

PM Space Demonstrator Manual

By: Kevin Claytor
LANL EES-GEO

Contents

1 Introduction	3
2 Discussion of PM Space	3
3 Operation	4
4 Walkthrough	9
Appendix A: Discussion of Code	9

1. Introduction

PM Space (described below), is a fairly simple system whose multiple elements can generate fairly complex hysteretic behavior. It is used to model hysteresis in magnets, rocks, piezoelements, shape memory alloys, and many other physical systems. The case taken in this demonstration is that of a rock. Here PM space is used to model the quasi-static (the slow application of stress) behavior, specifically the strain that is induced in the rock. The PM space is a phenomenological model, where the individual elements currently do not represent any physical analogue; however, they are thought to correspond to cracks and/or grain bonding. Moreover, because each element displays individual hysteresis, the PM space is a memory-dependent phenomenon, it matters what the previous values of, in this case, stress were.

This program was created to show how a stress – strain loop is generated through applying and releasing stress. It shows the current stress value, the current state of PM space, and the resultant stress – strain loop.

2. Discussion of PM Space

PM space is a method of simulating hysteresis originally developed for magnetic domains and the hysteresis they show. The possibility of using it to simulate rock hysteresis has been developed by Holcomb [1981], and more thoroughly by Guyer [1996]. It functions as a distribution of Hysteretic Elastic Units (HEU). As the pressure increases to a closing pressure P_c , the unit collapses to a length L_c and subsequently stays there. When the pressure is decreased, the unit stays closed until an open pressure P_o is reached, with P_o at or below P_c , when the unit returns to its open length L_o , forming the hysteresis loop (Figure 1).

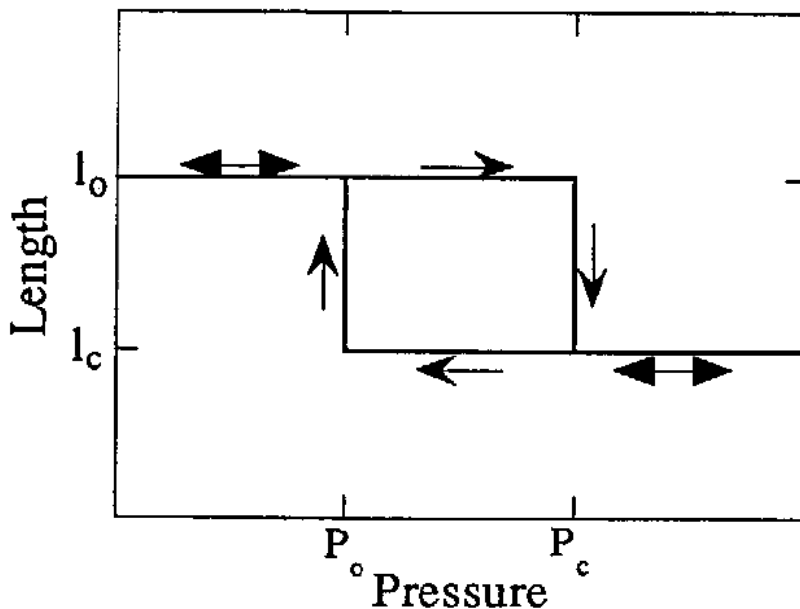


Figure 1: The open and close behavior of a HEU. The tendency to open and resume the previous state at a lower pressure than the pressure at which the HEU closes at results in the hysteretic behavior. It is noted here that they HEU is currently a hypothetical construct; no physical analogue is yet known, however, many have been proposed (small cracks and grain bonding being the main competitors).

The open and close pressure is determined by a function chosen, so that the behavior of all HEUs mimics the rock in question. This forms a PM space as illustrated in Figure 2, with many of the elements along the line $P_o = P_c$ meaning that they show no hysteresis, leaving the elements off of the diagonal responsible for hysteresis.

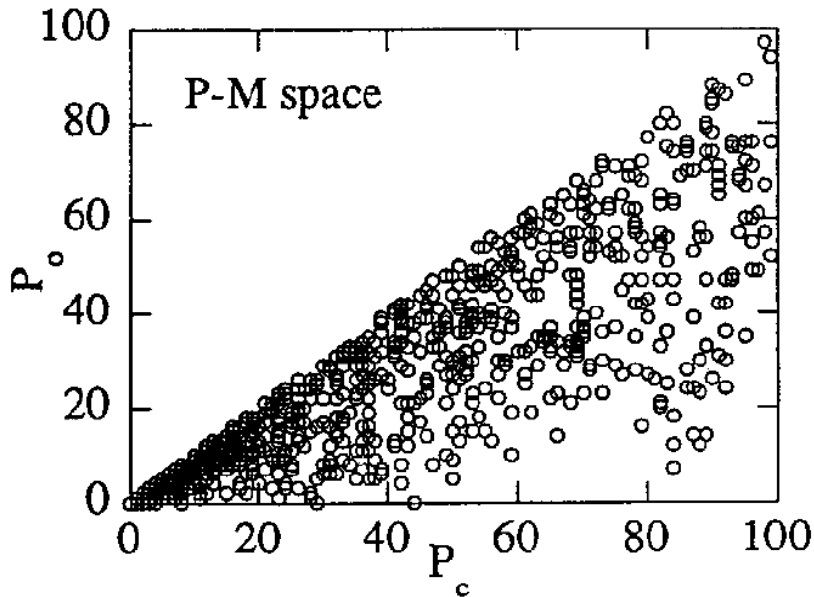


Figure 2: The distribution in PM space of many HEUs. Here two models are shown denoted by the circles and squares. Each distribution gives different hysteretic behavior.

Given the behavior of one HEU, it is easy to see how this model accounts for the overall hysteresis. By having stress procedures more complicated than a simple up down, procedure, which would generate a single hysteresis loop, multiple loops, some interior to others, can be generated as well. It will be left to the reader to investigate this further.

3. Operation

Launch the PMDemo m-file by typing "PMDemo" into the Matlab's command window and pressing Enter. A window such as that in Figure 3 will appear.

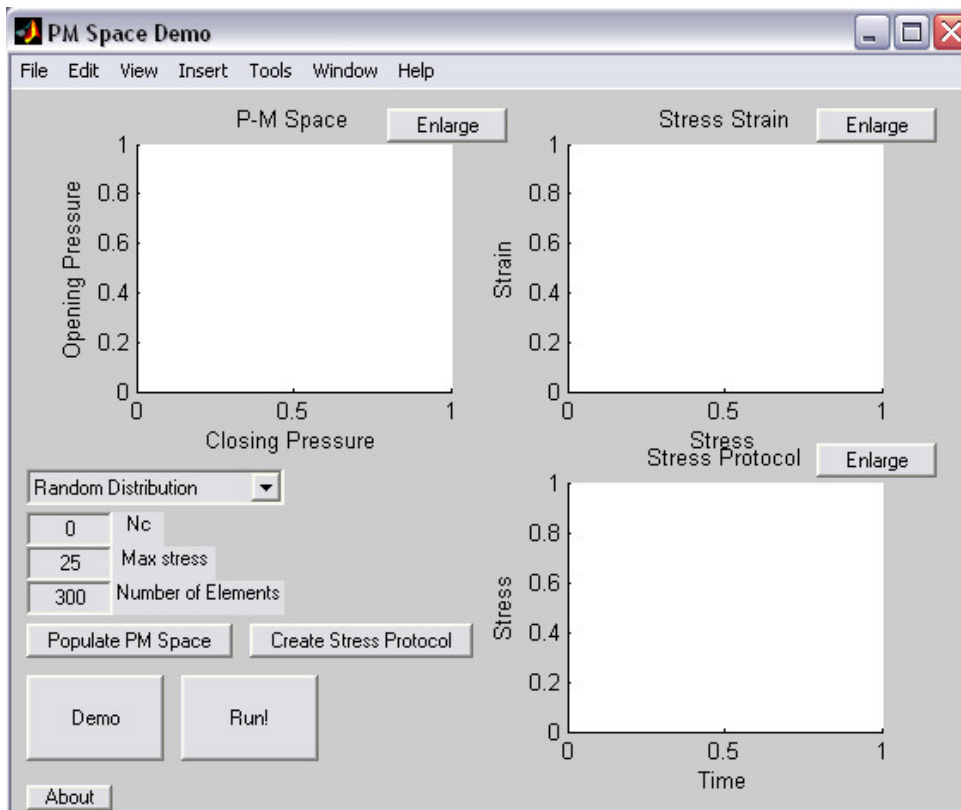


Figure 3: PMDemo upon launching.

At this point clicking “Demo” will create a random PM distribution and load a stress protocol, and proceed to run that stress protocol (Figure 4).

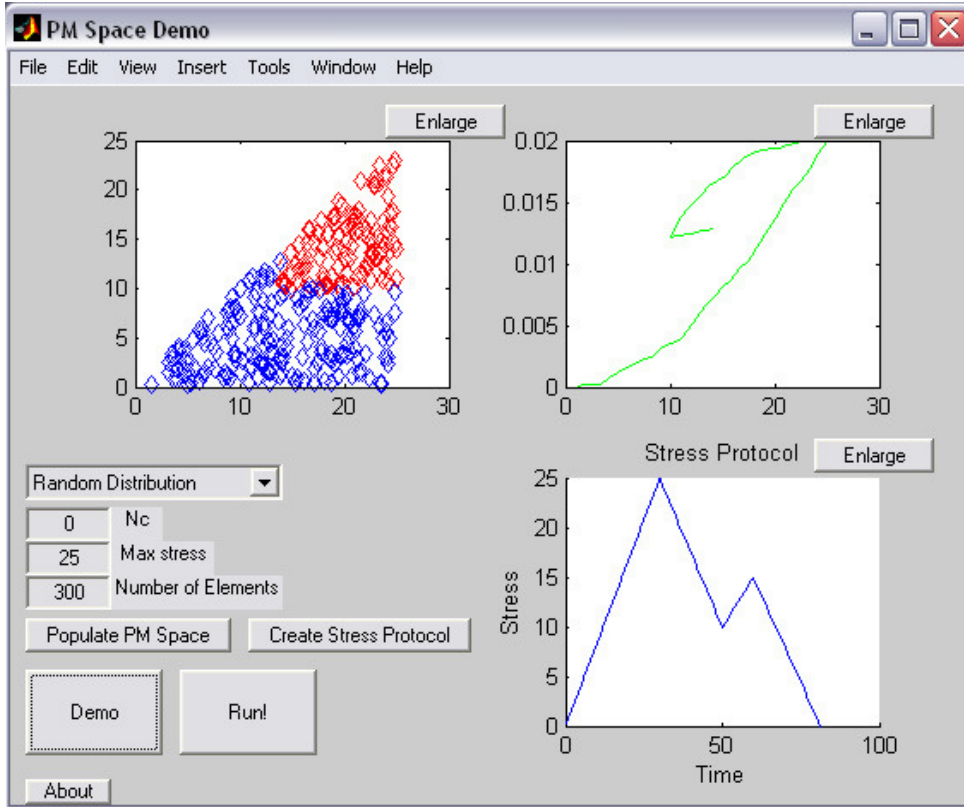


Figure 4: Demo mode running

A custom setup can be run by first generating a PM space. Use the pull down menu to select the method used to fill. Random generates a random opening and closing pressure independently of each other. The second and third methods generate a PM space according to Equations 1 and 2 respectively;

$$P_c = mr^2 \quad (1)$$

$$P_o = P_c \sqrt{r}$$

$$P_c = mr^2$$

$$P_o = P_c r^{.25+.75N_c} \quad (2)$$

where r is a random number, and m is the maxstress, and N_c is a constant chosen by the user, through the boxes below the pulldown menu. N_c must be zero or positive, and operates to skew the distribution away from the line $P_c = P_o$, see Figure 5. Max stress sets the maximum stress value that the PM space will be able to handle. The option below that controls the number of elements that are created in PM space.

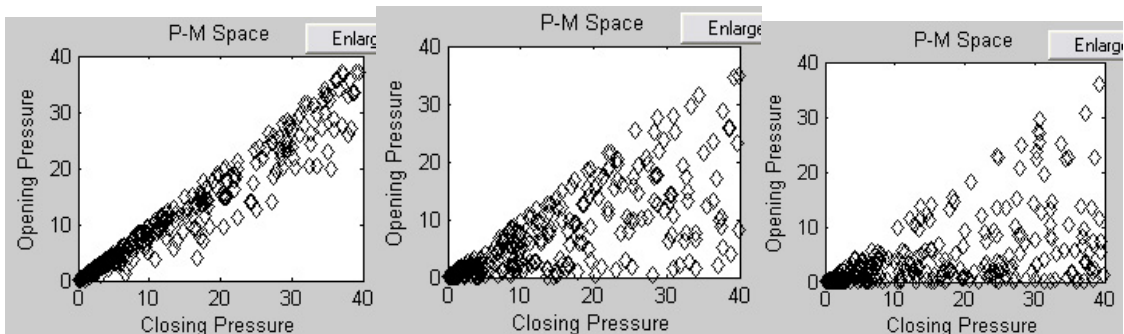


Figure 5: The effects of different N_c values. 0 – left, 1 – middle, and 2 – right, are shown. As N_c increases, the PM space skews away from the diagonal.

Once the pull down menu, N_c , max stress, and the number of elements are set, clicking “Populate PM Space” will run the function that generates PM space according to the previous three values.

With PM Space generated, a Stress Protocol can be created. Click on the “Create Stress Protocol.” This launches another window (Figure 6) where clicking in the graph will create a linear stress protocol. Clicking the right mouse button ends the stress protocol, and closes the window.

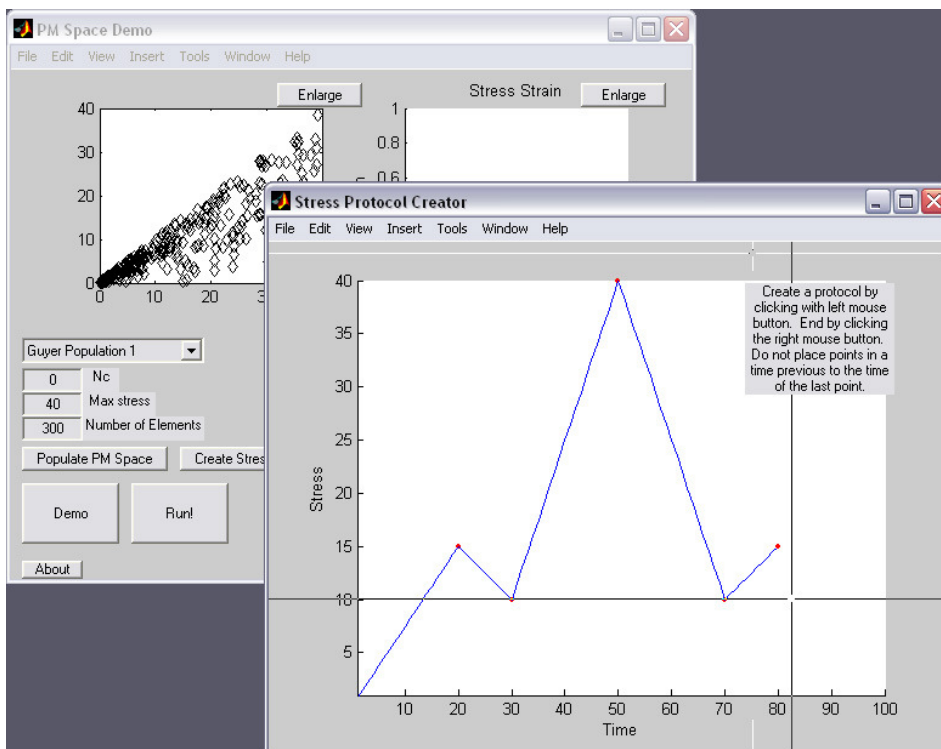


Figure 6: Creating a stress protocol.

Once PM space has been generated, and a stress protocol has been generated, the button “Run!” will put the model through the stress protocol and display the results in the stress-strain plot (Figure 7).

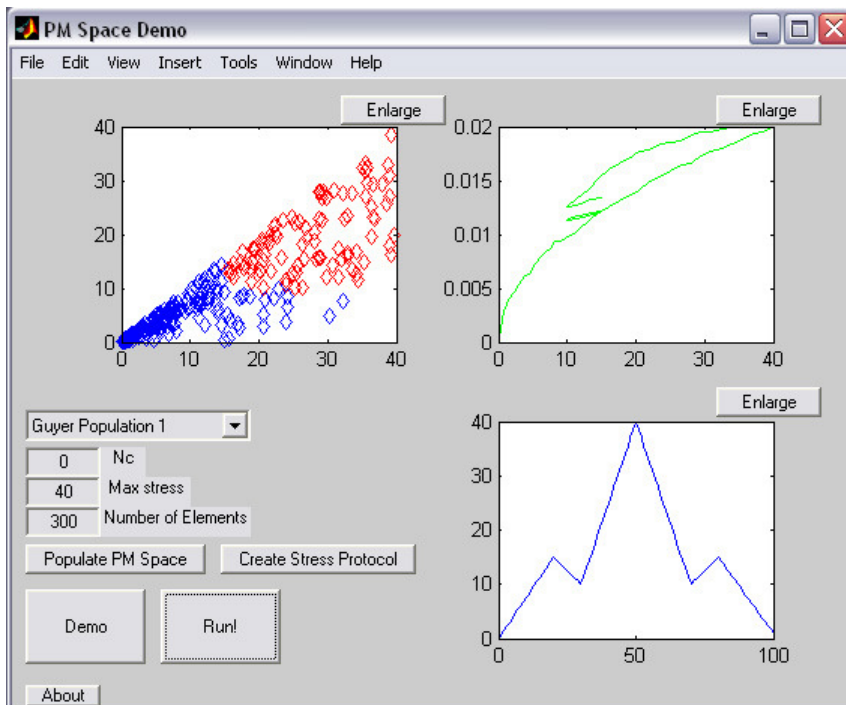


Figure 7: A custom stress protocol running with a custom PM space.

Clicking on the “Enlarge” button near any of the three graphs will enlarge that graph in a new window. Note that if a new window is launched while the stress protocol is running, the three graphs will update in that window.

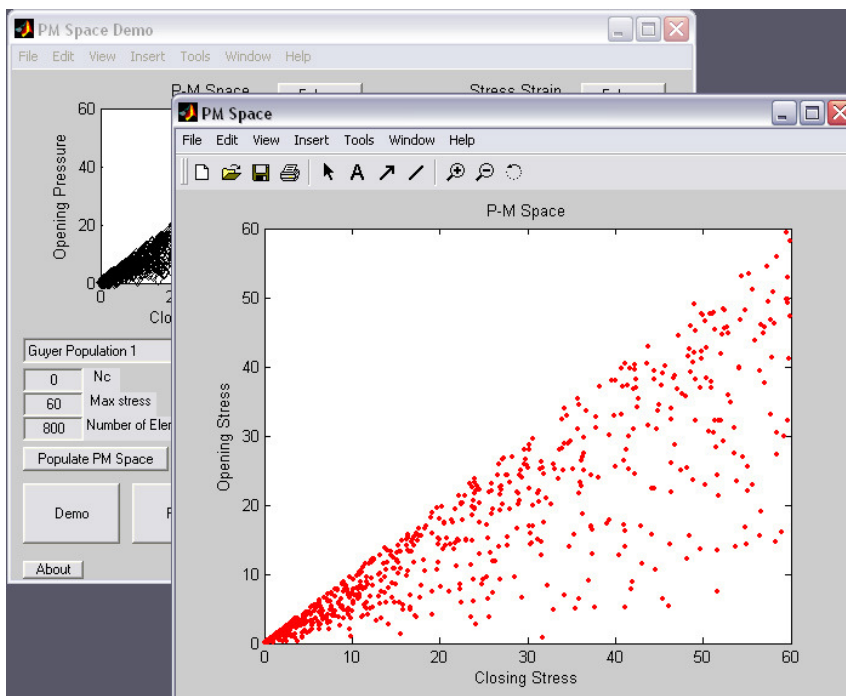


Figure 8: The PM space enlarged in a new window.

4. Walkthrough

Since the demo mode is pretty self explanatory (click “Demo”), the walkthrough will be for a custom run. With PMDemo open, click the pull down menu and select the second option. Since this option does not take advantage of N_c , leave that set at zero. Now change Max stress to 60, and set the number of elements to 800 (the more elements there are the smoother the stress-strain curve will be). Click “Populate PM Space” to generate our PM space, and then click “Create Stress Protocol” to launch the stress protocol creator. Follow the directions in the window that is created, or just click to make a pattern similar to that in Figure 6 and 7. Now that everything’s set up click “Run!” and sit back and watch the results!

Appendix A: Discussion of Code

Most of the code is fairly straightforward, however, there are a couple of important issues to consider.

The generation of PM elements is based on random numbers, however, the closing pressure always has to be greater than or equal to the opening pressure due to the physical constraints of the system. Guyer’s method assures this result, but if independent random numbers are used then a check must be made. Note that all the elements are initialized to the open state, as this is the unstressed state of the rock.

The creation of the stress protocol is also fairly simple, the previous point is recorded, which, in conjunction with the current point, is used to get a point-slope equation of the line for the segment between the current point and the last point. This equation is then used to fill in the points in-between, and store in the protocol.

The main loop used to create the stress-strain loop is displayed below. The tricky part comes in here, as hysteresis is a history dependent phenomenon. As mentioned above, the elements are initialized as open. As the stress increases, they remain open until the pressure exceeds their closing pressure and they close. They do not switch states back to the closed position until the stress is below their closing pressure. This recording of history is an important subtlety, as one may think that at any pressure one can just count the number of open and closed elements, but this is false, as it does not take into account which elements have been closed in the past.

The calculation of strain is then based on the fraction of elements that have closed and is then divided by the total initial length, which was randomly set here to be a fairly large positive number (to give a strain value closer to experiment), but could also be determined by the total elements (which would give a max strain of 1).

```

numelements = 300;    %Set some defaults
maxstress = 25;
state = 1;
nc = 0;
myPM=generatePMspace(numelements,maxstress,state,nc);    % Generate our PM space
Protocol = generateStress(25,'demo');    % Generate the stress-strain protocol
time = length(Protocol);    % using the prestored demo value
l = 50;    %Set an initial length to better model the behavior we want.
Strain=[];    %Initialize our strain vector
hold on    %Don't ditch this window
for i = 1:time
    pressure=Protocol(i,2); % Find the current pressure
    %initialize our closed and open matrices every time, because
    % then we can graph them and get a good idea of what's going
    % on as it happens!
    Open=[];
    Closed=[];
    for j = 1:numelements
        if pressure > myPM(j,1)    %if the pressure is higher than the
            if myPM(j,3) == 0    % closing pressure close it if it's open
                myPM(j,3) = 1;
            end
        elseif pressure < myPM(j,2)    %if the pressure is lower than the
            if myPM(j,3) == 1    % opening pressure open it if it's closed
                myPM(j,3) = 0;
            end
        end
    end
end

for k = 1:numelements
    if myPM(k,3)==1
        Closed=[Closed;myPM(k,:)];
    else
        Open=[Open;myPM(k,:)];
    end
end

if length(Open) ~= 0
    subplot(2,2,1), plot(Open(:,1),Open(:,2),'rd');
    hold on
end
if length(Closed) ~= 0
    subplot(2,2,1), plot(Closed(:,1),Closed(:,2),'bd');
end
dl = length(Closed)/numelements;    %find the strain
Strain = [Strain;(dl/l)];    %record the strain
subplot(2,2,2), plot(Protocol(1:length(Strain),2),Strain,'g-');    %Plot the strain on a subplot
subplot(2,2,4), plot(Protocol(:,1),Protocol(:,2),'r')    %Plot the stress protocol
%Plot the current position of the stress protocol
subplot(2,2,4), plot(Protocol(1:length(Strain),1),Protocol(1:length(Strain),2),'b')
pause(0.1);    %wait,
drawnow    % draw,
end    % and move on to the next time step

```