1. **Intquad.m**

function Q = intquad(n,m)

Q = zeros(n,m);

Q = [Q Q+1; Q+2 Q+3];

End

1. **sindeg.m**

function [s avg] = sindeg(d)

s = sin(d\*pi/180);

avg = mean(s(:));

end

1. **simple\_stats.m**

function S = simple\_stats(N)

S = [mean(N,2) median(N,2) min(N,[],2) max(N,[],2)];

end

OR

function S = simple\_stats(N)

S = [mean(N'); median(N'); min(N'); max(N')]';

end

1. **odd\_rms.m**

function or = odd\_rms(nn)

or = sqrt(mean((1:2:2\*nn).^2));

end

1. **fence.m**

function [nums , nump] = fence(lng,seg)

nums = ceil(lng/seg);

nump = nums+1;

end

1. **zero\_stat.m**

function pct = zero\_stat(A)

pct = 100 \* (numel(A) - sum(A(:))) / numel(A);

end

1. **reverse\_diag.m**

function D = reverse\_diag(n)

D = zeros(n);

D(n:max([1,n-1]):max([n,n^2-1])) = 1;

end

OR

function D = reverse\_diag(n)

D = zeros(n);

D(1:n+1:end) = 1;

D = flip(D);

end

1. **sum3and5muls.m**

function s = sum3and5muls(n)

s = sum(3:3:n) + sum(5:5:n) - sum(15:15:n);

end