```

[11]:
```
0.6339745962155612
```

Slope-normal erosion rate

The vertical and slope-normal erosion rates are related by:

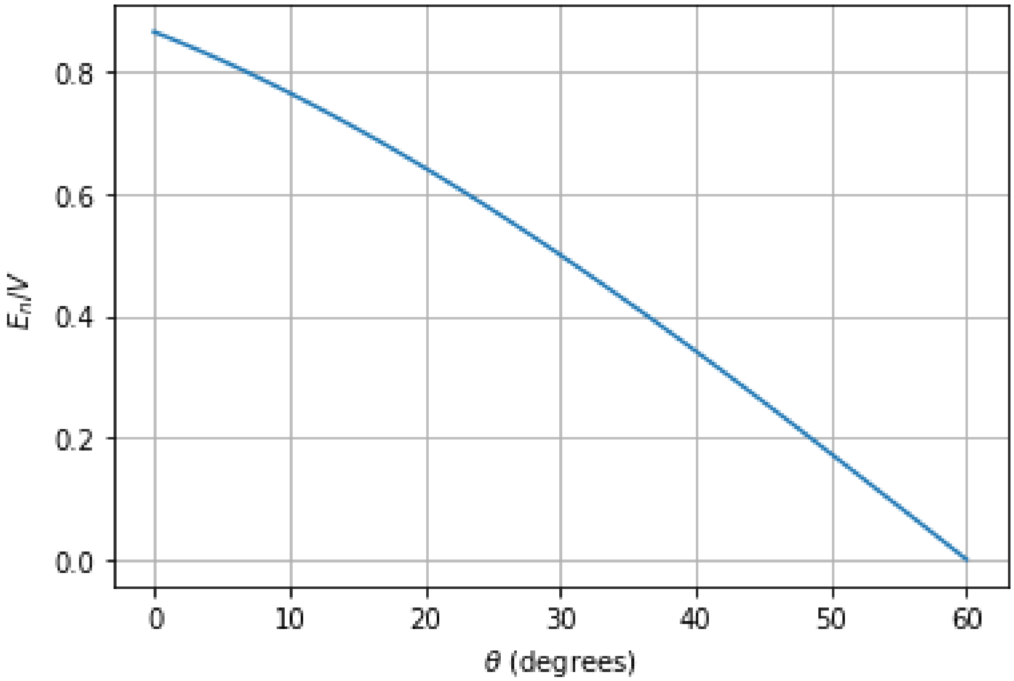
$$\frac{E_n}{E_v} = \cos\theta.$$

If we substitute this above, we find that:

$$E_n = V\left(\sin\alpha\cos\theta - \cos\alpha\sin\theta\right).$$

We plot this below:

[12]:
```
theta = np.arange(0, 61)
thetar = np.radians(theta)
alphar = np.radians(60.0)
EnV = np.sin(alphar) * np.cos(thetar) - np.cos(alphar) * np.sin(thetar)
plt.plot(theta, EnV)
plt.xlabel(r'$\theta$ (degrees)')
plt.ylabel(r'$E_n / V$')
plt.grid(True)
```



For $\theta = \alpha = 60°$, there is no erosion and the fault plane is unmodified. For $\theta = 0$, $E_n = E_v = \sqrt{3}/2$.

Note that this differs from the result in Tucker et al. (2011), which reported the case of an arc-wise erosion vector.

We can simplify the above formula by using two trigonometric identities:

$$\sin \alpha \cos \theta = (1/2)(\sin(a + b) + \sin(a - b)),$$

and

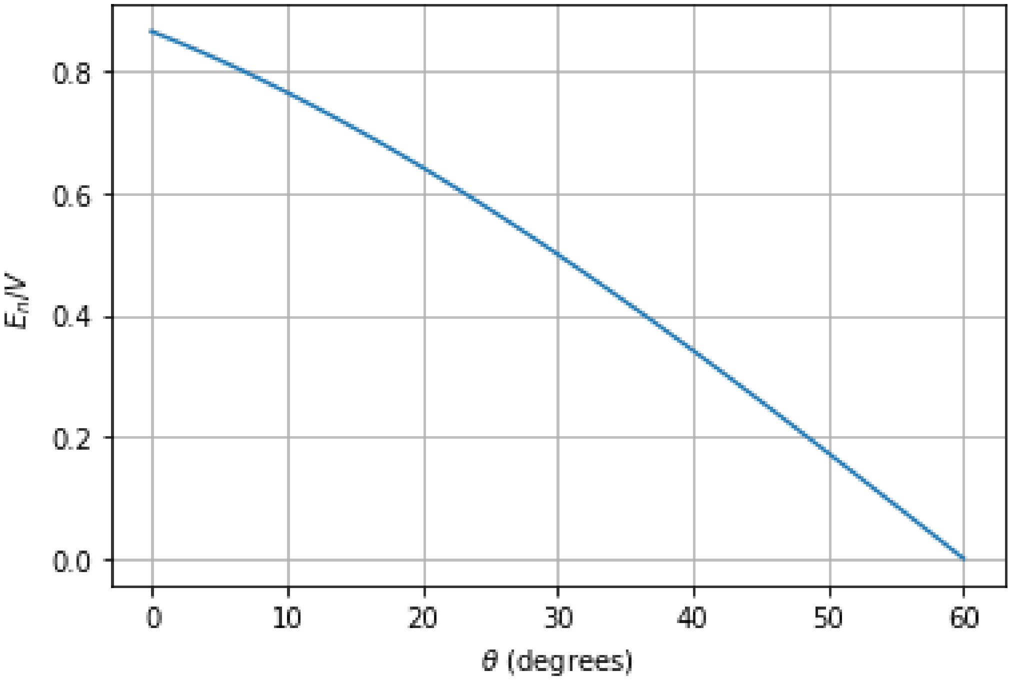
$$\cos \alpha \sin \theta = (1/2)(\sin(a + b) - \sin(a - b)).$$

Subtracting the two, as in the above equation, we get:

$$E_n / V = \sin(\alpha - \theta).$$

Let's just verify it by plotting:

[13]:
```
EnV = np.sin(alphar - thetar)
plt.plot(theta, EnV)
plt.xlabel(r'$\theta$ (degrees)')
plt.ylabel(r'$E_n / V$')
plt.grid(True)
```



Voila: c'est la meme truc.

How to calculate vertical erosion rate from a facet simulation

Suppose you have a snapshot from a GrainFacet simulation run that is in quasi-steady state with respect to tectonic forcing. How do you calculate the erosion rate? It is not quite as simple as it is for models with "vertical uplift" because mass can escape due to tectonic slip out the right side and possibly the top. However, we can use the basic theory above to calculate an effective erosion rate for each model column, and then average them together. Here's how it works.

Horizontal position of a column

Let c denote the column number in a run, starting from column zero at $x = 0$. The spacing between columns is $\sqrt{3}/2$ cell widths. Thus, the x coordinate, in cell widths, is:

$$x(c) = \frac{\sqrt{3}}{2} c.$$

Height of a column

We can ask: what is the effective height of the column at that location? It's easy enough to count the number of nonzero cells in the column. We'll call this count $N(c)$. But how does that translate into height? First, let's assume that the land surface lies at the *center* of the top-most cell (of course, in plots, it *looks* like the surface is at the top of the filled cell, but this assumption is more consistent with how the fault is approximated in the model).

If we happen to be in an even-numbered column, the center of the bottom-most cell is at $y = 0$, and so the center of the top cell is $N - 1$. But if we happen to be in an odd-numbered column, the column is shifted up by half a cell width. So for these, the center of the top cell is $N - 1/2$. We can generalize as follows to get the height, $H(c)$, of any arbitrary column:

$$H(c) = N - \frac{1}{1 + (c \mod 2)}.$$

This expression sets $H = N - 1$ for even columns, and $H = N - 0.5$ for odd columns.

Calculating erosion rate at a column

To calculate the erosion rate at a single column, we start with the expression for E_v derived above:

$$E_v/V = \sin\alpha - \frac{\cos\alpha\sin\theta}{\cos\theta}.$$

Using $\alpha = 60°$, and using numerical values of $\sin\alpha$ and $\cos\alpha$, this becomes:

$$E_v/V = \frac{\sqrt{3}}{2} - \frac{1}{2}\frac{\sin\theta}{\cos\theta}.$$

Noting that $\sin\theta/\cos\theta = \tan\theta = H/x$, we have:

$$\frac{E_v}{V} = \frac{\sqrt{3}}{2} - \frac{H}{2x}.$$

This is the formula we use to calculate vertical erosion rate, E_v, from column height, H, and horizontal position x.

The calculation is encapsulated in the following function:

[14]:
```
def vert_ero_rate(c, N):
    """Normalized vertical erosion rate in column c with N solid cells.

    Parameters
    ----------
    c : int
        Column number
    N : int
        Number of solid cells in column
    """
    x = 0.5 * np.sqrt(3) * c
    if N <= 1:  # if only one cell, assume it's totally eroded
        H = 0.0
    else:
```

```
        H = N - 1.0 / (1.0 + (c % 2))
    return 0.5 * np.sqrt(3) - 0.5 * H / x
```

Examples

Suppose you have only 1 cell in column 2. That's the minimum number of cells possible in the numerical model: the bottom-most cell is a boundary, and always contains rock. The erosion rate should equal the vertical component of slip rate, so E_v/V should equal $\sin 60^\circ = \sqrt{3}/2 \approx 0.866$. Let's try it:

[15]:
```
vert_ero_rate(2, 1)
```

[15]:
```
0.8660254037844386
```

The result should be the same in odd-numbered columns, say column 7:

[16]:
```
vert_ero_rate(7, 1)
```

[16]:
```
0.8660254037844386
```

If we had a 60° slope, columns 3 and 4 would have 5 and 7 solid cells, respectively, and the erosion rate should be zero:

[17]:
```
vert_ero_rate(3, 5)
```

[17]:
```
0.0
```

[18]:
```
vert_ero_rate(4, 7)
```

[18]:
```
-1.1102230246251565e-16
```

The latter result tells us there is a slight numerical inaccuracy: probably a subtraction-of-close-numbers error. To diagnose it, let's see how H/x compares with $\sqrt{3}$:

[19]:
```
c = 4
N = 7
H = N - 1.0 / (1.0 + (c % 2))
x = 0.5 * np.sqrt(3) * c
print('Square root of 3 is approxately: ' + str(np.sqrt(3)))
print('H is: ' + str(H))
print('x is: ' + str(x))
print('H / x is: ' + str(H/x))
```

```
Square root of 3 is approxately: 1.7320508075688772
H is: 6.0
x is: 3.4641016151377544
H / x is: 1.7320508075688774
```

We see that that there is a slight numerical error in H/x: it is off in the 16th decimal place (hence an error of order 10^{-16}). This seems perfectly acceptable.

As a last example, consider columns 5 and 6 on a 30° facet. The number of filled cells would be 3 and 4, respectively. From the analysis above, we expect a vertical erosion rate of $\sqrt{3}/3$, which is about:

[20]:
```
np.sqrt(3.0)/3.0
```

[20]:
```
0.5773502691896257
```

[21]:
```
vert_ero_rate(5, 3)
```

[21]:
```
0.5773502691896257
```

[22]:
```
vert_ero_rate(6, 4)
```

[22]:
```
0.5773502691896257
```

Summary

To summarize, the method for calculating average vertical erosion rate in each column (from columns 2 to the right-most inner column) of a snapshop of a GrainFacet model run consists of the following steps:

(1) Calculate the position of the column: $x(c) = \sqrt{3}c/2$.

(2) Calculate the height of the column: $H(N, c) = N - \frac{1}{1+(c \mod 2)}$, if $N > 1$, and $H = 0$ otherwise.

(3) Calculate the column's erosion rate: $E_v = \frac{\sqrt{3}\delta}{\tau}\left(\frac{\sqrt{3}}{2} - \frac{H}{2x}\right)$.

(Note that step 3 includes that $V = \sqrt{3}\delta/\tau_s$, where δ is cell width, τ_s is the slip interval, and $\sqrt{3}\delta$ is the slip distance.)

Once the individual columns have been calculated, their vertical erosion rates are averaged.

S4. Analysis and Plotting

S4.1. Mean and standard deviation of facet and colluvial wedge angles from Wilkinson et al. (2015)

Data below are from figure 5 of Wilkinson et al. (2015), which reports facet, scarp, and colluvial wedge slope angles along the Campo Felice fault in the Italian Central Apennines.

[1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:

```
hwdip = np.array([35.273, 34.000, 34.794, 35.248, 34.959, 37.323, 37.816, 34.
 ↪052, 36.514,
                  34.526, 37.496, 34.974, 39.445, 36.003, 35.286, 33.962, 32.
 ↪964, 34.315,
                  32.756, 32.337, 29.306, 38.188, 37.278, 36.432, 47.646])
fwdip = np.array([38.773, 36.761, 36.851, 36.385, 38.856, 40.236, 43.866, 35.
 ↪715, 39.876,
                  38.502, 42.782, 50.129, 43.082, 38.386, 40.335, 45.245, 37.
 ↪301, 35.804,
                  35.245, 36.919, 36.887, 39.073, 41.508, 44.534, 55.048])
```

[3]:

```
np.mean(hwdip)
```

[3]:

```
35.715720000000005
```

[4]:

```
np.std(hwdip)
```

[4]:

```
3.2233977665810962
```

[5]:

```
np.mean(fwdip)
```

[5]:

```
40.32396
```

[6]:

```
np.std(fwdip)
```

[6]:

```
4.631058958640021
```

S4.2. Relationship between facet angle and dissolution-rate parameter

This notebook does calculation and plotting to test whether the GrainFacetModel, in its “dissolution” configuration—that is, detachment-limited with no production of sediment—obeys the geometric formulation in Tucker et al. (2011). Modified to account for the slope-normal instead of arc-wise erosion vector, the formulation says that the sine difference in angle between the fault plane (α) and the facet (θ) equals the ratio of slope-normal erosion rate rate, E_n, to slip rate, V:

$$\sin(\alpha - \theta) = E_n / V$$

For the case of pure dissolution (no regolith production), E_n corresponds to the slope-normal erosion rate due to dissolution. One problem, however, is that we do not know the erosion rate *a priori*. Instead, we set the dissolution transition probability, s (dimensions of $1/\text{time}$). One might expect the resultant erosion rate to simply be cell size times transition rate, δs. This doesn’t quite work, for two reasons. First, there is a geometric correction related to the hex lattice geometry (a “cut” through the lattice would tend to produce two faces per cell). This first effect is straightforward to describe: because of the additive property of Poisson processes, the expected rate, if cells always expose exactly two faces, is simply $2\delta s$.

Second, surface roughness makes the actual slope-normal erosion rate a bit higher than the expected value of $2\delta s$, by a factor a. We therefore have

$$\sin(\alpha - \theta) = 2a\delta s / V.$$

Of note here is that a separate analysis of dissolution of a column predicts that surface roughness increases the lowering rate by a factor of $a \approx 1.8$, so that the effective rate is approximately $E_n = 3.6\delta s$. That factor comes from tests of vertical lowering; We can test whether it also applies to the simulated facet profiles.

We start by reading the results from a series of runs with s ranging from 4×10^{-5} yr^{-1} to 48×10^{-5} yr^{-1}, with $\delta = 0.5$ m, $V = 0.001$ m/yr, and a domain width of 80 cells (≈ 69 m). Then we’ll add several runs with a longer domain to test the degree to which slope length matters.

[1]:

```
import csv
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

[2]:

```
filename = 'data_analysis_dissolution20190604.csv'
```

[3]:

```
# Count number of lines in file
num_lines = len(open(filename).readlines(  ))

# Create data arrays
diss_param = np.zeros(num_lines - 2)  # skip 2 header lines
facet_angle = np.zeros(num_lines - 2)
ero_rate = np.zeros(num_lines - 2)  # slope-normal erosion rate, m/yr (NOT cells/
 ↪yr)

# Read data
with open(filename, 'r') as csvfile:
    myreader = csv.reader(csvfile)
    i = 0
```

```
    for row in myreader:
        print(' '.join(row))
        if i >= 2:
            diss_param[i-2] = row[1]
            facet_angle[i-2] = row[2]
            ero_rate[i-2] = row[5]
        i += 1
```

```
Run name Dissolution rate parameter (1/y) Slope angle (deg) Slope gradient (m/m)
Intercept Average vertical erosion rate (m/y) Fractional soil cover
dissolve_dr36 36.0e-5 14.9569455272 0.267143960973 2.96105160662
0.000659129383784 0.0
dissolve_dr40 40.0e-5 11.9060025534 0.210842471723 2.62025316456
0.000689201405244 0.0
dissolve_dr48 48.0e-5 5.32663961368 0.093236165672 3.02336903603 0.0007411955448
0.0
dissolve_dr4 4.0e-5 56.3002742582 1.49945229357 1.34858812074 9.83170797729e-05
0.0
dissolve_dr24 24.0e-5 35.2759196107 0.707408676957 2.43135345667
0.000453180781893 0.0
dissolve_dr12 12.0e-5 47.1173946606 1.07678354976 2.58519961052
0.000293422087718 0.0
dissolve_dr32 32.0e-5 23.4877741013 0.434558673889 2.25024342746
0.000584352261214 0.0
dissolve_dr28 28.0e-5 30.1471422173 0.580779516163 1.80525803311
0.000531503000746 0.0
dissolve_dr44 44.0e-5 6.01398172095 0.105350964407 3.57838364167
0.000726271369592 0.0
dissolve_dr8 8.0e-5 51.2908623697 1.24779616114 2.21811100292 0.00020483469493
0.0
dissolve_dr20 20.0e-5 38.7117594638 0.801488100081 1.1723466407
0.000431691113396 0.0
dissolve_dr16 16.0e-5 42.8380938761 0.927245894841 2.94255111977
0.000361581616457 0.0
```

Read a set of runs with a longer domain:

[6]:

```
filename = 'dissolution_long_domain20190311.csv'
```

[7]:

```
# Count number of lines in file
num_lines = len(open(filename).readlines(  ))

# Create data arrays
diss_param_long = np.zeros(num_lines - 2)  # skip 2 header lines
facet_angle_long = np.zeros(num_lines - 2)

# Read data
with open(filename, 'r') as csvfile:
    myreader = csv.reader(csvfile)
```

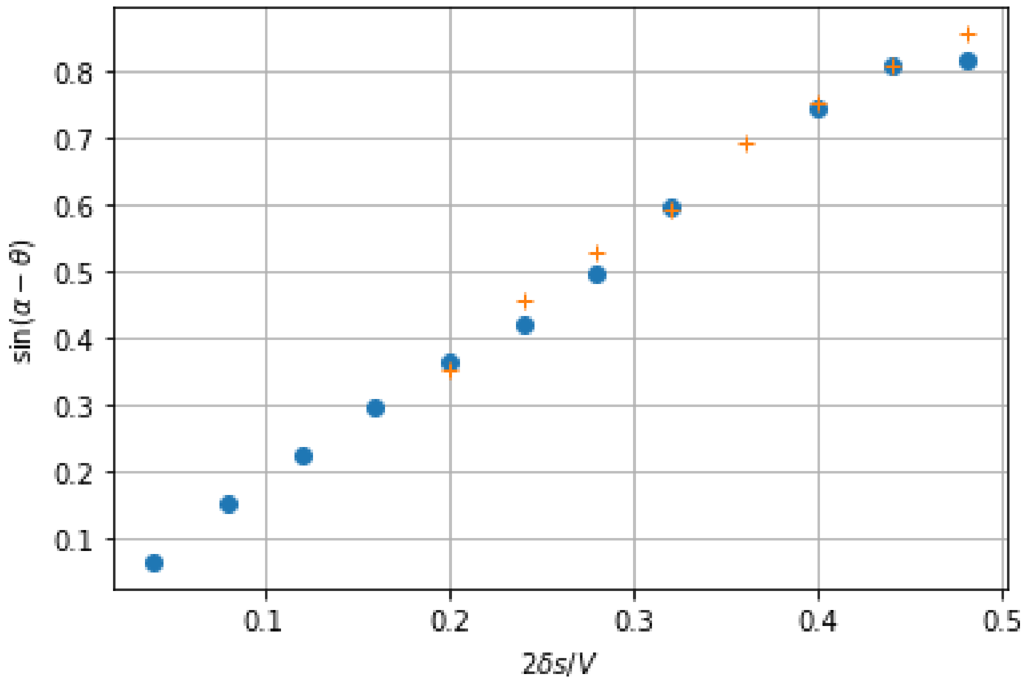
```
    i = 0
    for row in myreader:
        print(' '.join(row))
        if i >= 2:
            diss_param_long[i-2] = row[0]
            facet_angle_long[i-2] = row[2]
        i += 1
```

```
Landlab version 1.7.0+15.g1764eb3f
Dissolution rate parameter (1/yr) Gradient (m/m) Slope angle (deg)
0.0002 0.82404884912 39.4901895413
0.00024 0.644706866237 32.810153595
0.00028 0.534296429598 28.115431014
0.00032 0.441043524674 23.7995670076
0.00036 0.288452281903 16.0903267272
0.0004 0.201972724765 11.4185726704
0.00044 0.10976357524 6.26391395369
0.00048 0.0224836295957 1.28800007937
```

Plot $\sin(\alpha - \theta)$ versus $2\delta s / V$ to test the prediction that the relationship should be linear:

[8]:

```
# Quick data plot to check linearity
V = 0.001
expected_sin_angle_diff = 2 * delta * diss_param / V
expected_sin_angle_diff_long = 2 * delta * diss_param_long / V
plt.plot(expected_sin_angle_diff, np.sin(np.radians(60.0 - facet_angle)), 'o')
plt.plot(expected_sin_angle_diff_long, np.sin(np.radians(60.0 -␣
 ↪facet_angle_long)), '+')
plt.xlabel(r'$2 \delta s / V$')
plt.ylabel(r'$\sin ( \alpha - \theta )$')
plt.grid(True)
```



The linearity of the above plot means that part of the theory works: there is indeed a linear relation between the sine of the angle difference and the ratio of erosion to slip rate, and the relationship is insensitive to slope length. However, the fact that the observed slope difference is consistently greater than its predicted value implies that there is indeed a roughness effect that needs a correction.

Next, we'll fit the coefficient a in:

$$\sin(\alpha - \theta) = \frac{2a\delta}{V}s.$$

[9]:

```
# To use linalg.lstsq, we need to transpose $w$
x = expected_sin_angle_diff[:,np.newaxis]
y = np.sin(np.radians(60.0 - facet_angle))  # angle difference

# Fit with zero intercept (the "rcond" parameter supresses a
# warning)
a, _, _, _ = np.linalg.lstsq(x, y, rcond=None)
print(a)
```

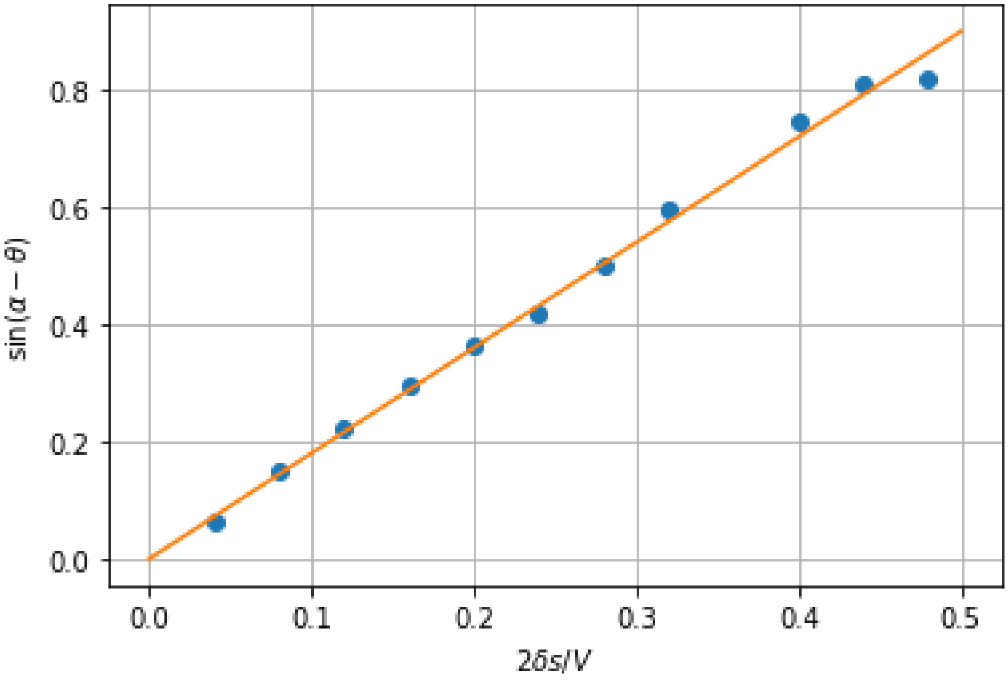
```
[1.7977213]
```

This value of $a \approx 1.8$ matches a separate set of experiments on dissolution of a flat surface, where the actual rate is about 3.6 times δs, or 1.8 times $2\delta s$.

Verify that the fitting parameter makes sense by plotting. The line should go through the data.

[10]:

```
plt.plot(x, y, 'o')
plt.plot([0, 0.5], [0, a * 0.5])
plt.grid(True)
plt.xlabel(r'$2 \delta s / V$')
plt.ylabel(r'$\sin ( \alpha - \theta )$')
plt.grid(True)
```



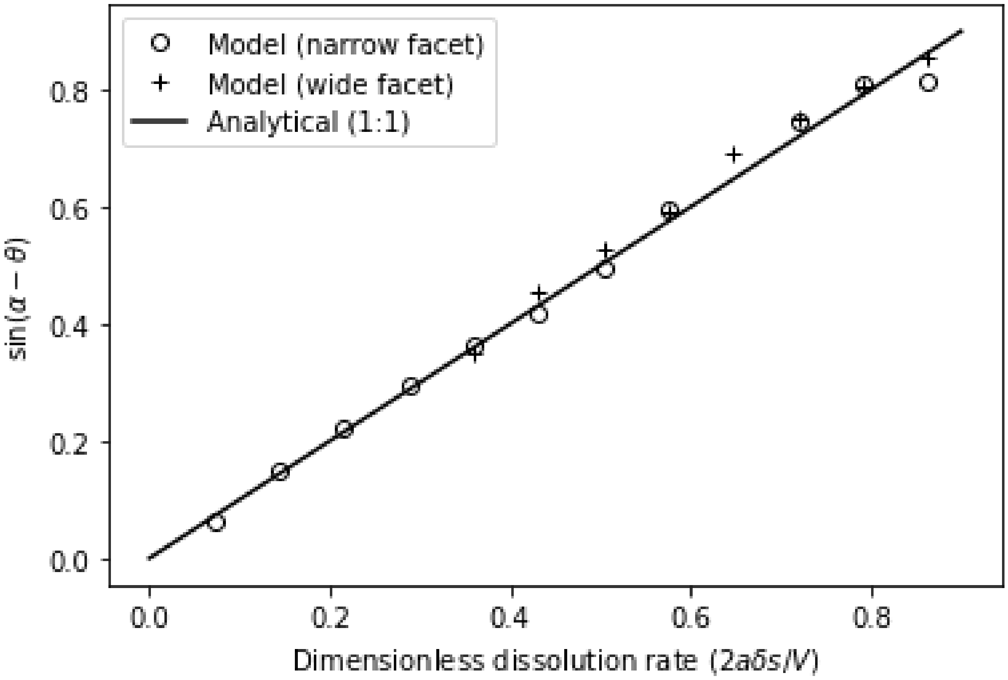
The next step is to create a version of this plot that includes the roughness-correction factor $a \approx 1.8$. We define a dimensionless dissolution-erosion rate as:

$$S' = 2a\delta/V.$$

[11]:

```
# expected dimensionless dissolution-erosion rate
Sprime = 1.8 * 2.0 * delta * diss_param / V
Sprime_long = 1.8 * 2.0 * delta * diss_param_long / V

sin_ang_diff = np.sin(np.radians(60.0 - facet_angle))
sin_ang_diff_long = np.sin(np.radians(60.0 - facet_angle_long))

plt.plot(Sprime, sin_ang_diff, 'ko', fillstyle='none')
plt.plot(Sprime_long, sin_ang_diff_long, 'k+')
plt.plot([0.0, 0.9], [0.0, 0.9], 'k')  # one-to-one line
plt.xlabel(r'Dimensionless dissolution rate ($2 a \delta s / V$)')
plt.ylabel(r'$\sin (\alpha - \theta)$')
plt.legend(['Model (narrow facet)', 'Model (wide facet)', 'Analytical (1:1)'])
```

```
plt.savefig('../modeling-shape-evolution/Figures/angle_vs_dissolution.pdf')
```



Visualizing predicted dip angle

The following block of code produces plots of the predicted facet angle for different values of *s*. Recall that the predicted relationship is:

$$\sin(\alpha-\theta) = 2a\delta s/V = S'.$$

Therefore,

$$\alpha-\theta = \sin^{-1} S',$$

or

$$\theta = \alpha - \sin^{-1} S'.$$

[18]:
```
import matplotlib.pyplot as plt
import numpy as np
from landlab.io.native_landlab import load_grid
from grainhill import plot_hill
%matplotlib inline

run_dir = '../../ModelRuns/Dissolution/'
runs = ['dissolve_dr8', 'dissolve_dr20', 'dissolve_dr32', 'dissolve_dr44',␣
 ↪'dissolve_dr_long20', 'dissolve_dr_long32']
xmax = [70.0, 70.0, 70.0, 70.0, 140., 140.0]
```

```
s = np.array([8.0e-5, 20.0e-5, 32.0e-5, 44.0e-5, 20.0e-5, 32.0e-5])
sprime = 3.6 * s * delta / V
theta_pred = np.radians(60.0) - np.arcsin(sprime)

for i in range(len(runs)):
    run_name = runs[i]
    g = load_grid(run_dir + run_name + '/' + run_name + '.grid')
    plot_hill(g, show=False)
    x = np.array([0., xmax[i]])
    y = x * np.tan(theta_pred[i])
    plt.plot(x, y, 'r')
    plt.savefig(run_name + '.png', bbox_inches='tight')
    plt.clf()
```

```
<Figure size 432x288 with 0 Axes>
```

Note that the above code for plotting sends output to a set of .png files.

S4.3. Facet slope angle as a function of w' and d'

The code in this document plots facet slope angle as a function of dimensionless weathering efficiency, w', and dimensionless disturbance efficiency, d'.

[1]:
```
import csv
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Read in data from 124 model runs that have already been compiled into a .csv file. Read into arrays for the disturbance-rate parameter, the weathering-rate parameter, and the resulting facet slope angle.

[2]:
```
filename = 'slope_analysis20180826.csv'
```

[3]:
```
# Count number of lines in file
num_lines = len(open(filename).readlines(  ))

# Create data arrays
dist_param = np.zeros(num_lines - 1)  # skip 1 header line
weath_param = np.zeros(num_lines - 1)
facet_angle = np.zeros(num_lines - 1)

# Read data
with open(filename, 'r') as csvfile:
    myreader = csv.reader(csvfile)
    i = 0
    for row in myreader:
        print(', '.join(row))
        if i >= 1:
            dist_param[i-1] = row[1]
            weath_param[i-1] = row[2]
            facet_angle[i-1] = row[3]
        i += 1
```

```
Run name, Disturbance rate parameter, Weathering rate parameter, Slope angle,
Slope gradient
d-30w-23, 0.001, 0.00501187233627, 29.0636804385, 0.555762878404
d-40w-36, 0.0001, 0.000251188643151, 48.7387796742, 1.13983109863
d-40w-31, 0.0001, 0.000794328234724, 35.5636262297, 0.714969885448
d-30w-24, 0.001, 0.00398107170553, 28.5455444664, 0.54398538265
d-40w-38, 0.0001, 0.000158489319246, 51.6253692569, 1.26283425238
d-30w-12, 0.001, 0.063095734448, 28.8849563451, 0.55168713405
d-30w-15, 0.001, 0.0316227766017, 28.7180031788, 0.547892475513
d-30w-14, 0.001, 0.0398107170553, 28.7539123029, 0.548707624384
d-40w-39, 0.0001, 0.000125892541179, 53.5138751285, 1.35210710802
d-30w-13, 0.001, 0.0501187233627, 28.6312243124, 0.545924874791
d-30w-25, 0.001, 0.00316227766017, 28.7390563482, 0.548370321403
d-40w-30, 0.0001, 0.001, 34.147478064, 0.67826007768
```

```
d-40w-37, 0.0001, 0.000199526231497, 50.4526909383, 1.21105824479
d-30w-22, 0.001, 0.0063095734448, 28.1363461643, 0.534765767835
d-30w-40, 0.001, 0.0001, 54.5622697427, 1.40517611037
d-20w-40, 0.01, 0.0001, 52.7881783203, 1.31688705509
d-20w-22, 0.01, 0.0063095734448, 19.6126352524, 0.356332490872
d-10w-32, 0.1, 0.00063095734448, 24.0932111102, 0.447179427095
d-20w-25, 0.01, 0.00316227766017, 21.5391845182, 0.394700704964
d-10w-35, 0.1, 0.000316227766017, 37.0188110111, 0.754068922666
d-20w-13, 0.01, 0.0501187233627, 18.7694432138, 0.339832753384
d-20w-14, 0.01, 0.0398107170553, 18.8444856762, 0.341294399635
d-20w-15, 0.01, 0.0316227766017, 18.5569428376, 0.335700791866
d-20w-12, 0.01, 0.063095734448, 18.6379581219, 0.337274872444
d-20w-24, 0.01, 0.00398107170553, 20.7589237896, 0.379044224928
d-10w-34, 0.1, 0.000398107170553, 33.9894462871, 0.674240550489
d-20w-23, 0.01, 0.00501187233627, 20.1041391448, 0.366029951577
d-10w-33, 0.1, 0.000501187233627, 28.0385730264, 0.532573298458
d-10w-11, 0.1, 0.0794328234724, 3.75032803171, 0.0655492126461
d-20w-39, 0.01, 0.000125892541179, 50.9098267664, 1.23093101209
d-10w-16, 0.1, 0.0251188643151, 3.94271849627, 0.0689222424563
d-10w-29, 0.1, 0.00125892541179, 11.8443103294, 0.209718128453
d-10w-20, 0.1, 0.01, 3.7551388438, 0.0656335383913
d-20w-30, 0.01, 0.001, 28.4074187176, 0.540865330075
d-10w-27, 0.1, 0.00199526231497, 6.51524622868, 0.114205167659
d-10w-18, 0.1, 0.0158489319246, 3.76796741664, 0.0658584070453
d-20w-37, 0.01, 0.000199526231497, 47.9996320968, 1.11059817361
d-10w-26, 0.1, 0.00251188643151, 6.32281123401, 0.110804029267
d-10w-19, 0.1, 0.0125892541179, 4.23914456684, 0.0741223300805
d-20w-36, 0.01, 0.000251188643151, 45.8914830876, 1.03161305889
d-10w-21, 0.1, 0.00794328234724, 3.70061321809, 0.0646778466117
d-20w-31, 0.01, 0.000794328234724, 30.8038673283, 0.596211127545
d-20w-38, 0.01, 0.000158489319246, 49.2449516123, 1.16035036331
d-10w-17, 0.1, 0.0199526231497, 3.74551716666, 0.0654648869008
d-10w-28, 0.1, 0.00158489319246, 9.64025791799, 0.169860159528
d-10w-10, 0.1, 0.1, 3.9683639264, 0.0693719797644
d-40w-15, 0.0001, 0.0316227766017, 29.9528704232, 0.576254034501
d-40w-12, 0.0001, 0.063095734448, 29.8621109244, 0.57414589087
d-30w-38, 0.001, 0.000158489319246, 51.4515372985, 1.25499195807
d-30w-31, 0.001, 0.000794328234724, 35.2641108947, 0.707099482558
d-40w-24, 0.0001, 0.00398107170553, 30.0, 0.57735026919
d-40w-23, 0.0001, 0.00501187233627, 29.8621109244, 0.57414589087
d-30w-36, 0.001, 0.000251188643151, 48.2700560713, 1.12119510893
d-40w-40, 0.0001, 0.0001, 54.7449923755, 1.41470491959
d-30w-37, 0.001, 0.000199526231497, 49.4220826916, 1.16763048599
d-40w-22, 0.0001, 0.0063095734448, 30.0386370201, 0.578249743806
d-40w-25, 0.0001, 0.00316227766017, 30.6741840989, 0.593147292134
d-30w-30, 0.001, 0.001, 33.3446681103, 0.657993790237
d-40w-13, 0.0001, 0.0501187233627, 30.0, 0.57735026919
d-30w-39, 0.001, 0.000125892541179, 53.1172864626, 1.33271218662
```

```
d-40w-14, 0.0001, 0.0398107170553, 29.9069062806, 0.575185908395
d-30w-18, 0.001, 0.0158489319246, 28.9466550832, 0.553092563138
d-40w-32, 0.0001, 0.00063095734448, 39.287572793, 0.818128380479
d-30w-27, 0.001, 0.00199526231497, 29.8512086704, 0.573892913634
d-30w-20, 0.001, 0.01, 28.9010050782, 0.552052545613
d-40w-35, 0.0001, 0.000316227766017, 46.002050409, 1.0356044775
d-30w-29, 0.001, 0.00125892541179, 31.2414735452, 0.606611302794
d-30w-16, 0.001, 0.0251188643151, 28.8231842116, 0.550281704962
d-30w-11, 0.001, 0.0794328234724, 28.6870272302, 0.54718976097
d-30w-10, 0.001, 0.1, 28.6907453123, 0.547274086715
d-30w-28, 0.001, 0.00158489319246, 31.1271385335, 0.603884770364
d-30w-17, 0.001, 0.0199526231497, 28.7093317618, 0.547695715441
d-40w-34, 0.0001, 0.000398107170553, 44.1117448863, 0.969464984632
d-30w-21, 0.001, 0.00794328234724, 29.014438909, 0.554638535134
d-30w-19, 0.001, 0.0125892541179, 28.7539123029, 0.548707624384
d-30w-26, 0.001, 0.00251188643151, 29.2442393247, 0.559894839922
d-40w-33, 0.0001, 0.000501187233627, 41.6951981048, 0.89081717289
d-20w-26, 0.01, 0.00251188643151, 22.7248064928, 0.418817868107
d-10w-36, 0.1, 0.000251188643151, 40.5083326112, 0.854332233776
d-20w-19, 0.01, 0.0125892541179, 18.8055296764, 0.340535467928
d-20w-21, 0.01, 0.00794328234724, 19.5297075312, 0.354702193131
d-10w-31, 0.1, 0.000794328234724, 19.8153040547, 0.360323909481
d-20w-17, 0.01, 0.0199526231497, 18.9093721731, 0.342559285814
d-10w-38, 0.1, 0.000158489319246, 46.7690877623, 1.06374116783
d-20w-28, 0.01, 0.00158489319246, 24.3289882389, 0.452126537484
d-20w-10, 0.01, 0.1, 19.2014554869, 0.348265327909
d-20w-11, 0.01, 0.0794328234724, 18.7838796544, 0.340113839201
d-20w-16, 0.01, 0.0251188643151, 18.9151374287, 0.342671720141
d-10w-39, 0.1, 0.000125892541179, 50.5510749115, 1.21530264064
d-20w-29, 0.01, 0.00125892541179, 26.2081400274, 0.492237483644
d-20w-20, 0.01, 0.01, 19.3463933239, 0.351104294666
d-10w-30, 0.1, 0.001, 16.7437117152, 0.300846150494
d-20w-27, 0.01, 0.00199526231497, 22.8589507218, 0.421572509119
d-10w-37, 0.1, 0.000199526231497, 43.8323100193, 0.960048609745
d-20w-18, 0.01, 0.0158489319246, 19.0174044917, 0.344667429445
d-10w-15, 0.1, 0.0316227766017, 3.85615354631, 0.0674043790417
d-10w-12, 0.1, 0.063095734448, 3.82729462492, 0.0668984245702
d-10w-24, 0.1, 0.00398107170553, 4.11579393387, 0.0719579692856
d-20w-34, 0.01, 0.000398107170553, 39.8360328571, 0.834234597823
d-10w-23, 0.1, 0.00501187233627, 4.37046168319, 0.0764272337842
d-20w-33, 0.01, 0.000501187233627, 38.109572176, 0.784369973794
d-10w-40, 0.1, 0.0001, 52.1438341144, 1.28658600396
d-10w-22, 0.1, 0.0063095734448, 3.95393856726, 0.0691190025286
d-20w-32, 0.01, 0.00063095734448, 33.7729974652, 0.668759377048
d-10w-25, 0.1, 0.00316227766017, 4.86008184482, 0.0850284598003
d-20w-35, 0.01, 0.000316227766017, 43.5256901354, 0.949817085988
d-10w-13, 0.1, 0.0501187233627, 3.65410075817, 0.0638626977409
d-10w-14, 0.1, 0.0398107170553, 3.7551388438, 0.0656335383913
```

```
d-40w-11, 0.0001, 0.0794328234724, 30.0, 0.57735026919
d-40w-16, 0.0001, 0.0251188643151, 29.9528704232, 0.576254034501
d-40w-29, 0.0001, 0.00125892541179, 33.1396370333, 0.652878028358
d-40w-20, 0.0001, 0.01, 29.9673765948, 0.576591337482
d-30w-35, 0.001, 0.000316227766017, 45.1395123886, 1.00488179764
d-30w-32, 0.001, 0.00063095734448, 37.2041897904, 0.759156575963
d-40w-27, 0.0001, 0.00199526231497, 30.0, 0.57735026919
d-40w-18, 0.0001, 0.0158489319246, 30.0, 0.57735026919
d-40w-26, 0.0001, 0.00251188643151, 30.0, 0.57735026919
d-30w-33, 0.001, 0.000501187233627, 41.207551197, 0.875666647325
d-40w-19, 0.0001, 0.0125892541179, 30.0, 0.57735026919
d-30w-34, 0.001, 0.000398107170553, 42.5177711916, 0.916901936756
d-40w-21, 0.0001, 0.00794328234724, 30.0, 0.57735026919
d-40w-17, 0.0001, 0.0199526231497, 29.9528704232, 0.576254034501
d-40w-28, 0.0001, 0.00158489319246, 30.7836644809, 0.595733281656
d-40w-10, 0.0001, 0.1, 29.9069062806, 0.575185908395
```

Convert the data into a pandas DataFrame. This allows us to sort the data set according to d' and w', which in turn will make it possible to turn the data into 4 x 31 array in which each row represents one set of experiments at a fixed value of d'.

[4]:
```
import pandas as pd
```

[5]:
```
tau = 500.0  # average interval between unit slips, years (= tau_s, or "model␣
 ↪input tau", divided by root 3)
dprime = dist_param * tau
wprime = weath_param * tau
data = {'dist_rate': dist_param, 'dprime': dprime, 'weath_rate': weath_param,␣
 ↪'wprime': wprime, 'angle': facet_angle}
df = pd.DataFrame(data)
df = df.sort_values(by=['dprime', 'wprime'])
df
```

[5]:
```
     dist_rate  dprime  weath_rate    wprime      angle
53      0.0001    0.05    0.000100  0.050000  54.744992
8       0.0001    0.05    0.000126  0.062946  53.513875
4       0.0001    0.05    0.000158  0.079245  51.625369
12      0.0001    0.05    0.000200  0.099763  50.452691
1       0.0001    0.05    0.000251  0.125594  48.738780
65      0.0001    0.05    0.000316  0.158114  46.002050
72      0.0001    0.05    0.000398  0.199054  44.111745
76      0.0001    0.05    0.000501  0.250594  41.695198
62      0.0001    0.05    0.000631  0.315479  39.287573
2       0.0001    0.05    0.000794  0.397164  35.563626
11      0.0001    0.05    0.001000  0.500000  34.147478
110     0.0001    0.05    0.001259  0.629463  33.139637
122     0.0001    0.05    0.001585  0.792447  30.783664
114     0.0001    0.05    0.001995  0.997631  30.000000
116     0.0001    0.05    0.002512  1.255943  30.000000
```

```
56      0.0001    0.05    0.003162   1.581139  30.674184
50      0.0001    0.05    0.003981   1.990536  30.000000
51      0.0001    0.05    0.005012   2.505936  29.862111
55      0.0001    0.05    0.006310   3.154787  30.038637
120     0.0001    0.05    0.007943   3.971641  30.000000
111     0.0001    0.05    0.010000   5.000000  29.967377
118     0.0001    0.05    0.012589   6.294627  30.000000
115     0.0001    0.05    0.015849   7.924466  30.000000
121     0.0001    0.05    0.019953   9.976312  29.952870
109     0.0001    0.05    0.025119  12.559432  29.952870
46      0.0001    0.05    0.031623  15.811388  29.952870
60      0.0001    0.05    0.039811  19.905359  29.906906
58      0.0001    0.05    0.050119  25.059362  30.000000
47      0.0001    0.05    0.063096  31.547867  29.862111
108     0.0001    0.05    0.079433  39.716412  30.000000
..         ...     ...         ...        ...        ...
88      0.1000   50.00    0.000126   0.062946  50.551075
83      0.1000   50.00    0.000158   0.079245  46.769088
93      0.1000   50.00    0.000200   0.099763  43.832310
78      0.1000   50.00    0.000251   0.125594  40.508333
19      0.1000   50.00    0.000316   0.158114  37.018811
25      0.1000   50.00    0.000398   0.199054  33.989446
27      0.1000   50.00    0.000501   0.250594  28.038573
17      0.1000   50.00    0.000631   0.315479  24.093211
81      0.1000   50.00    0.000794   0.397164  19.815304
91      0.1000   50.00    0.001000   0.500000  16.743712
31      0.1000   50.00    0.001259   0.629463  11.844310
44      0.1000   50.00    0.001585   0.792447   9.640258
34      0.1000   50.00    0.001995   0.997631   6.515246
37      0.1000   50.00    0.002512   1.255943   6.322811
104     0.1000   50.00    0.003162   1.581139   4.860082
97      0.1000   50.00    0.003981   1.990536   4.115794
99      0.1000   50.00    0.005012   2.505936   4.370462
102     0.1000   50.00    0.006310   3.154787   3.953939
40      0.1000   50.00    0.007943   3.971641   3.700613
32      0.1000   50.00    0.010000   5.000000   3.755139
38      0.1000   50.00    0.012589   6.294627   4.239145
35      0.1000   50.00    0.015849   7.924466   3.767967
43      0.1000   50.00    0.019953   9.976312   3.745517
30      0.1000   50.00    0.025119  12.559432   3.942718
95      0.1000   50.00    0.031623  15.811388   3.856154
107     0.1000   50.00    0.039811  19.905359   3.755139
106     0.1000   50.00    0.050119  25.059362   3.654101
96      0.1000   50.00    0.063096  31.547867   3.827295
28      0.1000   50.00    0.079433  39.716412   3.750328
45      0.1000   50.00    0.100000  50.000000   3.968364
```

```
[124 rows x 5 columns]
```

Reshape into 4 x 31 arrays in which each row is a unique value of d'. Print the `dprime` values to make sure it worked.

[6]:
```
facet_angle = df['angle'].values.reshape((4, 31))
wprime = df['wprime'].values.reshape((4, 31))
dprime = df['dprime'].values.reshape((4, 31))
dprime
```

[6]:
```
array([[ 0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,
         0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,
         0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,  0.05,
         0.05,  0.05,  0.05,  0.05],
       [ 0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,
         0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,
         0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,  0.5 ,
         0.5 ,  0.5 ,  0.5 ,  0.5 ],
       [ 5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,
         5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,
         5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,  5.  ,
         5.  ,  5.  ,  5.  ,  5.  ],
       [50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  ,
        50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  ,
        50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  , 50.  ,
        50.  , 50.  , 50.  , 50.  ]])
```

Read facet angle from dissolution-only runs. This is another set of model runs that we want to include on the figure.

[7]:
```
# Name of file
filename = 'data_analysis_dissolution20190604.csv'

# Count number of lines in file
num_lines = len(open(filename).readlines(  ))

# Create data arrays
diss_param = np.zeros(num_lines - 2)   # we'll skip 2 header lines
diss_facet_angle = np.zeros(num_lines - 2)

# Read data
with open(filename, 'r') as csvfile:
    myreader = csv.reader(csvfile)
    i = 0
    for row in myreader:
        print(', '.join(row))
        if i >= 2:
            diss_param[i-2] = row[1]
            diss_facet_angle[i-2] = row[2]
        i += 1
```

```
Run name, Dissolution rate parameter (1/y), Slope angle (deg), Slope gradient
(m/m), Intercept, Average vertical erosion rate (m/y), Fractional soil cover
dissolve_dr36, 36.0e-5, 14.9569455272, 0.267143960973, 2.96105160662,
0.000659129383784, 0.0
dissolve_dr40, 40.0e-5, 11.9060025534, 0.210842471723, 2.62025316456,
0.000689201405244, 0.0
dissolve_dr48, 48.0e-5, 5.32663961368, 0.093236165672, 3.02336903603,
0.0007411955448, 0.0
dissolve_dr4, 4.0e-5, 56.3002742582, 1.49945229357, 1.34858812074,
9.83170797729e-05, 0.0
dissolve_dr24, 24.0e-5, 35.2759196107, 0.707408676957, 2.43135345667,
0.000453180781893, 0.0
dissolve_dr12, 12.0e-5, 47.1173946606, 1.07678354976, 2.58519961052,
0.000293422087718, 0.0
dissolve_dr32, 32.0e-5, 23.4877741013, 0.434558673889, 2.25024342746,
0.000584352261214, 0.0
dissolve_dr28, 28.0e-5, 30.1471422173, 0.580779516163, 1.80525803311,
0.000531503000746, 0.0
dissolve_dr44, 44.0e-5, 6.01398172095, 0.105350964407, 3.57838364167,
0.000726271369592, 0.0
dissolve_dr8, 8.0e-5, 51.2908623697, 1.24779616114, 2.21811100292,
0.00020483469493, 0.0
dissolve_dr20, 20.0e-5, 38.7117594638, 0.801488100081, 1.1723466407,
0.000431691113396, 0.0
dissolve_dr16, 16.0e-5, 42.8380938761, 0.927245894841, 2.94255111977,
0.000361581616457, 0.0
```

Plot the data. First, we'll define four plotting symbols for each of the four values of d', defined as $d\tau$. Then we'll plot, for each of the four, the facet angle as a function of weathering parameter (w', defined as $w\tau$).

Next, we add the dissolution runs (using the dissolution parameter, which corresponds to a weathering parameter in a world where $d' \rightarrow \infty$).

Then add the curve that represents the analytical solution for the dissolution-only case:

$\theta = \alpha - \sin^{-1}\left(\frac{2a\delta s}{V}\right) = \alpha - \sin^{-1}(2as\tau)$.

Because in the dissolution case $w' = s\tau$, this becomes:

$\theta = \alpha - \sin^{-1}(2aw')$.

(Note that here τ is as defined in the paper, the average time between slip events of one cell, rather than the time between numerical slips of $\sqrt{3}$ cells used in the numerical model, which in the paper is called τ_s).

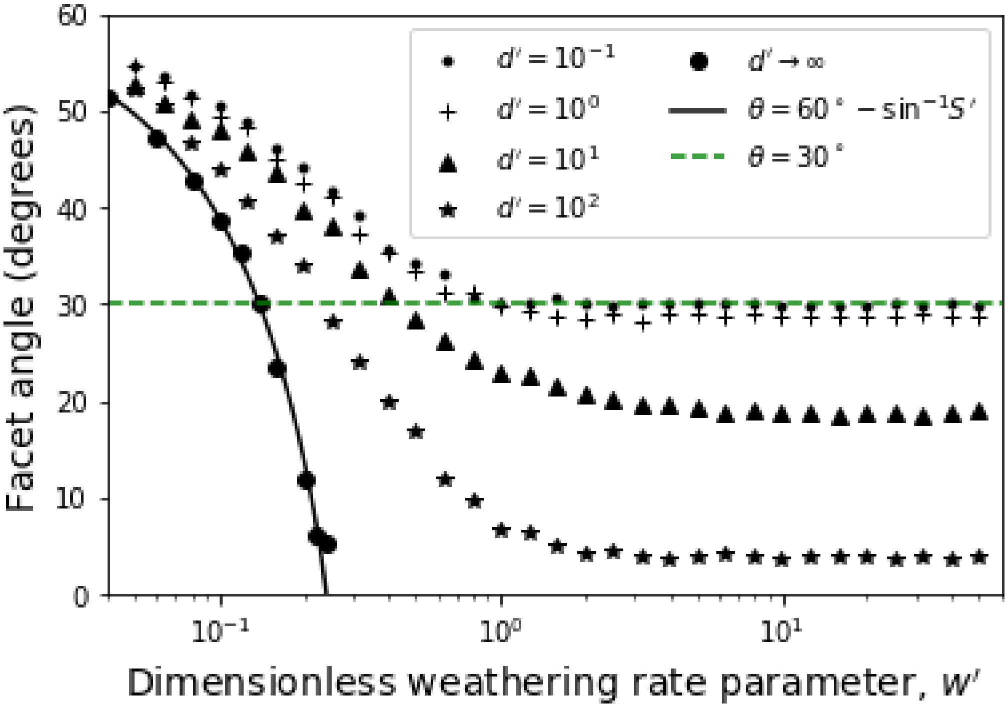
Also plot the model's effective 30° angle of repose for reference.

Finally, add axis labels, legend, etc., and save the figure to a file.

[8]:
```
psyms = ['k.', 'k+', 'k^', 'k*']
```

[9]:
```
# Plot the weathering + disturbance runs

for d in range(4):
    plt.semilogx(wprime[d,:], facet_angle[d,:], psyms[d])
```

```
# Plot the dissolution runs
diss_prime = diss_param * tau
plt.semilogx(diss_prime, diss_facet_angle, 'ko')

# Analytical solution for dissolution only
a = 1.8       # roughness factor, dimensionless
delta = 0.5   # cell width, m
V = 0.001     # slip rate, m/y
wprime_for_diss = 10.0 ** np.arange(-2, -0.6, 0.01)
diss_ang_pred = np.degrees(np.radians(60.0) - np.arcsin(2 * a * wprime_for_diss))
plt.plot(wprime_for_diss, diss_ang_pred, 'k')
plt.ylim([0, 60])

# Angle of repose
ww = np.array([0.04, 60.0])
angrep = np.array([30.0, 30.0])
plt.plot(ww, angrep, 'g--')

# Axis limits
plt.xlim([0.04, 60])

# Labels and legend
plt.xlabel(r"Dimensionless weathering rate parameter, $w'$", fontsize=14)
plt.ylabel('Facet angle (degrees)', fontsize=14)
plt.legend([r"$d'= 10^{-1}$", r"$d'= 10^0$", r"$d'= 10^1$", r"$d'= 10^2$", r"$d'␣
 ↪\rightarrow \infty$", r"$\theta = 60^\circ - \sin^{-1}S'$", r"$\theta =␣
 ↪30^\circ$"], ncol=2, fontsize=10)

plt.savefig('facet_angle_vs_wprime.pdf')
```


60
50
40
30
20
10
0
Facet angle (degrees)
10^{-1}
10^{0}
10^{1}
Dimensionless weathering rate parameter, w'
$d' = 10^{-1}$
$d' = 10^{0}$
$d' = 10^{1}$
$d' = 10^{2}$
$d' \to \infty$
$\theta = 60^\circ - \sin^{-1} S'$
$\theta = 30^\circ$

Fin.

S4.4. Fractional regolith cover as a function of w' and d'

The code below reads and plots fractional regolith cover from a series of sensitivity analysis runs.

[1]:

```
import csv
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
```

[2]:

```
filename = 'regolith_analysis20180910.csv'
```

[3]:

```
# Count number of lines in file
num_lines = len(open(filename).readlines( ))

# Create data arrays
dist_param = np.zeros(num_lines - 1)  # skip 1 header line
weath_param = np.zeros(num_lines - 1)
reg_cover_proportion = np.zeros(num_lines - 1)
reg_thickness = np.zeros(num_lines - 1)

# Read data
with open(filename, 'r') as csvfile:
    myreader = csv.reader(csvfile)
    i = 0
    for row in myreader:
        print(','.join(row) + '\n')
        if i >= 1:
            dist_param[i-1] = row[1]
            weath_param[i-1] = row[2]
            reg_cover_proportion[i-1] = row[3]
            reg_thickness[i-1] = row[9]
        i += 1
```

```
Run name,Disturbance rate parameter,Weathering rate parameter,Proportion
regolith,Proportion rock,Number regolith-air pairs,Number rock-air pairs,Number
surface pairs,Total regolith cells,Regolith thickness

d-30w-23,0.001,0.00501187233627,0.643312101911,0.356687898089,101,56,157,51,0.64
5569620253

d-40w-36,0.0001,0.000251188643151,0.207171314741,0.792828685259,52,199,251,30,0.
379746835443

d-40w-31,0.0001,0.000794328234724,0.40782122905,0.59217877095,73,106,179,38,0.48
1012658228

d-30w-24,0.001,0.00398107170553,0.906832298137,0.0931677018634,146,15,161,73,0.9
24050632911
```

```
d-40w-38,0.0001,0.000158489319246,0.211320754717,0.788679245283,56,209,265,31,0.
392405063291

d-30w-12,0.001,0.063095734448,0.987261146497,0.0127388535032,155,2,157,78,0.9873
41772152

d-30w-15,0.001,0.0316227766017,0.936305732484,0.0636942675159,147,10,157,74,0.93
6708860759

d-30w-14,0.001,0.0398107170553,0.987261146497,0.0127388535032,155,2,157,78,0.987
341772152

d-40w-39,0.0001,0.000125892541179,0.154098360656,0.845901639344,47,258,305,27,0.
341772151899

d-30w-13,0.001,0.0501187233627,0.987261146497,0.0127388535032,155,2,157,78,0.987
341772152

d-30w-25,0.001,0.00316227766017,0.867924528302,0.132075471698,138,21,159,74,0.93
6708860759

d-40w-30,0.0001,0.001,0.611428571429,0.388571428571,107,68,175,56,0.708860759494

d-40w-37,0.0001,0.000199526231497,0.267657992565,0.732342007435,72,197,269,40,0.
506329113924

d-30w-22,0.001,0.0063095734448,0.905660377358,0.0943396226415,144,15,159,71,0.89
8734177215

d-30w-40,0.001,0.0001,0.16717325228,0.83282674772,55,274,329,29,0.367088607595

d-20w-40,0.01,0.0001,0.121813031161,0.878186968839,43,310,353,20,0.253164556962

d-20w-22,0.01,0.0063095734448,0.834394904459,0.165605095541,131,26,157,70,0.8860
75949367

d-10w-32,0.1,0.00063095734448,0.257918552036,0.742081447964,57,164,221,29,0.3670
88607595

d-20w-25,0.01,0.00316227766017,0.754716981132,0.245283018868,120,39,159,62,0.784
810126582

d-10w-35,0.1,0.000316227766017,0.135458167331,0.864541832669,34,217,251,18,0.227
848101266

d-20w-13,0.01,0.0501187233627,0.980891719745,0.0191082802548,154,3,157,89,1.1265
8227848
```

```
d-20w-14,0.01,0.0398107170553,0.936305732484,0.0636942675159,147,10,157,80,1.01265822785

d-20w-15,0.01,0.0316227766017,0.980891719745,0.0191082802548,154,3,157,91,1.15189873418

d-20w-12,0.01,0.063095734448,0.987261146497,0.0127388535032,155,2,157,93,1.17721518987

d-20w-24,0.01,0.00398107170553,0.84076433121,0.15923566879,132,25,157,73,0.924050632911

d-10w-34,0.1,0.000398107170553,0.126637554585,0.873362445415,29,200,229,16,0.20253164557

d-20w-23,0.01,0.00501187233627,0.802547770701,0.197452229299,126,31,157,67,0.848101265823

d-10w-33,0.1,0.000501187233627,0.153526970954,0.846473029046,37,204,241,19,0.240506329114

d-10w-11,0.1,0.0794328234724,0.980891719745,0.0191082802548,154,3,157,98,1.24050632911

d-20w-39,0.01,0.000125892541179,0.129909365559,0.870090634441,43,288,331,24,0.303797468354

d-10w-16,0.1,0.0251188643151,0.917197452229,0.0828025477707,144,13,157,85,1.07594936709

d-10w-29,0.1,0.00125892541179,0.445714285714,0.554285714286,78,97,175,43,0.544303797468

d-10w-20,0.1,0.01,0.898089171975,0.101910828025,141,16,157,76,0.962025316456

d-20w-30,0.01,0.001,0.486033519553,0.513966480447,87,92,179,45,0.569620253165

d-10w-27,0.1,0.00199526231497,0.635220125786,0.364779874214,101,58,159,49,0.620253164557

d-10w-18,0.1,0.0158489319246,0.904458598726,0.0955414012739,142,15,157,81,1.0253164557

d-20w-37,0.01,0.000199526231497,0.158878504673,0.841121495327,51,270,321,28,0.354430379747

d-10w-26,0.1,0.00251188643151,0.668789808917,0.331210191083,105,52,157,52,0.658227848101
```

```
d-10w-19,0.1,0.0125892541179,0.891719745223,0.108280254777,140,17,157,74,0.93670
8860759

d-20w-36,0.01,0.000251188643151,0.191335740072,0.808664259928,53,224,277,29,0.36
7088607595

d-10w-21,0.1,0.00794328234724,0.880503144654,0.119496855346,140,19,159,73,0.9240
50632911

d-20w-31,0.01,0.000794328234724,0.364102564103,0.635897435897,71,124,195,34,0.43
0379746835

d-20w-38,0.01,0.000158489319246,0.123028391167,0.876971608833,39,278,317,18,0.22
7848101266

d-10w-17,0.1,0.0199526231497,0.898089171975,0.101910828025,141,16,157,84,1.06329
113924

d-10w-28,0.1,0.00158489319246,0.479289940828,0.520710059172,81,88,169,40,0.50632
9113924

d-10w-10,0.1,0.1,0.96178343949,0.0382165605096,151,6,157,96,1.21518987342

d-40w-15,0.0001,0.0316227766017,0.968152866242,0.031847133758,152,5,157,77,0.974
683544304

d-40w-12,0.0001,0.063095734448,0.974522292994,0.0254777070064,153,4,157,77,0.974
683544304

d-30w-38,0.001,0.000158489319246,0.148288973384,0.851711026616,39,224,263,26,0.3
29113924051

d-30w-31,0.001,0.000794328234724,0.438502673797,0.561497326203,82,105,187,46,0.5
82278481013

d-40w-24,0.0001,0.00398107170553,0.0764331210191,0.923566878981,12,145,157,7,0.0
886075949367

d-40w-23,0.0001,0.00501187233627,0.968553459119,0.0314465408805,154,5,159,78,0.9
87341772152

d-30w-36,0.001,0.000251188643151,0.22433460076,0.77566539924,59,204,263,31,0.392
405063291

d-40w-40,0.0001,0.0001,0.175895765472,0.824104234528,54,253,307,28,0.35443037974
7
```

```
d-30w-37,0.001,0.000199526231497,0.256603773585,0.743396226415,68,197,265,37,0.4
6835443038

d-40w-22,0.0001,0.0063095734448,0.446540880503,0.553459119497,71,88,159,36,0.455
696202532

d-40w-25,0.0001,0.00316227766017,0.726708074534,0.273291925466,117,44,161,60,0.7
59493670886

d-30w-30,0.001,0.001,0.456140350877,0.543859649123,78,93,171,41,0.518987341772

d-40w-13,0.0001,0.0501187233627,0.993630573248,0.00636942675159,156,1,157,79,1.0

d-30w-39,0.001,0.000125892541179,0.186119873817,0.813880126183,59,258,317,32,0.4
05063291139

d-40w-14,0.0001,0.0398107170553,0.993630573248,0.00636942675159,156,1,157,79,1.0

d-30w-18,0.001,0.0158489319246,0.867924528302,0.132075471698,138,21,159,69,0.873
417721519

d-40w-32,0.0001,0.00063095734448,0.406091370558,0.593908629442,80,117,197,43,0.5
44303797468

d-30w-27,0.001,0.00199526231497,0.769696969697,0.230303030303,127,38,165,67,0.84
8101265823

d-30w-20,0.001,0.01,0.732484076433,0.267515923567,115,42,157,59,0.746835443038

d-40w-35,0.0001,0.000316227766017,0.297872340426,0.702127659574,70,165,235,37,0.
46835443038

d-30w-29,0.001,0.00125892541179,0.538922155689,0.461077844311,90,77,167,47,0.594
936708861

d-30w-16,0.001,0.0251188643151,0.974522292994,0.0254777070064,153,4,157,78,0.987
341772152

d-30w-11,0.001,0.0794328234724,0.993630573248,0.00636942675159,156,1,157,79,1.0

d-30w-10,0.001,0.1,0.993630573248,0.00636942675159,156,1,157,86,1.08860759494

d-30w-28,0.001,0.00158489319246,0.754491017964,0.245508982036,126,41,167,63,0.79
746835443

d-30w-17,0.001,0.0199526231497,0.898089171975,0.101910828025,141,16,157,73,0.924
050632911
```

```
d-40w-34,0.0001,0.000398107170553,0.280542986425,0.719457013575,62,159,221,32,0.
405063291139

d-30w-21,0.001,0.00794328234724,0.826086956522,0.173913043478,133,28,161,65,0.82
2784810127

d-30w-19,0.001,0.0125892541179,0.891719745223,0.108280254777,140,17,157,71,0.898
734177215

d-30w-26,0.001,0.00251188643151,0.779874213836,0.220125786164,124,35,159,67,0.84
8101265823

d-40w-33,0.0001,0.000501187233627,0.449275362319,0.550724637681,93,114,207,48,0.
607594936709

d-20w-26,0.01,0.00251188643151,0.662721893491,0.337278106509,112,57,169,53,0.670
886075949

d-10w-36,0.1,0.000251188643151,0.0903010033445,0.909698996656,27,272,299,11,0.13
9240506329

d-20w-19,0.01,0.0125892541179,0.917197452229,0.0828025477707,144,13,157,81,1.025
3164557

d-20w-21,0.01,0.00794328234724,0.859872611465,0.140127388535,135,22,157,74,0.936
708860759

d-10w-31,0.1,0.000794328234724,0.227513227513,0.772486772487,43,146,189,23,0.291
139240506

d-20w-17,0.01,0.0199526231497,0.955414012739,0.0445859872611,150,7,157,85,1.0759
4936709

d-10w-38,0.1,0.000158489319246,0.0571428571429,0.942857142857,22,363,385,11,0.13
9240506329

d-20w-28,0.01,0.00158489319246,0.560693641618,0.439306358382,97,76,173,51,0.6455
69620253

d-20w-10,0.01,0.1,0.987261146497,0.0127388535032,155,2,157,91,1.15189873418

d-20w-11,0.01,0.0794328234724,0.980891719745,0.0191082802548,154,3,157,87,1.1012
6582278

d-20w-16,0.01,0.0251188643151,0.96178343949,0.0382165605096,151,6,157,88,1.11392
405063

d-10w-39,0.1,0.000125892541179,0.0481586402266,0.951841359773,17,336,353,9,0.113
```

```
924050633

d-20w-29,0.01,0.00125892541179,0.666666666667,0.333333333333,110,55,165,63,0.797
46835443

d-20w-20,0.01,0.01,0.905660377358,0.0943396226415,144,15,159,78,0.987341772152

d-10w-30,0.1,0.001,0.307262569832,0.692737430168,55,124,179,27,0.341772151899

d-20w-27,0.01,0.00199526231497,0.666666666667,0.333333333333,110,55,165,55,0.696
202531646

d-10w-37,0.1,0.000199526231497,0.094395280236,0.905604719764,32,307,339,14,0.177
215189873

d-20w-18,0.01,0.0158489319246,0.87898089172,0.12101910828,138,19,157,77,0.974683
544304

d-10w-15,0.1,0.0316227766017,0.949044585987,0.0509554140127,149,8,157,94,1.18987
341772

d-10w-12,0.1,0.063095734448,0.949044585987,0.0509554140127,149,8,157,93,1.177215
18987

d-10w-24,0.1,0.00398107170553,0.779874213836,0.220125786164,124,35,159,63,0.7974
6835443

d-20w-34,0.01,0.000398107170553,0.236514522822,0.763485477178,57,184,241,30,0.37
9746835443

d-10w-23,0.1,0.00501187233627,0.867924528302,0.132075471698,138,21,159,68,0.8607
59493671

d-20w-33,0.01,0.000501187233627,0.2466367713,0.7533632287,55,168,223,29,0.367088
607595

d-10w-40,0.1,0.0001,0.0356164383562,0.964383561644,13,352,365,6,0.0759493670886

d-10w-22,0.1,0.0063095734448,0.796178343949,0.203821656051,125,32,157,67,0.84810
1265823

d-20w-32,0.01,0.00063095734448,0.284518828452,0.715481171548,68,171,239,34,0.430
379746835

d-10w-25,0.1,0.00316227766017,0.723270440252,0.276729559748,115,44,159,56,0.7088
60759494

d-20w-35,0.01,0.000316227766017,0.32319391635,0.67680608365,85,178,263,42,0.5316
```

```
4556962

d-10w-13,0.1,0.0501187233627,0.974522292994,0.0254777070064,153,4,157,97,1.22784
810127

d-10w-14,0.1,0.0398107170553,0.917197452229,0.0828025477707,144,13,157,89,1.1265
8227848

d-40w-11,0.0001,0.0794328234724,0.993630573248,0.00636942675159,156,1,157,79,1.0

d-40w-16,0.0001,0.0251188643151,0.929936305732,0.0700636942675,146,11,157,74,0.9
36708860759

d-40w-29,0.0001,0.00125892541179,0.449101796407,0.550898203593,75,92,167,39,0.49
3670886076

d-40w-20,0.0001,0.01,0.798742138365,0.201257861635,127,32,159,65,0.822784810127

d-30w-35,0.001,0.000316227766017,0.289361702128,0.710638297872,68,167,235,35,0.4
43037974684

d-30w-32,0.001,0.00063095734448,0.31455399061,0.68544600939,67,146,213,33,0.4177
21518987

d-40w-27,0.0001,0.00199526231497,0.704402515723,0.295597484277,112,47,159,58,0.7
3417721519

d-40w-18,0.0001,0.0158489319246,0.929936305732,0.0700636942675,146,11,157,74,0.9
36708860759

d-40w-26,0.0001,0.00251188643151,0.863354037267,0.136645962733,139,22,161,71,0.8
98734177215

d-30w-33,0.001,0.000501187233627,0.323943661972,0.676056338028,69,144,213,36,0.4
55696202532

d-40w-19,0.0001,0.0125892541179,0.891719745223,0.108280254777,140,17,157,71,0.89
8734177215

d-30w-34,0.001,0.000398107170553,0.351598173516,0.648401826484,77,142,219,43,0.5
44303797468

d-40w-21,0.0001,0.00794328234724,0.687898089172,0.312101910828,108,49,157,55,0.6
96202531646

d-40w-17,0.0001,0.0199526231497,0.974522292994,0.0254777070064,153,4,157,77,0.97
4683544304
```

```
d-40w-28,0.0001,0.00158489319246,0.534161490683,0.465838509317,86,75,161,44,0.55
6962025316

d-40w-10,0.0001,0.1,0.993630573248,0.00636942675159,156,1,157,79,1.0
```

Create a Pandas DataFrame, add d' and w', and sort by d':

[4]:

```
tau = 500.0  # average interval between one-cell slip events (corresponds to␣
 ↪numerical model interval of 866 yr)
dprime = dist_param * tau
wprime = weath_param * tau
data = {'d': dist_param, 'dprime': dprime, 'w': weath_param, 'wprime': wprime,␣
 ↪'cover': reg_cover_proportion}
df = pd.DataFrame(data)
df = df.sort_values(by=['dprime', 'wprime'])
df
```

[4]:

```
          d  dprime         w     wprime     cover
53   0.0001    0.05  0.000100   0.050000  0.175896
8    0.0001    0.05  0.000126   0.062946  0.154098
4    0.0001    0.05  0.000158   0.079245  0.211321
12   0.0001    0.05  0.000200   0.099763  0.267658
1    0.0001    0.05  0.000251   0.125594  0.207171
65   0.0001    0.05  0.000316   0.158114  0.297872
72   0.0001    0.05  0.000398   0.199054  0.280543
76   0.0001    0.05  0.000501   0.250594  0.449275
62   0.0001    0.05  0.000631   0.315479  0.406091
2    0.0001    0.05  0.000794   0.397164  0.407821
11   0.0001    0.05  0.001000   0.500000  0.611429
110  0.0001    0.05  0.001259   0.629463  0.449102
122  0.0001    0.05  0.001585   0.792447  0.534161
114  0.0001    0.05  0.001995   0.997631  0.704403
116  0.0001    0.05  0.002512   1.255943  0.863354
56   0.0001    0.05  0.003162   1.581139  0.726708
50   0.0001    0.05  0.003981   1.990536  0.076433
51   0.0001    0.05  0.005012   2.505936  0.968553
55   0.0001    0.05  0.006310   3.154787  0.446541
120  0.0001    0.05  0.007943   3.971641  0.687898
111  0.0001    0.05  0.010000   5.000000  0.798742
118  0.0001    0.05  0.012589   6.294627  0.891720
115  0.0001    0.05  0.015849   7.924466  0.929936
121  0.0001    0.05  0.019953   9.976312  0.974522
109  0.0001    0.05  0.025119  12.559432  0.929936
46   0.0001    0.05  0.031623  15.811388  0.968153
60   0.0001    0.05  0.039811  19.905359  0.993631
58   0.0001    0.05  0.050119  25.059362  0.993631
47   0.0001    0.05  0.063096  31.547867  0.974522
108  0.0001    0.05  0.079433  39.716412  0.993631
```

```
..      ...    ...       ...        ...       ...
88   0.1000  50.00  0.000126   0.062946  0.048159
83   0.1000  50.00  0.000158   0.079245  0.057143
93   0.1000  50.00  0.000200   0.099763  0.094395
78   0.1000  50.00  0.000251   0.125594  0.090301
19   0.1000  50.00  0.000316   0.158114  0.135458
25   0.1000  50.00  0.000398   0.199054  0.126638
27   0.1000  50.00  0.000501   0.250594  0.153527
17   0.1000  50.00  0.000631   0.315479  0.257919
81   0.1000  50.00  0.000794   0.397164  0.227513
91   0.1000  50.00  0.001000   0.500000  0.307263
31   0.1000  50.00  0.001259   0.629463  0.445714
44   0.1000  50.00  0.001585   0.792447  0.479290
34   0.1000  50.00  0.001995   0.997631  0.635220
37   0.1000  50.00  0.002512   1.255943  0.668790
104  0.1000  50.00  0.003162   1.581139  0.723270
97   0.1000  50.00  0.003981   1.990536  0.779874
99   0.1000  50.00  0.005012   2.505936  0.867925
102  0.1000  50.00  0.006310   3.154787  0.796178
40   0.1000  50.00  0.007943   3.971641  0.880503
32   0.1000  50.00  0.010000   5.000000  0.898089
38   0.1000  50.00  0.012589   6.294627  0.891720
35   0.1000  50.00  0.015849   7.924466  0.904459
43   0.1000  50.00  0.019953   9.976312  0.898089
30   0.1000  50.00  0.025119  12.559432  0.917197
95   0.1000  50.00  0.031623  15.811388  0.949045
107  0.1000  50.00  0.039811  19.905359  0.917197
106  0.1000  50.00  0.050119  25.059362  0.974522
96   0.1000  50.00  0.063096  31.547867  0.949045
28   0.1000  50.00  0.079433  39.716412  0.980892
45   0.1000  50.00  0.100000  50.000000  0.961783

[124 rows x 5 columns]
```

Reshape into a 4 x 31 array, in which each row has a constant d':

[5]:

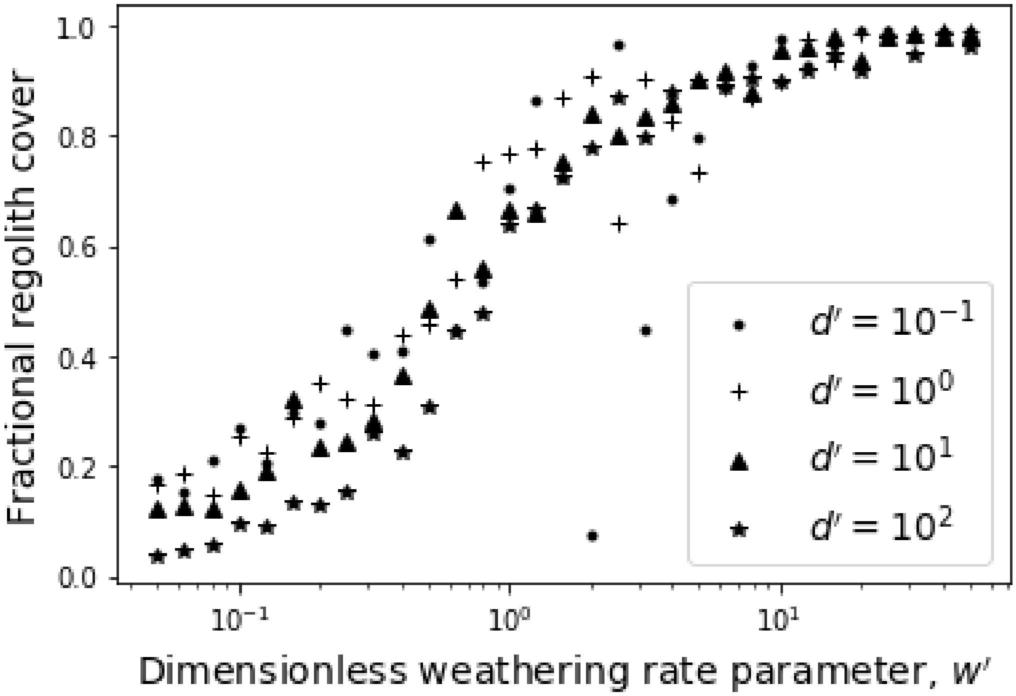
```python
reg_cover_proportion = df['cover'].values.reshape((4, 31))
wprime = df['wprime'].values.reshape((4, 31))
dprime = df['dprime'].values.reshape((4, 31))
```

Plot the cover fraction versus w', with a different symbol for each value of d':

[7]:

```python
psyms = ['k.', 'k+', 'k^', 'k*']

# Plot the weathering + disturbance runs
for d in range(4):
    plt.semilogx(wprime[d,:], reg_cover_proportion[d,:], psyms[d])

# Labels and legend
```

```
plt.xlabel(r"Dimensionless weathering rate parameter, $w'$", fontsize=14)
plt.ylabel('Fractional regolith cover', fontsize=14)
plt.legend([r"$d'= 10^{-1}$", r"$d'= 10^0$", r"$d'= 10^1$", r"$d'= 10^2$"],␣
 ↪fontsize=14)

plt.savefig('reg_cover_vs_wprime.pdf')
```



S4.5 Relationships among erosion rate, slip rate, and regolith cover

The code below plots the relationship between erosion rate and facet angle, and how that relation varies with weathering efficiency W and regolith cover fraction. Also plot the relationship between slip rate and facet angle.

[1]:
```
import csv
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Read in data from 121 model runs that have already been compiled into a .csv file. Read into arrays for the slip-interval parameter, the weathering-rate parameter, and the resulting facet slope angle.

[2]:
```
filename = 'slope_erorate_vw_series20190517.csv'
```

[3]:
```
# Count number of lines in file
num_lines = len(open(filename).readlines(  ))
```

```python
print('The file has ' + str(num_lines) + ' lines.')

# Create data arrays
slip_interval = np.zeros(num_lines - 1)  # skip 1 header line
weath_param = np.zeros(num_lines - 1)
facet_angle = np.zeros(num_lines - 1)
ero_rate = np.zeros(num_lines - 1)
frac_soil = np.zeros(num_lines - 1)

# Read data
with open(filename, 'r') as csvfile:
    myreader = csv.reader(csvfile)
    i = 0
    for row in myreader:
        print(', '.join(row))
        if i >= 1:
            slip_interval[i-1] = row[1]
            weath_param[i-1] = row[2]
            facet_angle[i-1] = row[3]
            ero_rate[i-1] = row[6]
            frac_soil[i-1] = row[7]
        i += 1
```

```
The file has 257 lines.
Run name, Slip interval, Weathering rate parameter, Slope angle, Slope gradient,
Intercept, Average erosion rateFractional soil cover
tau40w-48, 10000.0, 1.5848931924611107e-05, 50.8257947296, 1.22724878788,
1.33495618306, 1.96257060177e-05, 0.26755852842809363
tau40w-46, 10000.0, 2.511886431509577e-05, 46.7343097218, 1.06244817307,
0.0691333982473, 2.80707889797e-05, 0.23293172690763053
tau26w-30, 398.1071705534973, 0.001, 42.8484836728, 0.927583197822,
0.88023369036, 0.000812968192091, 0.3813953488372093
tau32w-30, 1584.893192461114, 0.001, 30.8418731839, 0.597110602161,
-0.203505355404, 0.000312752784842, 0.7904191616766467
tau28w-44, 630.9573444801937, 3.9810717055349695e-05, 58.5858411295,
1.63735299565, 0.508276533593, 5.599422234e-05, 0.13314447592067988
tau42w-22, 15848.93192461114, 0.00630957344480193, 26.541646699, 0.499489497736,
1.98831548199, 3.09205943141e-05, 0.950920245398773
tau28w-28, 630.9573444801937, 0.0015848931924611126, 32.2788021367,
0.631656049135, 1.97955209348, 0.000678371832473, 0.7225433526011561
tau24w-38, 251.18864315095823, 0.00015848931924611126, 58.023493086,
1.60179563973, -0.272638753651, 0.000252433306338, 0.22485207100591717
tau30w-38, 1000.0, 0.00015848931924611126, 51.5277132438, 1.25842120504,
0.305744888023, 0.000198777668051, 0.18466898954703834
tau40w-24, 10000.0, 0.003981071705534969, 27.7681486865, 0.526529953382,
1.79844206426, 4.82579792443e-05, 0.950920245398773
tau30w-36, 1000.0, 0.00025118864315095795, 46.2025866587, 1.04288460017,
```

```
1.84810126582, 0.000258249125459, 0.27848101265822783
tau24w-36, 251.18864315095823, 0.00025118864315095795, 56.6549078948,
1.51974668959, -0.26582278481, 0.000404679103126, 0.1620795107033639
tau42w-40, 15848.93192461114, 0.0001, 29.1669346777, 0.558123999271,
1.24829600779, 3.01947258603e-05, 0.7543859649122807
tau28w-26, 630.9573444801937, 0.0025118864315095794, 30.092919389,
0.579514629985, 0.950340798442, 0.00075202716654, 0.593939393939394
tau40w-40, 10000.0, 0.0001, 33.5621596181, 0.663446855097, 0.182083739046,
4.53473295447e-05, 0.6408839779005525
tau32w-36, 1584.893192461114, 0.00025118864315095795, 42.9978821575,
0.932445982465, -0.693281402142, 0.000225999205235, 0.30985915492957744
tau26w-36, 398.1071705534973, 0.00025118864315095795, 54.1493553243,
1.38395413115, 0.754625121714, 0.000355597754992, 0.16393442622950818
tau28w-42, 630.9573444801937, 6.309573444801929e-05, 57.829133356,
1.58976516674, 0.270691333982, 9.71095094829e-05, 0.14244186046511628
tau42w-24, 15848.93192461114, 0.003981071705534969, 26.5738608616,
0.50019221228, 1.95131450828, 3.09433512885e-05, 0.9570552147239264
tau26w-38, 398.1071705534973, 0.00015848931924611126, 56.012551273,
1.48326175048, 1.27653359299, 0.000218067590743, 0.167192429022082
tau32w-38, 1584.893192461114, 0.00015848931924611126, 47.1911254005,
1.07956629936, 0.881207400195, 0.000163519110143, 0.23715415019762845
tau40w-22, 10000.0, 0.00630957344480193, 27.5685570472, 0.522088797465,
1.83836416748, 4.83704275959e-05, 0.9570552147239264
tau24w-30, 251.18864315095823, 0.001, 47.2208576183, 1.08069064263,
0.297955209348, 0.00113527020524, 0.30739299610894943
tau30w-30, 1000.0, 0.001, 32.5759809338, 0.638936171809, 1.7147030185,
0.000438713074031, 0.5771428571428572
tau42w-46, 15848.93192461114, 2.511886431509577e-05, 40.1291556494,
0.842948258166, 2.47030185005, 2.09136907221e-05, 0.4028436018957346
tau28w-20, 630.9573444801937, 0.01, 29.8184876189, 0.573133981927,
1.09542356378, 0.000752810493793, 0.9386503067484663
tau42w-48, 15848.93192461114, 1.5848931924611107e-05, 45.82508707,
1.02922382944, 1.59980525803, 1.7203895328e-05, 0.2222222222222222
tau46w-48, 39810.71705534977, 1.5848931924611107e-05, 34.6125886775,
0.690178116343, 1.06621226874, 1.08966979666e-05, 0.36787564766839376
tau38w-20, 6309.573444801937, 0.01, 28.3837393666, 0.540331267022,
0.52288218111, 8.02766604736e-05, 0.9813664596273292
tau34w-30, 2511.886431509582, 0.001, 29.7129073174, 0.570688535314,
0.104186952288, 0.000199456980249, 0.9079754601226994
tau20w-30, 100.0, 0.001, 54.0174593335, 1.37726428869, 0.657254138267,
0.00142036212421, 0.18181818181818182
tau46w-46, 39810.71705534977, 2.511886431509577e-05, 29.8463624597,
0.573780479307, 0.592015579357, 1.22975736095e-05, 0.56
tau48w-32, 63095.73444801943, 0.000630957344480193, 20.4103046787,
0.372101405235, 2.95813047712, 8.34289967712e-06, 0.9386503067484663
tau44w-22, 25118.864315095823, 0.00630957344480193, 24.9649182165,
0.465562439561, 2.36611489776, 1.98105649207e-05, 0.9570552147239264
tau50w-22, 100000.0, 0.00630957344480193, 16.8765341046, 0.303375922851,
```

```
2.9717624148, 5.54369301377e-06, 0.9570552147239264
tau36w-38, 3981.0717055349733, 0.00015848931924611126, 36.2775998947,
0.733971286713, 0.270691333982, 0.000105838226802, 0.40512820512820513
tau22w-38, 158.48931924611142, 0.00015848931924611126, 58.518332051,
1.63302427406, 0.316455696203, 0.000260585247116, 0.14114114114114115
tau46w-24, 39810.71705534977, 0.003981071705534969, 22.8821937595,
0.422050355009, 1.58227848101, 1.35022456872e-05, 0.9815950920245399
tau38w-42, 6309.573444801937, 6.309573444801929e-05, 41.6043782866,
0.887978206133, 0.973709834469, 5.47461637743e-05, 0.34977578475336324
tau50w-40, 100000.0, 0.0001, 17.2018932397, 0.309587919418, 3.02239532619,
5.5112598777e-06, 0.9079754601226994
tau44w-40, 25118.864315095823, 0.0001, 26.8732253136, 0.506741511828,
1.99026290166, 1.94064990651e-05, 0.8545454545454545
tau22w-36, 158.48931924611142, 0.00025118864315095795, 57.3971262316,
1.5634836428, 0.902629016553, 0.000399111228389, 0.14492753623188406
tau36w-36, 3981.0717055349733, 0.00025118864315095795, 32.5336467014,
0.637896154284, 0.00389483933788, 0.000119059682964, 0.6171428571428571
tau48w-50, 63095.73444801943, 1e-05, 32.4373882569, 0.635535033417,
2.44011684518, 6.71425323156e-06, 0.36097560975609755
tau38w-26, 6309.573444801937, 0.0025118864315095794, 28.3600494352,
0.539797203969, 0.604673807205, 8.0143388661e-05, 0.9693251533742331
tau20w-36, 100.0, 0.00025118864315095795, 58.40862545, 1.6260252372,
0.444985394352, 0.000407560125432, 0.1671388101983003
tau34w-36, 2511.886431509582, 0.00025118864315095795, 37.6591365145,
0.771749220588, -0.126582278481, 0.000168060460327, 0.541871921182266
tau46w-40, 39810.71705534977, 0.0001, 23.9319544256, 0.443806397285,
1.48685491723, 1.33273038662e-05, 0.9041916167664671
tau48w-34, 63095.73444801943, 0.0003981071705534969, 20.6207960108,
0.376289583916, 2.711781889, 8.38365671107e-06, 0.950920245398773
tau50w-24, 100000.0, 0.003981071705534969, 17.0886614544, 0.307423558624,
2.94547224927, 5.53476725333e-06, 0.9386503067484663
tau44w-24, 25118.864315095823, 0.003981071705534969, 25.1288270963,
0.469047903699, 2.1694255112, 1.98790547524e-05, 0.9386503067484663
tau34w-38, 2511.886431509582, 0.00015848931924611126, 43.155129606,
0.937589852925, -0.80817916261, 0.000143364177841, 0.2556053811659193
tau20w-38, 100.0, 0.00015848931924611126, 59.1848605245, 1.67650825003,
0.151898734177, 0.000241332654018, 0.11041009463722397
tau38w-28, 6309.573444801937, 0.0015848931924611126, 28.4086647061,
0.540893438657, 0.629990262902, 8.00342754212e-05, 0.9570552147239264
tau46w-22, 39810.71705534977, 0.00630957344480193, 22.7206958752,
0.418733542362, 1.63388510224, 1.35139281459e-05, 0.9754601226993865
tau38w-44, 6309.573444801937, 3.9810717055349695e-05, 45.666661222,
1.02354589592, 1.39143135346, 4.38782387872e-05, 0.2549800796812749
tau44w-46, 25118.864315095823, 2.511886431509577e-05, 36.4707512993,
0.739171374337, 0.698149951315, 1.64308586335e-05, 0.39593908629441626
tau50w-46, 100000.0, 2.511886431509577e-05, 20.6207960108, 0.376289583916,
2.72444011685, 5.29328422872e-06, 0.7177914110429447
tau36w-30, 3981.0717055349733, 0.001, 29.1878061494, 0.558601845161,
```

```
0.370983446933, 0.000126101938839, 0.9254658385093167
tau22w-30, 158.48931924611142, 0.001, 51.5943141442, 1.26142882329,
0.986368062317, 0.00115311522267, 0.22968197879858657
tau50w-48, 100000.0, 1.5848931924611107e-05, 22.9668950373, 0.423793087078,
2.12950340798, 5.19864277017e-06, 0.6627218934911243
tau44w-48, 25118.864315095823, 1.5848931924611107e-05, 41.0513058865,
0.870860079846, 0.617332035054, 1.44630338013e-05, 0.19574468085106383
tau38w-50, 6309.573444801937, 1e-05, 56.0170801893, 1.48351472772,
0.267770204479, 1.6873718323e-05, 0.1836734693877551
tau34w-40, 2511.886431509582, 0.0001, 47.454574354, 1.08957295446,
-0.111002921129, 0.000113890527007, 0.19540229885057472
tau20w-40, 100.0, 0.0001, 59.5072118096, 1.69815185798, 0.17429406037,
0.000131279954256, 0.10031347962382445
tau46w-36, 39810.71705534977, 0.00025118864315095795, 23.2339623646,
0.429302369101, 1.58422590068, 1.34279708151e-05, 0.9263803680981595
tau48w-42, 63095.73444801943, 6.309573444801929e-05, 21.9726262513,
0.403470582471, 2.44109055501, 8.28691585297e-06, 0.8404907975460123
tau36w-24, 3981.0717055349733, 0.003981071705534969, 29.0661412814,
0.555819095567, 0.391431353457, 0.000126273088268, 0.9325153374233128
tau22w-24, 158.48931924611142, 0.003981071705534969, 38.0496989816,
0.782683458889, 0.317429406037, 0.00251455416987, 0.3544973544973545
tau46w-38, 39810.71705534977, 0.00015848931924611126, 23.7352143136,
0.439702544349, 1.35053554041, 1.34187394199e-05, 0.901840490797546
tau20w-22, 100.0, 0.00630957344480193, 37.1407920607, 0.757413843894,
0.749756572541, 0.00403372187114, 0.32085561497326204
tau34w-22, 2511.886431509582, 0.00630957344480193, 29.1005807458,
0.556606135857, 0.427458617332, 0.000199872760448, 0.9515151515151515
tau38w-32, 6309.573444801937, 0.000630957344480193, 28.5530004311,
0.54415403414, 0.580331061344, 7.99519680015e-05, 0.8711656441717791
tau50w-30, 100000.0, 0.001, 16.7451885407, 0.300874259075, 3.1723466407,
5.52681738677e-06, 0.9570552147239264
tau44w-30, 25118.864315095823, 0.001, 25.1882017149, 0.470312789877,
2.18889970789, 1.98528667381e-05, 0.9570552147239264
tau22w-46, 158.48931924611142, 2.511886431509577e-05, 59.4115649931,
1.69168688417, 0.664070107108, 3.18024723968e-05, 0.03986710963455149
tau36w-46, 3981.0717055349733, 2.511886431509577e-05, 53.8219637369,
1.36742628508, 0.732229795521, 3.81070775308e-05, 0.21580547112462006
tau48w-20, 63095.73444801943, 0.01, 20.4979619691, 0.373844137304,
2.85978578384, 8.36829689019e-06, 0.950920245398773
tau36w-48, 3981.0717055349733, 1.5848931924611107e-05, 55.684681647,
1.46510360667, 1.00681596884, 2.63225799152e-05, 0.15615615615615616
tau22w-48, 158.48931924611142, 1.5848931924611107e-05, 59.4731737029,
1.69584695427, 0.671859785784, 1.97625379783e-05, 0.013201320132013201
tau34w-48, 2511.886431509582, 1.5848931924611107e-05, 57.4377588804,
1.56592908941, 0.754625121714, 2.53774116108e-05, 0.12181303116147309
tau20w-48, 100.0, 1.5848931924611107e-05, 59.5440780489, 1.70065352175,
0.631937682571, 1.42640236418e-05, 0.013468013468013467
tau20w-46, 100.0, 2.511886431509577e-05, 59.554005715, 1.70132812771,
```

```
0.519961051607, 3.77655851524e-05, 0.03908794788273615
tau34w-46, 2511.886431509582, 2.511886431509577e-05, 55.7240532492,
1.46726796746, 1.28627069133, 4.04637413616e-05, 0.1580547112462006
tau46w-30, 39810.71705534977, 0.001, 22.8411712679, 0.421207097556,
1.5735150925, 1.34997948863e-05, 0.9627329192546584
tau48w-44, 63095.73444801943, 3.9810717055349695e-05, 23.0487630934,
0.425479601983, 2.38753651412, 8.14291681687e-06, 0.7791411042944786
tau22w-22, 158.48931924611142, 0.00630957344480193, 33.3671403646,
0.658555961872, 0.64264849075, 0.00282615993625, 0.4915254237288136
tau36w-22, 3981.0717055349733, 0.00630957344480193, 28.8602562985,
0.551124962415, 0.452775073028, 0.000126493551123, 0.9212121212121213
tau48w-28, 63095.73444801943, 0.0015848931924611126, 20.4527319881,
0.372944662688, 2.8276533593, 8.3767142582e-06, 0.9570552147239264
tau50w-38, 100000.0, 0.00015848931924611126, 17.2004235826, 0.309559810837,
2.99805258033, 5.51110059686e-06, 0.9386503067484663
tau44w-38, 25118.864315095823, 0.00015848931924611126, 25.9743222699,
0.487177938929, 2.04771178189, 1.96669478967e-05, 0.8711656441717791
tau34w-24, 2511.886431509582, 0.003981071705534969, 29.2037609576,
0.558967256724, 0.383641674781, 0.000199758788806, 0.9570552147239264
tau20w-24, 100.0, 0.003981071705534969, 41.2703870868, 0.877606139466,
1.67478091529, 0.00333186778086, 0.37209302325581395
tau38w-34, 6309.573444801937, 0.0003981071705534969, 29.0759834782,
0.556043964221, 0.408958130477, 7.96306986391e-05, 0.8404907975460123
tau44w-36, 25118.864315095823, 0.00025118864315095795, 25.953493263,
0.486728201621, 1.9746835443, 1.97311313105e-05, 0.8834355828220859
tau50w-36, 100000.0, 0.00025118864315095795, 16.7924348544, 0.301773733691,
3.28042843233, 5.49137930727e-06, 0.9202453987730062
tau36w-40, 3981.0717055349733, 0.0001, 43.0288798773, 0.933457891408,
-0.791626095424, 9.18394996639e-05, 0.3209302325581395
tau22w-40, 158.48931924611142, 0.0001, 59.1124378887, 1.67170168255,
0.33106134372, 0.000153107379112, 0.10122699386503067
tau48w-26, 63095.73444801943, 0.0025118864315095794, 20.444248399,
0.372776011197, 2.79552093476, 8.37570642814e-06, 0.9570552147239264
tau40w-36, 10000.0, 0.00025118864315095795, 28.4186315591, 0.541118307311,
1.60954235638, 4.79801414806e-05, 0.8711656441717791
tau26w-40, 398.1071705534973, 0.0001, 57.8359806873, 1.59018679547,
0.218111002921, 0.000148630604911, 0.15709969788519637
tau32w-40, 1584.893192461114, 0.0001, 50.7872054814, 1.22556227298,
0.49464459591, 0.00013152420349, 0.32727272727272727
tau28w-34, 630.9573444801937, 0.0003981071705534969, 47.1203776485,
1.07689598409, 0.783836416748, 0.000425141922846, 0.2413793103448276
tau24w-24, 251.18864315095823, 0.003981071705534969, 32.1496855028,
0.628507887979, 0.240506329114, 0.00188909187163, 0.52046783625731
tau30w-24, 1000.0, 0.003981071705534969, 29.8184876189, 0.573133981927,
1.09542356378, 0.000474991310061, 0.9386503067484663
tau40w-38, 10000.0, 0.00015848931924611126, 29.6934660243, 0.570238798006,
1.14508276534, 4.75390944415e-05, 0.47904191616766467
tau32w-22, 1584.893192461114, 0.00630957344480193, 29.3226516192,
```

```
0.561693789154, 0.327166504382, 0.000316427402013, 0.9754601226993865
tau26w-22, 398.1071705534973, 0.00630957344480193, 29.9975841865,
0.577294052026, 0.976630963973, 0.00119318211626, 0.8982035928143712
tau30w-46, 1000.0, 2.511886431509577e-05, 58.5560639955, 1.63544161209,
0.371957156767, 3.77717615139e-05, 0.1337386018237082
tau24w-46, 251.18864315095823, 2.511886431509577e-05, 59.4003013353,
1.69092795247, 0.424537487829, 4.01444034954e-05, 0.08306709265175719
tau42w-30, 15848.93192461114, 0.001, 26.8501546021, 0.506235557357,
1.8558909445, 3.0892193372e-05, 0.9570552147239264
tau24w-48, 251.18864315095823, 1.5848931924611107e-05, 59.4690180165,
1.69556586845, 0.517039922103, 2.80201865063e-05, 0.06666666666666667
tau30w-48, 1000.0, 1.5848931924611107e-05, 58.9147984844, 1.6586874092,
0.617332035054, 2.28161622257e-05, 0.13505747126436782
tau26w-48, 398.1071705534973, 1.5848931924611107e-05, 59.3973799048,
1.69073119239, 0.481986368062, 2.81298008331e-05, 0.07120743034055728
tau32w-48, 1584.893192461114, 1.5848931924611107e-05, 58.3142347468,
1.62003810929, 0.753651411879, 2.37088184034e-05, 0.12931034482758622
tau40w-30, 10000.0, 0.001, 27.847529311, 0.528300794032, 1.74975657254,
4.82631911187e-05, 0.9202453987730062
tau32w-46, 1584.893192461114, 2.511886431509577e-05, 57.8122321407,
1.58872514921, 0.268743914314, 4.07932676767e-05, 0.16310160427807488
tau26w-46, 398.1071705534973, 2.511886431509577e-05, 59.3002696104,
1.68421000143, 0.50535540409, 3.08871106411e-05, 0.06309148264984227
tau28w-32, 630.9573444801937, 0.000630957344480193, 42.1693689636,
0.905770938382, 0.990262901655, 0.000529734719929, 0.4155251141552511
tau30w-22, 1000.0, 0.00630957344480193, 29.7347697756, 0.571194489786,
1.13729308666, 0.000475163405233, 0.9570552147239264
tau24w-22, 251.18864315095823, 0.00630957344480193, 30.7337151597,
0.594552721222, -0.317429406037, 0.00198217506537, 0.6727272727272727
tau42w-38, 15848.93192461114, 0.00015848931924611126, 28.0034315454,
0.531786258169, 1.52775073028, 3.05661219561e-05, 0.806060606060606
tau26w-24, 398.1071705534973, 0.003981071705534969, 30.0, 0.57735026919, 1.0,
0.0011922929913, 0.3067484662576687
tau32w-24, 1584.893192461114, 0.003981071705534969, 29.416831885,
0.563858149949, 0.277507302824, 0.00031632435074, 0.9627329192546584
tau24w-40, 251.18864315095823, 0.0001, 58.5525574719, 1.63521674343,
0.493670886076, 0.000152060290514, 0.1174785100286533
tau30w-40, 1000.0, 0.0001, 54.1912971471, 1.38609038336, 0.857838364167,
0.000137743596766, 0.1806853582554517
tau42w-36, 15848.93192461114, 0.00025118864315095795, 27.3199498698,
0.516579515441, 1.76338851022, 3.07258591349e-05, 0.8711656441717791
tau28w-50, 630.9573444801937, 1e-05, 59.3254376275, 1.68589651633,
0.674780915287, 1.25355653954e-05, 0.031746031746031744
tau40w-20, 10000.0, 0.01, 27.5343766937, 0.521329865758, 1.77604673807,
4.84739164939e-05, 0.950920245398773
tau42w-44, 15848.93192461114, 3.9810717055349695e-05, 36.6152455222,
0.7430784672, 0.91723466407, 2.53924231288e-05, 0.4088669950738916
tau24w-32, 251.18864315095823, 0.000630957344480193, 51.3719684237,
```

```
1.25142216819, 0.573515092502, 0.000784715654309, 0.21554770318021202
tau30w-32, 1000.0, 0.000630957344480193, 37.403909567, 0.764665857986,
0.371957156767, 0.00040454796923, 0.34054054054054056
tau28w-22, 630.9573444801937, 0.00630957344480193, 29.7347697756,
0.571194489786, 1.13729308666, 0.00075308324626, 0.9386503067484663
tau42w-28, 15848.93192461114, 0.0015848931924611126, 26.5815895662,
0.50036086377, 2.07205452775, 3.0822088772e-05, 0.950920245398773
tau32w-34, 1584.893192461114, 0.0003981071705534969, 37.2848167235,
0.761377153921, -0.121713729309, 0.000270989539387, 0.41361256544502617
tau26w-34, 398.1071705534973, 0.0003981071705534969, 50.6542293347,
1.21977190514, 1.18889970789, 0.000503805612428, 0.24528301886792453
tau40w-42, 10000.0, 6.309573444801929e-05, 36.4447057921, 0.738468659793,
0.785783836417, 4.06780357354e-05, 0.5128205128205128
tau28w-40, 630.9573444801937, 0.0001, 55.8918937516, 1.47654379944,
0.939629990263, 0.00015622579891, 0.19939577039274925
tau42w-26, 15848.93192461114, 0.0025118864315095794, 26.7153879747,
0.503284156273, 1.83154819864, 3.09961227137e-05, 0.9447852760736196
tau30w-50, 1000.0, 1e-05, 59.2258063956, 1.67923478246, 0.677702044791,
1.28193611667e-05, 0.07761194029850746
tau24w-50, 251.18864315095823, 1e-05, 59.5134308716, 1.6985734867,
0.628042843233, 1.1828755499e-05, 0.0297029702970297
tau32w-50, 1584.893192461114, 1e-05, 59.1827472498, 1.67636770712,
0.118792599805, 1.50486965434e-05, 0.12037037037037036
tau26w-50, 398.1071705534973, 1e-05, 59.4719271043, 1.69576262853,
0.624148003895, 1.2729533595e-05, 0.03164556962025317
tau40w-26, 10000.0, 0.0025118864315095794, 27.6317978076, 0.523494226552,
1.75170399221, 4.84262068533e-05, 0.9570552147239264
tau42w-42, 15848.93192461114, 6.309573444801929e-05, 31.8810625555,
0.621986697012, 1.06134371957, 2.87133782759e-05, 0.6057142857142858
tau30w-34, 1000.0, 0.0003981071705534969, 42.8337637627, 0.927105351932,
0.453748782863, 0.000328263877101, 0.31336405529953915
tau24w-34, 251.18864315095823, 0.0003981071705534969, 53.8331843177,
1.36798845671, 0.877312560857, 0.000582020450883, 0.1941747572815534
tau28w-24, 630.9573444801937, 0.003981071705534969, 29.7760395399,
0.572150181565, 1.11684518014, 0.00075294595475, 0.9386503067484663
tau40w-28, 10000.0, 0.0015848931924611126, 27.9016431139, 0.529509463047,
1.68257059396, 4.83052143806e-05, 0.9447852760736196
tau26w-32, 398.1071705534973, 0.000630957344480193, 48.1943093607,
1.11821559926, 0.580331061344, 0.000628751573357, 0.3715415019762846
tau32w-32, 1584.893192461114, 0.000630957344480193, 33.5252406291,
0.662519271899, -0.912366114898, 0.000303245543753, 0.45614035087719296
tau40w-44, 10000.0, 3.9810717055349695e-05, 42.8606006862, 0.927976717966,
0.676728334956, 3.3141430658e-05, 0.2643171806167401
tau28w-46, 630.9573444801937, 2.511886431509577e-05, 58.704879437,
1.64502663846, 0.888023369036, 3.7370821389e-05, 0.05846153846153846
tau42w-20, 15848.93192461114, 0.01, 26.289755313, 0.494008324294, 2.15287244401,
3.09130467562e-05, 0.9570552147239264
tau28w-48, 630.9573444801937, 1.5848931924611107e-05, 59.2274927782,
```

```
1.67934721678, 0.55988315482, 2.32789459507e-05, 0.07761194029850746
tau38w-48, 6309.573444801937, 1.5848931924611107e-05, 54.1725442126,
1.38513469158, -0.159688412853, 2.42953484339e-05, 0.18429003021148035
tau46w-20, 39810.71705534977, 0.01, 22.9614336356, 0.42368065275, 1.6017526777,
1.34749059958e-05, 0.9754601226993865
tau38w-46, 6309.573444801937, 2.511886431509577e-05, 51.0158760062,
1.23559703666, 0.260954235638, 3.50924166443e-05, 0.19933554817275748
tau36w-32, 3981.0717055349733, 0.000630957344480193, 29.4217196053,
0.563970584276, 0.286270691334, 0.000125899642068, 0.9151515151515152
tau22w-32, 158.48931924611142, 0.000630957344480193, 53.9757213742,
1.37515614506, 0.996105160662, 0.000840916966421, 0.20875420875420875
tau44w-44, 25118.864315095823, 3.9810717055349695e-05, 31.8055192952,
0.620159639198, 1.01071080818, 1.82669331851e-05, 0.5081967213114754
tau50w-44, 100000.0, 3.9810717055349695e-05, 18.6437419764, 0.337387306771,
3.00876338851, 5.40343001847e-06, 0.8023952095808383
tau44w-28, 25118.864315095823, 0.0015848931924611126, 25.0429617698,
0.467220845885, 2.24537487829, 1.98913670825e-05, 0.9386503067484663
tau50w-28, 100000.0, 0.0015848931924611126, 16.8071945099, 0.302054819509,
3.1441090555, 5.51678293924e-06, 0.9570552147239264
tau48w-38, 63095.73444801943, 0.00015848931924611126, 20.6334916986,
0.376542561152, 2.90555014606, 8.31315917649e-06, 0.9263803680981595
tau38w-24, 6309.573444801937, 0.003981071705534969, 28.2126856558,
0.536480391322, 0.694255111977, 8.01536273951e-05, 0.9515151515151515
tau46w-42, 39810.71705534977, 6.309573444801929e-05, 25.2092987589,
0.470762527185, 1.0847127556, 1.32200034952e-05, 0.7543859649122807
tau20w-34, 100.0, 0.0003981071705534969, 57.6331160363, 1.57776280233,
0.104186952288, 0.000639207095463, 0.1277258566978193
tau34w-34, 2511.886431509582, 0.0003981071705534969, 33.5353125727,
0.662772249134, -0.819863680623, 0.000191176443403, 0.48520710059171596
tau48w-36, 63095.73444801943, 0.00025118864315095795, 20.2446148981,
0.36881270117, 3.0593962999, 8.34339267264e-06, 0.9386503067484663
tau22w-50, 158.48931924611142, 1e-05, 59.5155033827, 1.69871402961,
0.661148977605, 1.29849496092e-05, 0.016722408026755852
tau36w-50, 3981.0717055349733, 1e-05, 57.8081186375, 1.58847217198,
-0.000973709834452, 1.66015950524e-05, 0.08635097493036212
tau50w-26, 100000.0, 0.0025118864315095794, 16.5174884365, 0.296545537485,
3.29698149951, 5.51216065339e-06, 0.9570552147239264
tau44w-26, 25118.864315095823, 0.0025118864315095794, 24.9146003982,
0.464494313455, 2.42843232717, 1.97880321965e-05, 0.9325153374233128
tau46w-26, 39810.71705534977, 0.0025118864315095794, 22.9505095087,
0.423455784096, 1.4323271665, 1.35685613222e-05, 0.9813664596273292
tau20w-50, 100.0, 1e-05, 59.5498698986, 1.7010470419, 0.630963972736,
1.31127299376e-05, 0.0
tau34w-50, 2511.886431509582, 1e-05, 58.5121819499, 1.63263075391,
0.00097370983449, 1.79474058206e-05, 0.08498583569405099
tau38w-40, 6309.573444801937, 0.0001, 38.2538613959, 0.788445718148,
-1.00876338851, 6.83346243669e-05, 0.4371859296482412
tau22w-34, 158.48931924611142, 0.0003981071705534969, 56.2645440015,
```

```
1.49742847568, 0.456669912366, 0.000587520812564, 0.21003134796238246
tau36w-34, 3981.0717055349733, 0.0003981071705534969, 30.0458781094,
0.578418395296, -0.0749756572541, 0.000126049653348, 0.7485029940119761
tau50w-42, 100000.0, 6.309573444801929e-05, 17.9162746659, 0.323304907313,
3.04089581305, 5.44813124336e-06, 0.8773006134969326
tau44w-42, 25118.864315095823, 6.309573444801929e-05, 29.0624499729,
0.555734769822, 1.30574488802, 1.90666183645e-05, 0.6627218934911243
tau46w-28, 39810.71705534977, 0.0015848931924611126, 22.735766932,
0.419042736761, 1.63583252191, 1.35082942749e-05, 0.9636363636363636
tau38w-22, 6309.573444801937, 0.00630957344480193, 28.3263665974,
0.539038272261, 0.66893865628, 8.00651922234e-05, 0.9565217391304348
tau46w-44, 39810.71705534977, 3.9810717055349695e-05, 26.6034818974,
0.50083870966, 0.992210321324, 1.29579798937e-05, 0.6198830409356725
tau34w-32, 2511.886431509582, 0.000630957344480193, 29.2381082955,
0.559754297013, 1.30574488802, 0.000189825173452, 0.8502994011976048
tau20w-32, 100.0, 0.000630957344480193, 56.5009999762, 1.51089248634,
0.749756572541, 0.000869207522444, 0.20249221183800623
tau48w-30, 63095.73444801943, 0.001, 20.6659266091, 0.377189058532,
2.66796494645, 8.3889127157e-06, 0.950920245398773
tau44w-20, 25118.864315095823, 0.01, 25.2079804078, 0.470734418604,
2.17429406037, 1.98611057085e-05, 0.950920245398773
tau50w-20, 100000.0, 0.01, 16.997387663, 0.305680826555, 2.89191820837,
5.56926863162e-06, 0.9570552147239264
tau46w-50, 39810.71705534977, 1e-05, 40.8731421275, 0.865407014986,
0.27458617332, 9.50076624086e-06, 0.36213991769547327
tau34w-26, 2511.886431509582, 0.0025118864315095794, 29.0968919047,
0.556521810111, 0.405063291139, 0.000199978500057, 0.9447852760736196
tau20w-26, 100.0, 0.0025118864315095794, 46.8249409556, 1.06582120288,
1.35735150925, 0.0026321530155, 0.2749003984063745
tau38w-36, 6309.573444801937, 0.00025118864315095795, 29.8935929469,
0.574876713995, 0.566699123661, 7.8138980113e-05, 0.6826347305389222
tau36w-42, 3981.0717055349733, 6.309573444801929e-05, 46.3326542555,
1.04763495049, 0.873417721519, 6.82016181485e-05, 0.3076923076923077
tau22w-42, 158.48931924611142, 6.309573444801929e-05, 59.1962676934,
1.67726718173, 0.657254138267, 9.01797489301e-05, 0.10060975609756098
tau44w-34, 25118.864315095823, 0.0003981071705534969, 25.5406115603,
0.477845889787, 2.13047711782, 1.9761486642e-05, 0.9386503067484663
tau50w-34, 100000.0, 0.0003981071705534969, 16.8971786238, 0.303769442996,
3.1100292113, 5.51562593987e-06, 0.9386503067484663
tau48w-24, 63095.73444801943, 0.003981071705534969, 20.5177416979,
0.374237657448, 2.74488802337, 8.42650138264e-06, 0.9570552147239264
tau38w-38, 6309.573444801937, 0.00015848931924611126, 31.9738690679,
0.624235383552, -0.117818889971, 7.64118044441e-05, 0.45614035087719296
tau20w-28, 100.0, 0.0015848931924611126, 51.7442269722, 1.26823110008,
0.560856864654, 0.00194170840467, 0.1872791519434629
tau34w-28, 2511.886431509582, 0.0015848931924611126, 29.369151969,
0.56276191526, 0.30282375852, 0.000199619475743, 0.9333333333333333
tau46w-32, 39810.71705534977, 0.000630957344480193, 23.2149213704,
```

```
0.428908848956, 1.20545277507, 1.36103666353e-05, 0.9515151515151515
tau20w-44, 100.0, 3.9810717055349695e-05, 59.4785745673, 1.69621236583,
0.595910418695, 4.0749245143e-05, 0.0429042904290429
tau34w-44, 2511.886431509582, 3.9810717055349695e-05, 53.4945051205,
1.35115141625, 1.18208373905, 5.97369773874e-05, 0.23174603174603176
tau48w-46, 63095.73444801943, 2.511886431509577e-05, 25.5851732383,
0.478801581567, 1.95813047712, 7.91547913411e-06, 0.6568047337278107
tau22w-20, 158.48931924611142, 0.01, 30.0, 0.57735026919, 1.94936708861,
0.0028562759566, 0.9212121212121213
tau36w-20, 3981.0717055349733, 0.01, 28.8058748635, 0.549888184818,
0.495618305745, 0.000126473390152, 0.968944099378882
tau48w-48, 63095.73444801943, 1.5848931924611107e-05, 28.689505981,
0.547245978133, 1.30866601753, 7.68393746569e-06, 0.5138121546961326
tau20w-20, 100.0, 0.01, 33.7206634302, 0.667438273705, 0.233690360273,
0.00451824100281, 0.3815028901734104
tau34w-20, 2511.886431509582, 0.01, 29.1681626464, 0.558152107853,
0.361246348588, 0.000199957411436, 0.9754601226993865
tau38w-30, 6309.573444801937, 0.001, 28.6225385192, 0.545728114718,
0.55111976631, 7.99222288043e-05, 0.950920245398773
tau22w-44, 158.48931924611142, 3.9810717055349695e-05, 59.3426143606,
1.68704896818, 0.622200584226, 5.22472215518e-05, 0.056074766355140186
tau36w-44, 3981.0717055349733, 3.9810717055349695e-05, 51.407714698,
1.25302435735, 0.593962999026, 5.1407577526e-05, 0.22910216718266255
tau50w-32, 100000.0, 0.000630957344480193, 16.7289422008, 0.300565064676,
3.24634858812, 5.50163087901e-06, 0.950920245398773
tau44w-32, 25118.864315095823, 0.000630957344480193, 25.086569992,
0.468148429082, 2.2259006816, 1.98795873207e-05, 0.9325153374233128
tau48w-22, 63095.73444801943, 0.00630957344480193, 20.3551142071,
0.371005170547, 2.98344693281, 8.33717026064e-06, 0.9570552147239264
tau22w-28, 158.48931924611142, 0.0015848931924611126, 47.8296179736,
1.1039926569, 0.680623174294, 0.00162485809139, 0.28185328185328185
tau36w-28, 3981.0717055349733, 0.0015848931924611126, 28.854079452,
0.550984419506, 0.482960077897, 0.000126418383448, 0.9515151515151515
tau46w-34, 39810.71705534977, 0.0003981071705534969, 23.0473994401,
0.425451493401, 1.48977604674, 1.35356121944e-05, 0.9212121212121213
tau34w-42, 2511.886431509582, 6.309573444801929e-05, 50.9916423828,
1.23452891056, 0.0321324245375, 8.7078649434e-05, 0.18021201413427562
tau20w-42, 100.0, 6.309573444801929e-05, 59.4374013645, 1.69342961624,
0.350535540409, 0.000104336130953, 0.04966887417218543
tau48w-40, 63095.73444801943, 0.0001, 21.2655607818, 0.389191422941,
2.84712755599, 8.25563761359e-06, 0.9079754601226994
tau36w-26, 3981.0717055349733, 0.0025118864315095794, 28.9318540804,
0.552755260156, 0.434274586173, 0.00012640810449, 0.9751552795031055
tau22w-26, 158.48931924611142, 0.0025118864315095794, 40.9531592774,
0.867852461598, 2.35443037975, 0.00206888161672, 0.35555555555555557
tau44w-50, 25118.864315095823, 1e-05, 46.058719934, 1.03765640396,
1.09250243427, 1.11528016886e-05, 0.2271062271062271
tau50w-50, 100000.0, 1e-05, 25.7487177194, 0.482315154286, 2.33008763389,
```

```
4.92717494832e-06, 0.39378238341968913
tau26w-26, 398.1071705534973, 0.0025118864315095794, 32.3995160867,
0.634607450219, 1.32035053554, 0.00111955315874, 0.48554913294797686
tau32w-26, 1584.893192461114, 0.0025118864315095794, 29.4473724213,
0.564560864492, 0.26582278481, 0.00031626212615, 0.9751552795031055
tau40w-50, 10000.0, 1e-05, 53.6410479775, 1.35840343034, 1.08276533593,
1.41030660252e-05, 0.19174041297935104
tau42w-34, 15848.93192461114, 0.0003981071705534969, 26.8886005597,
0.507078814809, 2.05452775073, 3.06849430377e-05, 0.9202453987730062
tau24w-42, 251.18864315095823, 6.309573444801929e-05, 58.8369154863,
1.6535997559, 0.565725413827, 0.000112796483824, 0.10144927536231885
tau30w-42, 1000.0, 6.309573444801929e-05, 56.346821739, 1.50209450025,
0.548198636806, 9.21862582003e-05, 0.12912912912912913
tau32w-28, 1584.893192461114, 0.0015848931924611126, 29.7347697756,
0.571194489786, 0.13729308666, 0.000315796053877, 0.9325153374233128
tau26w-28, 398.1071705534973, 0.0015848931924611126, 37.3316846181,
0.762670148682, 0.60564751704, 0.00102162998672, 0.4385026737967914
tau32w-44, 1584.893192461114, 3.9810717055349695e-05, 55.9065743345,
1.47735894831, 0.0886075949367, 6.67562204042e-05, 0.14469453376205788
tau26w-44, 398.1071705534973, 3.9810717055349695e-05, 58.8657859922,
1.65548303088, 0.601752677702, 5.58724917837e-05, 0.1033434650455927
tau40w-32, 10000.0, 0.000630957344480193, 27.9355939144, 0.530268394755,
1.70691333982, 4.82500270435e-05, 0.9447852760736196
tau28w-30, 630.9573444801937, 0.001, 38.2776886573, 0.78912032411,
0.0438169425511, 0.000638585745243, 0.36180904522613067
tau30w-20, 1000.0, 0.01, 29.6180599263, 0.568496065938, 1.19279454722,
0.000475430533605, 0.950920245398773
tau24w-20, 251.18864315095823, 0.01, 29.9528704232, 0.576254034501,
0.0253164556962, 0.00199085888682, 0.9636363636363636
tau32w-20, 1584.893192461114, 0.01, 29.3226516192, 0.561693789154,
0.327166504382, 0.000316427402013, 0.9570552147239264
tau26w-20, 398.1071705534973, 0.01, 30.0, 0.57735026919, 1.0, 0.0011922929913,
0.9570552147239264
tau42w-32, 15848.93192461114, 0.000630957344480193, 26.84887262, 0.506207448775,
1.90749756573, 3.08323686759e-05, 0.950920245398773
tau30w-44, 1000.0, 3.9810717055349695e-05, 57.6012411297, 1.57582331019,
0.310613437196, 6.39148738264e-05, 0.11918604651162791
tau24w-44, 251.18864315095823, 3.9810717055349695e-05, 59.1696391114,
1.67549634108, 0.667964946446, 5.34121932159e-05, 0.06289308176100629
tau30w-28, 1000.0, 0.0015848931924611126, 30.0084544209, 0.577547029262,
0.967867575463, 0.000475043325995, 0.7005988023952096
tau24w-28, 251.18864315095823, 0.0015848931924611126, 41.8921390125,
0.897001060875, 1.01557935735, 0.00139589407015, 0.47417840375586856
tau28w-38, 630.9573444801937, 0.00015848931924611126, 54.6422445987,
1.40933618047, 0.179162609542, 0.00022438572306, 0.14006514657980457
tau26w-42, 398.1071705534973, 6.309573444801929e-05, 58.5670173592,
1.63614432663, 0.398247322298, 8.85046787604e-05, 0.11799410029498525
tau32w-42, 1584.893192461114, 6.309573444801929e-05, 53.8169125141,
```

```
1.36717330784, 1.24732229796, 8.53597603054e-05, 0.23659305993690852
tau40w-34, 10000.0, 0.0003981071705534969, 28.3450819224, 0.539459900988,
1.57838364167, 4.8079658654e-05, 0.8834355828220859
tau28w-36, 630.9573444801937, 0.00025118864315095795, 51.024160646,
1.23596244822, 1.02044790652, 0.000312638686296, 0.26022304832713755
tau42w-50, 15848.93192461114, 1e-05, 50.8200104055, 1.22699581065,
0.407010710808, 1.29308509309e-05, 0.2621359223300971
tau24w-26, 251.18864315095823, 0.0025118864315095794, 36.505104587,
0.740098957535, 0.210321324245, 0.00168077883011, 0.39037433155080214
tau30w-26, 1000.0, 0.0025118864315095794, 29.9069062806, 0.575185908395,
1.04965920156, 0.000474823714498, 0.9386503067484663
```

Convert the data into a pandas DataFrame. This allows us to sort the data set according to τ and w.

[4]:

```python
import pandas as pd
```

[5]:

```python
data = {'tau_s' : slip_interval, 'w': weath_param, 'angle': facet_angle,␣
 ↪'erorate': ero_rate,
        'fracsoil' : frac_soil}
df = pd.DataFrame(data)
df = df.sort_values(by=['w', 'tau_s'])
df
```

[5]:

```
             tau_s         w      angle   erorate  fracsoil
175     100.000000  0.000010  59.549870  0.000013  0.000000
170     158.489319  0.000010  59.515503  0.000013  0.016722
140     251.188643  0.000010  59.513431  0.000012  0.029703
142     398.107171  0.000010  59.471927  0.000013  0.031646
127     630.957344  0.000010  59.325438  0.000013  0.031746
139    1000.000000  0.000010  59.225806  0.000013  0.077612
141    1584.893192  0.000010  59.182747  0.000015  0.120370
176    2511.886432  0.000010  58.512182  0.000018  0.084986
171    3981.071706  0.000010  57.808119  0.000017  0.086351
62     6309.573445  0.000010  56.017080  0.000017  0.183673
229   10000.000000  0.000010  53.641048  0.000014  0.191740
253   15848.931925  0.000010  50.820010  0.000013  0.262136
225   25118.864315  0.000010  46.058720  0.000011  0.227106
190   39810.717055  0.000010  40.873142  0.000010  0.362140
43    63095.734448  0.000010  32.437388  0.000007  0.360976
226  100000.000000  0.000010  25.748718  0.000005  0.393782
81      100.000000  0.000016  59.544078  0.000014  0.013468
79      158.489319  0.000016  59.473174  0.000020  0.013201
111     251.188643  0.000016  59.469018  0.000028  0.066667
113     398.107171  0.000016  59.397380  0.000028  0.071207
154     630.957344  0.000016  59.227493  0.000023  0.077612
112    1000.000000  0.000016  58.914798  0.000023  0.135057
114    1584.893192  0.000016  58.314235  0.000024  0.129310
80     2511.886432  0.000016  57.437759  0.000025  0.121813
78     3981.071706  0.000016  55.684682  0.000026  0.156156
```

```
155    6309.573445  0.000016  54.172544  0.000024  0.184290
0     10000.000000  0.000016  50.825795  0.000020  0.267559
26    15848.931925  0.000016  45.825087  0.000017  0.222222
61    25118.864315  0.000016  41.051306  0.000014  0.195745
27    39810.717055  0.000016  34.612589  0.000011  0.367876
..             ...       ...        ...       ...       ...
120     251.188643  0.006310  30.733715  0.001982  0.672727
107     398.107171  0.006310  29.997584  0.001193  0.898204
132     630.957344  0.006310  29.734770  0.000753  0.938650
119    1000.000000  0.006310  29.734770  0.000475  0.957055
106    1584.893192  0.006310  29.322652  0.000316  0.975460
71     2511.886432  0.006310  29.100581  0.000200  0.951515
87     3981.071706  0.006310  28.860256  0.000126  0.921212
183    6309.573445  0.006310  28.326367  0.000080  0.956522
21    10000.000000  0.006310  27.568557  0.000048  0.957055
5     15848.931925  0.006310  26.541647  0.000031  0.950920
33    25118.864315  0.006310  24.964918  0.000020  0.957055
54    39810.717055  0.006310  22.720696  0.000014  0.975460
216   63095.734448  0.006310  20.355114  0.000008  0.957055
34   100000.000000  0.006310  16.876534  0.000006  0.957055
209     100.000000  0.010000  33.720663  0.004518  0.381503
206     158.489319  0.010000  30.000000  0.002856  0.921212
240     251.188643  0.010000  29.952870  0.001991  0.963636
242     398.107171  0.010000  30.000000  0.001192  0.957055
25      630.957344  0.010000  29.818488  0.000753  0.938650
239    1000.000000  0.010000  29.618060  0.000475  0.950920
241    1584.893192  0.010000  29.322652  0.000316  0.957055
210    2511.886432  0.010000  29.168163  0.000200  0.975460
207    3981.071706  0.010000  28.805875  0.000126  0.968944
28     6309.573445  0.010000  28.383739  0.000080  0.981366
128   10000.000000  0.010000  27.534377  0.000048  0.950920
153   15848.931925  0.010000  26.289755  0.000031  0.957055
188   25118.864315  0.010000  25.207980  0.000020  0.950920
156   39810.717055  0.010000  22.961434  0.000013  0.975460
77    63095.734448  0.010000  20.497962  0.000008  0.950920
189  100000.000000  0.010000  16.997388  0.000006  0.957055

[256 rows x 5 columns]
```

Calculate the dimensionless slip rate. The slip rate is defined as:
$V = \sqrt{3}\delta / \tau_s$
The dimensionless slip rate is:
$V^* = V / \delta d$
We also want to calculate the $w - d$ ratio, which we'll also call w^*.

For purposes of these calculations, we happen to know that the runs were performed with $d = 10^{-4}$ yr^{-1}.

[6]:

```
d = 1.0e-4
delta = 0.5
df['V'] = 3.0**0.5 * delta / df['tau_s']
df['Vstar'] = df['V'] / (delta * d)
df['wstar'] = df['w'] / d
df
```

[6]:

```
             tau_s         w      angle   erorate  fracsoil         V  \
175     100.000000  0.000010  59.549870  0.000013  0.000000  0.008660
170     158.489319  0.000010  59.515503  0.000013  0.016722  0.005464
140     251.188643  0.000010  59.513431  0.000012  0.029703  0.003448
142     398.107171  0.000010  59.471927  0.000013  0.031646  0.002175
127     630.957344  0.000010  59.325438  0.000013  0.031746  0.001373
139    1000.000000  0.000010  59.225806  0.000013  0.077612  0.000866
141    1584.893192  0.000010  59.182747  0.000015  0.120370  0.000546
176    2511.886432  0.000010  58.512182  0.000018  0.084986  0.000345
171    3981.071706  0.000010  57.808119  0.000017  0.086351  0.000218
62     6309.573445  0.000010  56.017080  0.000017  0.183673  0.000137
229   10000.000000  0.000010  53.641048  0.000014  0.191740  0.000087
253   15848.931925  0.000010  50.820010  0.000013  0.262136  0.000055
225   25118.864315  0.000010  46.058720  0.000011  0.227106  0.000034
190   39810.717055  0.000010  40.873142  0.000010  0.362140  0.000022
43    63095.734448  0.000010  32.437388  0.000007  0.360976  0.000014
226  100000.000000  0.000010  25.748718  0.000005  0.393782  0.000009
81      100.000000  0.000016  59.544078  0.000014  0.013468  0.008660
79      158.489319  0.000016  59.473174  0.000020  0.013201  0.005464
111     251.188643  0.000016  59.469018  0.000028  0.066667  0.003448
113     398.107171  0.000016  59.397380  0.000028  0.071207  0.002175
154     630.957344  0.000016  59.227493  0.000023  0.077612  0.001373
112    1000.000000  0.000016  58.914798  0.000023  0.135057  0.000866
114    1584.893192  0.000016  58.314235  0.000024  0.129310  0.000546
80     2511.886432  0.000016  57.437759  0.000025  0.121813  0.000345
78     3981.071706  0.000016  55.684682  0.000026  0.156156  0.000218
155    6309.573445  0.000016  54.172544  0.000024  0.184290  0.000137
0     10000.000000  0.000016  50.825795  0.000020  0.267559  0.000087
26    15848.931925  0.000016  45.825087  0.000017  0.222222  0.000055
61    25118.864315  0.000016  41.051306  0.000014  0.195745  0.000034
27    39810.717055  0.000016  34.612589  0.000011  0.367876  0.000022
..             ...       ...        ...       ...       ...       ...
120     251.188643  0.006310  30.733715  0.001982  0.672727  0.003448
107     398.107171  0.006310  29.997584  0.001193  0.898204  0.002175
132     630.957344  0.006310  29.734770  0.000753  0.938650  0.001373
119    1000.000000  0.006310  29.734770  0.000475  0.957055  0.000866
106    1584.893192  0.006310  29.322652  0.000316  0.975460  0.000546
71     2511.886432  0.006310  29.100581  0.000200  0.951515  0.000345
87     3981.071706  0.006310  28.860256  0.000126  0.921212  0.000218
183    6309.573445  0.006310  28.326367  0.000080  0.956522  0.000137
```

```
21    10000.000000  0.006310  27.568557  0.000048  0.957055  0.000087
5     15848.931925  0.006310  26.541647  0.000031  0.950920  0.000055
33    25118.864315  0.006310  24.964918  0.000020  0.957055  0.000034
54    39810.717055  0.006310  22.720696  0.000014  0.975460  0.000022
216   63095.734448  0.006310  20.355114  0.000008  0.957055  0.000014
34   100000.000000  0.006310  16.876534  0.000006  0.957055  0.000009
209     100.000000  0.010000  33.720663  0.004518  0.381503  0.008660
206     158.489319  0.010000  30.000000  0.002856  0.921212  0.005464
240     251.188643  0.010000  29.952870  0.001991  0.963636  0.003448
242     398.107171  0.010000  30.000000  0.001192  0.957055  0.002175
25      630.957344  0.010000  29.818488  0.000753  0.938650  0.001373
239    1000.000000  0.010000  29.618060  0.000475  0.950920  0.000866
241    1584.893192  0.010000  29.322652  0.000316  0.957055  0.000546
210    2511.886432  0.010000  29.168163  0.000200  0.975460  0.000345
207    3981.071706  0.010000  28.805875  0.000126  0.968944  0.000218
28     6309.573445  0.010000  28.383739  0.000080  0.981366  0.000137
128   10000.000000  0.010000  27.534377  0.000048  0.950920  0.000087
153   15848.931925  0.010000  26.289755  0.000031  0.957055  0.000055
188   25118.864315  0.010000  25.207980  0.000020  0.950920  0.000034
156   39810.717055  0.010000  22.961434  0.000013  0.975460  0.000022
77    63095.734448  0.010000  20.497962  0.000008  0.950920  0.000014
189  100000.000000  0.010000  16.997388  0.000006  0.957055  0.000009

          Vstar      wstar
175  173.205081   0.100000
170  109.285018   0.100000
140   68.954185   0.100000
142   43.507149   0.100000
127   27.451155   0.100000
139   17.320508   0.100000
141   10.928502   0.100000
176    6.895418   0.100000
171    4.350715   0.100000
62     2.745116   0.100000
229    1.732051   0.100000
253    1.092850   0.100000
225    0.689542   0.100000
190    0.435071   0.100000
43     0.274512   0.100000
226    0.173205   0.100000
81   173.205081   0.158489
79   109.285018   0.158489
111   68.954185   0.158489
113   43.507149   0.158489
154   27.451155   0.158489
112   17.320508   0.158489
114   10.928502   0.158489
```

```
80     6.895418    0.158489
78     4.350715    0.158489
155    2.745116    0.158489
0      1.732051    0.158489
26     1.092850    0.158489
61     0.689542    0.158489
27     0.435071    0.158489
..          ...         ...
120   68.954185   63.095734
107   43.507149   63.095734
132   27.451155   63.095734
119   17.320508   63.095734
106   10.928502   63.095734
71     6.895418   63.095734
87     4.350715   63.095734
183    2.745116   63.095734
21     1.732051   63.095734
5      1.092850   63.095734
33     0.689542   63.095734
54     0.435071   63.095734
216    0.274512   63.095734
34     0.173205   63.095734
209  173.205081  100.000000
206  109.285018  100.000000
240   68.954185  100.000000
242   43.507149  100.000000
25    27.451155  100.000000
239   17.320508  100.000000
241   10.928502  100.000000
210    6.895418  100.000000
207    4.350715  100.000000
28     2.745116  100.000000
128    1.732051  100.000000
153    1.092850  100.000000
188    0.689542  100.000000
156    0.435071  100.000000
77     0.274512  100.000000
189    0.173205  100.000000

[256 rows x 8 columns]
```

We can now look at how the weathering-rate parameter influences erosion rate. Start with some dimensional analysis. Our variables are:

- vertical erosion rate, E_v (L/T)
- slip interval, τ_s (T)
- weathering rate parameter, w (1/T)
- disturbance rate parameter, d (1/T)

- cell width, δ (L)
- facet angle, θ (-)

From these, we can construct dimensionless parameters that are inter-related: $\frac{E_v \tau_s}{\delta} = f\left(\theta, \frac{w}{d}\right)$.

This suggests that a logical approach to plotting is to plot dimensionless erosion rate (basically erosion rate relative to slip rate) as a function of angle, for each value of w.

An alternative is: $\frac{E_v}{\delta d} = f\left(\theta, \frac{V}{\delta d}, \frac{w}{d}\right)$

[7]:

```
Eprime1 = df['erorate'] * df['tau_s'] / delta
Eprime2 = df['erorate'] / (delta * d)
```

[8]:

```
a = 1.8
delta = 0.5
max_wx_rate = 2.0 * a * delta * df['w']
En = df['erorate'] * np.cos(np.radians(df['angle']))
df['ero_w_ratio'] = En / max_wx_rate
df
```

[8]:

```
             tau_s         w      angle   erorate  fracsoil         V  \
175     100.000000  0.000010  59.549870  0.000013  0.000000  0.008660
170     158.489319  0.000010  59.515503  0.000013  0.016722  0.005464
140     251.188643  0.000010  59.513431  0.000012  0.029703  0.003448
142     398.107171  0.000010  59.471927  0.000013  0.031646  0.002175
127     630.957344  0.000010  59.325438  0.000013  0.031746  0.001373
139    1000.000000  0.000010  59.225806  0.000013  0.077612  0.000866
141    1584.893192  0.000010  59.182747  0.000015  0.120370  0.000546
176    2511.886432  0.000010  58.512182  0.000018  0.084986  0.000345
171    3981.071706  0.000010  57.808119  0.000017  0.086351  0.000218
62     6309.573445  0.000010  56.017080  0.000017  0.183673  0.000137
229   10000.000000  0.000010  53.641048  0.000014  0.191740  0.000087
253   15848.931925  0.000010  50.820010  0.000013  0.262136  0.000055
225   25118.864315  0.000010  46.058720  0.000011  0.227106  0.000034
190   39810.717055  0.000010  40.873142  0.000010  0.362140  0.000022
43    63095.734448  0.000010  32.437388  0.000007  0.360976  0.000014
226  100000.000000  0.000010  25.748718  0.000005  0.393782  0.000009
81      100.000000  0.000016  59.544078  0.000014  0.013468  0.008660
79      158.489319  0.000016  59.473174  0.000020  0.013201  0.005464
111     251.188643  0.000016  59.469018  0.000028  0.066667  0.003448
113     398.107171  0.000016  59.397380  0.000028  0.071207  0.002175
154     630.957344  0.000016  59.227493  0.000023  0.077612  0.001373
112    1000.000000  0.000016  58.914798  0.000023  0.135057  0.000866
114    1584.893192  0.000016  58.314235  0.000024  0.129310  0.000546
80     2511.886432  0.000016  57.437759  0.000025  0.121813  0.000345
78     3981.071706  0.000016  55.684682  0.000026  0.156156  0.000218
155    6309.573445  0.000016  54.172544  0.000024  0.184290  0.000137
0     10000.000000  0.000016  50.825795  0.000020  0.267559  0.000087
26    15848.931925  0.000016  45.825087  0.000017  0.222222  0.000055
```

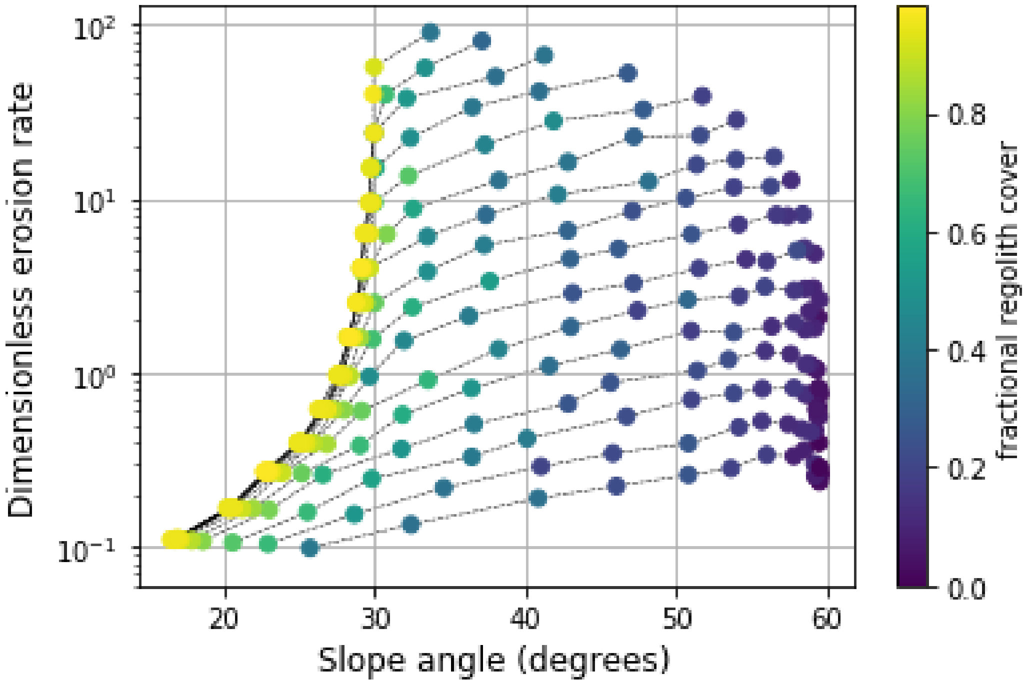
```
61    25118.864315  0.000016  41.051306  0.000014  0.195745  0.000034
27    39810.717055  0.000016  34.612589  0.000011  0.367876  0.000022
..             ...       ...        ...       ...       ...       ...
120     251.188643  0.006310  30.733715  0.001982  0.672727  0.003448
107     398.107171  0.006310  29.997584  0.001193  0.898204  0.002175
132     630.957344  0.006310  29.734770  0.000753  0.938650  0.001373
119    1000.000000  0.006310  29.734770  0.000475  0.957055  0.000866
106    1584.893192  0.006310  29.322652  0.000316  0.975460  0.000546
71     2511.886432  0.006310  29.100581  0.000200  0.951515  0.000345
87     3981.071706  0.006310  28.860256  0.000126  0.921212  0.000218
183    6309.573445  0.006310  28.326367  0.000080  0.956522  0.000137
21    10000.000000  0.006310  27.568557  0.000048  0.957055  0.000087
5     15848.931925  0.006310  26.541647  0.000031  0.950920  0.000055
33    25118.864315  0.006310  24.964918  0.000020  0.957055  0.000034
54    39810.717055  0.006310  22.720696  0.000014  0.975460  0.000022
216   63095.734448  0.006310  20.355114  0.000008  0.957055  0.000014
34   100000.000000  0.006310  16.876534  0.000006  0.957055  0.000009
209     100.000000  0.010000  33.720663  0.004518  0.381503  0.008660
206     158.489319  0.010000  30.000000  0.002856  0.921212  0.005464
240     251.188643  0.010000  29.952870  0.001991  0.963636  0.003448
242     398.107171  0.010000  30.000000  0.001192  0.957055  0.002175
25      630.957344  0.010000  29.818488  0.000753  0.938650  0.001373
239    1000.000000  0.010000  29.618060  0.000475  0.950920  0.000866
241    1584.893192  0.010000  29.322652  0.000316  0.957055  0.000546
210    2511.886432  0.010000  29.168163  0.000200  0.975460  0.000345
207    3981.071706  0.010000  28.805875  0.000126  0.968944  0.000218
28     6309.573445  0.010000  28.383739  0.000080  0.981366  0.000137
128   10000.000000  0.010000  27.534377  0.000048  0.950920  0.000087
153   15848.931925  0.010000  26.289755  0.000031  0.957055  0.000055
188   25118.864315  0.010000  25.207980  0.000020  0.950920  0.000034
156   39810.717055  0.010000  22.961434  0.000013  0.975460  0.000022
77    63095.734448  0.010000  20.497962  0.000008  0.950920  0.000014
189  100000.000000  0.010000  16.997388  0.000006  0.957055  0.000009

          Vstar       wstar  ero_w_ratio
175  173.205081    0.100000     0.369188
170  109.285018    0.100000     0.365963
140   68.954185    0.100000     0.333398
142   43.507149    0.100000     0.359228
127   27.451155    0.100000     0.355287
139   17.320508    0.100000     0.364395
141   10.928502    0.100000     0.428304
176    6.895418    0.100000     0.520791
171    4.350715    0.100000     0.491367
62     2.745116    0.100000     0.523972
229    1.732051    0.100000     0.464494
253    1.092850    0.100000     0.453843
```

```
225    0.689542    0.100000     0.429953
190    0.435071    0.100000     0.399117
43     0.274512    0.100000     0.314816
226    0.173205    0.100000     0.246553
81   173.205081    0.158489     0.253437
79   109.285018    0.158489     0.351872
111   68.954185    0.158489     0.498960
113   43.507149    0.158489     0.501974
154   27.451155    0.158489     0.417491
112   17.320508    0.158489     0.412936
114   10.928502    0.158489     0.436528
80     6.895418    0.158489     0.478774
78     4.350715    0.158489     0.520164
155    2.745116    0.158489     0.498498
0      1.732051    0.158489     0.434560
26     1.092850    0.158489     0.420237
61     0.689542    0.158489     0.382321
27     0.435071    0.158489     0.314361
..          ...         ...          ...
120   68.954185   63.095734     0.150017
107   43.507149   63.095734     0.090986
132   27.451155   63.095734     0.057578
119   17.320508   63.095734     0.036329
106   10.928502   63.095734     0.024292
71     6.895418   63.095734     0.015377
87     4.350715   63.095734     0.009754
183    2.745116   63.095734     0.006206
21     1.732051   63.095734     0.003775
5      1.092850   63.095734     0.002436
33     0.689542   63.095734     0.001581
54     0.435071   63.095734     0.001098
216    0.274512   63.095734     0.000688
34     0.173205   63.095734     0.000467
209  173.205081  100.000000     0.208781
206  109.285018  100.000000     0.137423
240   68.954185  100.000000     0.095831
242   43.507149  100.000000     0.057364
25    27.451155  100.000000     0.036286
239   17.320508  100.000000     0.022962
241   10.928502  100.000000     0.015327
210    6.895418  100.000000     0.009700
207    4.350715  100.000000     0.006157
28     2.745116  100.000000     0.003924
128    1.732051  100.000000     0.002388
153    1.092850  100.000000     0.001540
188    0.689542  100.000000     0.000998
156    0.435071  100.000000     0.000689
```

```
77      0.274512  100.000000     0.000435
189     0.173205  100.000000     0.000296

[256 rows x 9 columns]
```

[9]:

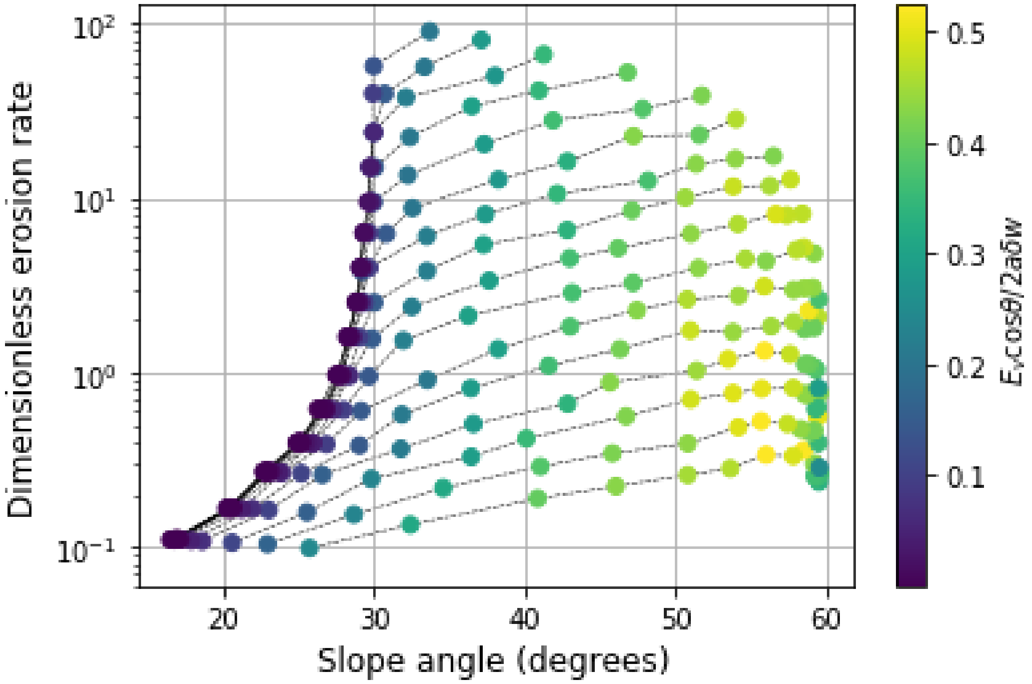
```
facet_angle = df['angle'].values.reshape((16, 16))
nd_ero_rate = Eprime2.values.reshape((16, 16))
fracsoil = df['fracsoil'].values.reshape(16, 16)
psyms = ['ko-', 'k^-', 'k.-', 'k+-', 'kx-', 'k*-']
fig = plt.figure()
ax = plt.gca()
sc = ax.scatter(facet_angle, nd_ero_rate, c=fracsoil, zorder=2)
ax.set_yscale('log')
for i in range(0, 16, 1):
    ax.plot(facet_angle[i], nd_ero_rate[i], 'k--', zorder=1, linewidth=0.5)
plt.xlabel('Slope angle (degrees)', fontsize=12)
plt.ylabel('Dimensionless erosion rate', fontsize=12)
plt.grid(True)
fig.colorbar(sc, label='fractional regolith cover')
plt.savefig('ero_rate_vs_slope_angle_reg_cover.pdf')
```



[10]:

```
ero_w_ratio = df['ero_w_ratio'].values.reshape(16, 16)
ero_w_ratio_capped = np.minimum(ero_w_ratio, 1.0)
psyms = ['ko-', 'k^-', 'k.-', 'k+-', 'kx-', 'k*-']
fig = plt.figure()
```

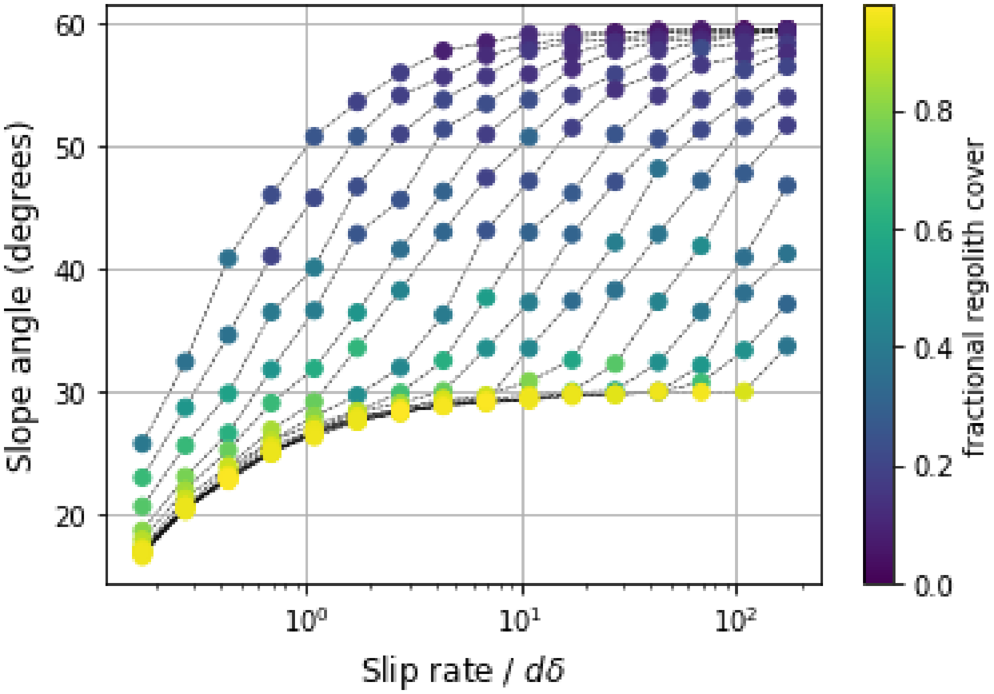
```
ax = plt.gca()
sc = ax.scatter(facet_angle, nd_ero_rate, c=ero_w_ratio, zorder=2)
ax.set_yscale('log')
for i in range(0, 16, 1):
    ax.plot(facet_angle[i], nd_ero_rate[i], 'k--', zorder=1, linewidth=0.5)
plt.xlabel('Slope angle (degrees)', fontsize=12)
plt.ylabel('Dimensionless erosion rate', fontsize=12)
plt.grid(True)
fig.colorbar(sc, label=r'$E_v \cos\theta / 2a\delta w$')
plt.savefig('ero_rate_vs_slope_angle_by_eroratio.pdf')
```



Plot slip rate versus facet angle:

[11]:
```
fracsoil = df['fracsoil'].values.reshape(16, 16)
nd_slip_rate = df['V'].values.reshape(16, 16) / (delta * d)
psyms = ['ko-', 'k^-', 'k.-', 'k+-', 'kx-', 'k*-']
fig = plt.figure()
ax = plt.gca()
sc = ax.scatter(nd_slip_rate, facet_angle, c=fracsoil, zorder=2)
ax.set_xscale('log')
for i in range(0, 16, 1):
    ax.plot(nd_slip_rate[i], facet_angle[i], 'k--', zorder=1, linewidth=0.5)
plt.ylabel('Slope angle (degrees)', fontsize=12)
plt.xlabel(r'Slip rate / $d\delta$', fontsize=12)
plt.grid(True)
fig.colorbar(sc, label='fractional regolith cover')
```

```
plt.savefig('slope_angle_vs_slip_rate_by_frac_soil.pdf')
```



Fin.

S5. Table of mathematical symbols

Table S1. List of mathematical symbols (L = length dimension, T = time dimension).

Symbol	Description	Dimensions
a	surface-roughness coefficient	-
d	disturbance transition frequency	T^{-1}
d'	dimensionless disturbance frequency $= d\tau$	-
D	characteristic disturbance rate $= \delta d$	LT^{-1}
D_s	soil-creep transport coefficient	$\mathrm{L}^2\mathrm{T}^{-1}$
E_n	slope-normal erosion rate	LT^{-1}
E_v	vertical erosion rate	LT^{-1}
E_v'	dimensionless vertical erosion rate $= E_v/\delta d$	-
q_s	sediment flux per unit width	$\mathrm{L}^2\mathrm{T}^{-1}$
s	dissolution transition frequency	T^{-1}
S'	dimensionless dissolution efficiency	-
S	slope gradient	-
S_c	critical slope gradient	-
V	fault slip rate $= \delta/\tau$	LT^{-1}
w	weathering transition frequency	T^{-1}
w'	dimensionless weathering efficiency $= w\tau$	-
W	characteristic weathering rate $= \delta w$	LT^{-1}
W_s	characteristic dissolution rate $= \delta s$	LT^{-1}
α	fault dip angle	-
β	$\alpha - \theta$	-
δ	cell width	L
θ	facet dip angle	-
τ	unit slip interval $= \tau_s/\sqrt{3}$	T
τ_s	lattice slip interval	T

S6. Coordinates of profiles in Figures 2 and 3.

Table S2. Coordinates of Wasatch fault system facet profiles shown in Figure 2.

Latitude (degrees N)	Longitude (degrees W)	Fault segment
39.7857	111.8232	Nephi
39.5562	111.8336	Levan
41.6406	112.0590	Brigham City
39.8223	111.8126	Nephi
39.4327	111.9142	Levan
39.6095	111.8273	Levan
39.2890	111.8642	Fayette
39.2510	111.8449	Fayette

Table S3. Coordinates of Wasatch fault system facet profiles shown in Figure 3.

Latitude (degrees N)	Longitude (degrees W)	Fault segment
Panel (a):		
39.9145	111.8013	Nephi
39.7804	111.8237	Nephi
39.9098	111.8002	Nephi
39.2890	111.8642	Fayette
Panel (b):		
39.7550	111.8290	Nephi
39.5224	111.8640	Levan
39.5281	111.8640	Levan
39.2510	111.8449	Fayette
Panel (c):		
39.8199	111.8128	Nephi
39.8187	111.8133	Nephi
39.7775	111.8243	Nephi
39.8173	111.8136	Nephi

References

Niño, Y., García, M., & Ayala, L. (1994). Gravel saltation: 1. experiments. Water resources research, 30(6), 1907-1914.

Tucker, Gregory E., Daniel EJ Hobley, Eric Hutton, Nicole M. Gasparini, Erkan Istanbulluoglu, Jordan M. Adams, and Sai Siddartha Nudurupati. "CellLab-CTS 2015: continuous-time stochastic cellular automaton modeling using Landlab." Geoscientific Model Development 9, no. 2 (2016): 823.

Tucker, G. E., McCoy, S. W., Whittaker, A. C., Roberts, G. P., Lancaster, S. T., & Phillips, R. (2011). Geomorphic significance of postglacial bedrock scarps on normal-fault footwalls. Journal of Geophysical Research: Earth Surface, 116(F1).