```
x = discretize_load(10,27.25,2001)
```

x = 10×2
       1.3625          4907.5
       4.0875          4634.8
       6.8125          4089.5
       9.5375          3544.3
       12.262            2999
       14.987          2453.7
       17.712          1908.5
       20.438          1363.2
       23.163          817.91
       25.888          272.64

```
format shortG
[y, moment] = wall_reactions(x)
```

y =
       26991
moment =
    2.4851e+05

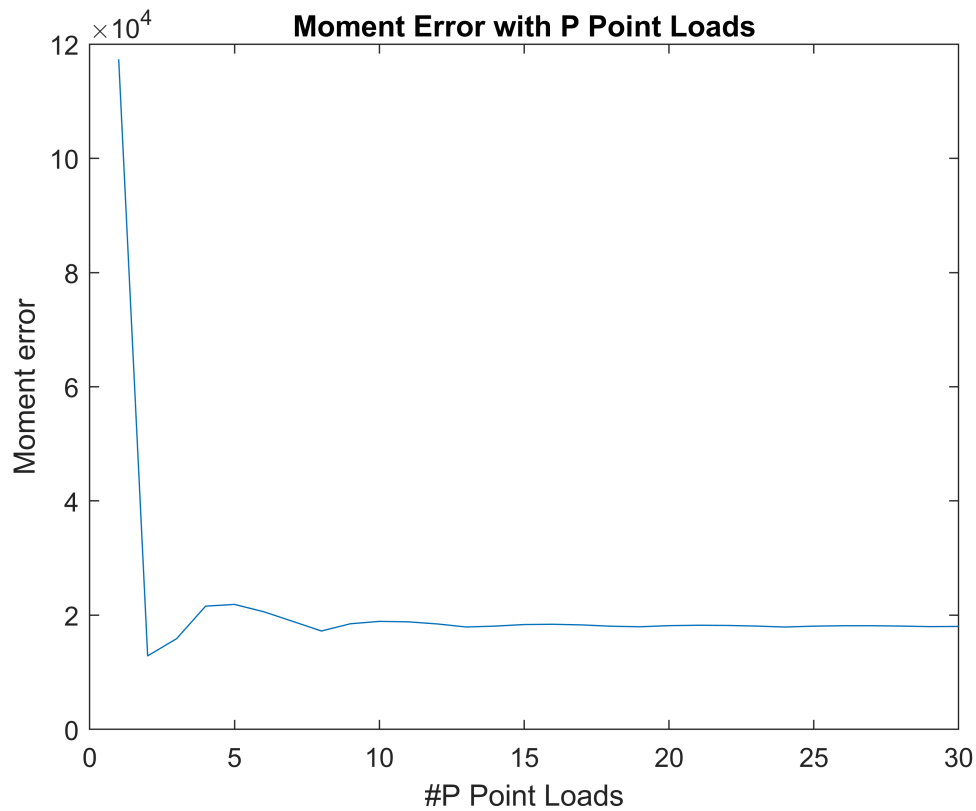```
error = moment_error(x,27.25,2001)
```

error =
       18914

```
errors = loopP(2001,27.25)
```

errors = 1×30
    1.1735e+05          12878          15888          21584          21886          20617 ···

```
plot((1:30),errors)
xlabel("#P Point Loads")
ylabel("Moment error")
title("Moment Error with P Point Loads")
```

Moment Error with P Point Loads

Midpoint rule approximation.

```
function errorP = loopP(w,L)
    for k = 1:30
        f = discretize_load(k,L,w);
        errorVec(k) = moment_error(f,L,w);
    end
    errorP = errorVec;
end
function error = moment_error(matrix,L,w)
    point = (3*L)/16;
    holderMatrix = matrix(matrix(:,1) >= point,:);
    newMatrix = [point - holderMatrix(:,1), holderMatrix(:,2)];
    Mpoint = 0;
    for d = 1:size(newMatrix,1)
        Mpoint = Mpoint - (newMatrix(d,1) * newMatrix(d,2));
    end
    Mpoint;
    Mdist = ((w * point^2) / 3) - ((L * w * point) / 2) + ((w * (1 - point / L) * point^2)/6);
    error = Mpoint + Mdist;
end
function [Fy, M] = wall_reactions(load)
    Fy = sum(load(:,2));
    M = 0;
    if size(load,1) == 1
        M = (load(1,1)*load(1,2));
    else
```

```matlab
        for j = 1:length(load)
            M = M + (load(j,1)*load(j,2));
        end
    end
end
function resForce = discretize_load(p,L,w) % using midpoint approximation, assuming uniform loa
    dx = L/p;
    resForce = zeros(p,2);
    if p == 1
        resForce(2,1) = L/2;
        resForce(2,2) = w*(1-(resForce(2,1)./L))*dx;
        resForce(1,:) = [];
    else
        for i = 1:p
            resForce(2,1) = dx/2; % since it's just the center of each cut, start with dx/2
            resForce(i+1,1) = resForce(i,1) + dx; % every other point is a multiple of dx away
            resForce(i+1,2) = w*(1-(resForce(i+1,1)./L)) * dx;
        end
        resForce(1,:) = [];
    end
end
```