In class I made a claim. Given
\[
f(x) = \begin{cases} 
0 & \text{if } |x| > 1 \\
1 & \text{if } |x| \leq 1 
\end{cases},
\]
let \( f_1 = f \ast f, f_2 = f \ast f, f_3 = f \ast f \ast f \), etc. Then define two scalars for each function,
\[\eta_n = (f_n(0))^{-1}\]
(to fix the height) and
\[\mu_n = \int_{-\infty}^{\infty} \eta_n f_n(x) \, dx\]
(to fix the area) The claim is this–
\[\lim_{n \to \infty} \eta_n f_n(\mu x) = e^{-x^2}.\]

We’ll be able to prove this using the Fourier transform. For now, I offer computational evidence that this is likely to be true. The graphs of a few of the \( \eta_n f_n(\mu x) \) are plotted against \( e^{-x^2} \). After that is some repetitive Matlab code that makes those plots.

(I avoided writing a subroutine, since this may be some of the first Matlab code some of you have seen.)
$b(x) = e^{-x^2}$

$f(x) = 1$ on $[-1, 1]$

$f * f$

$f * f * f$ (rescaled)

$f * f * f * f$ (rescaled)
area = sqrt(pi);
height = 1;

xIncrement = 0.005;

x1 = -1 : xIncrement : 1;
y1 = ones(size(x1));

x = x1;
y = y1;
midIndex = ceil(length(y)/2);
yfactor = height / y(midIndex);
delta = (x(length(x)) - x(1)) / (length(x)+1);
I = sum(y) * delta;
xfactor = area / (I * yfactor)
y = y .* yfactor;
x = x .* xfactor;

x2 = -2 : xIncrement : 2;
y2 = conv(y1,y1);

x = x2;
y = y2;
midIndex = ceil(length(y)/2);
yfactor = height / y(midIndex);
delta = (x(length(x)) - x(1)) / (length(x)+1);
I = sum(y) * delta;
xfactor = area / (I * yfactor)
y = y .* yfactor;
x = x .* xfactor;

x3 = -3 : xIncrement : 3;
y3 = conv(y1,y2);

x = x3;
y = y3;
midIndex = ceil(length(y)/2);
yfactor = height / y(midIndex);
delta = (x(length(x)) - x(1)) / (length(x)+1);
I = sum(y) * delta;
xfactor = area / (I * yfactor)
y = y .* yfactor;
x = x .* xfactor;

x3 = x;
y3 = y;
x4 = -4 : xIncrement : 4;
y4 = conv(y1,y3);

x = x4;
y = y4;
midIndex = ceil( length(y)/2 );
yfactor = height / y(midIndex);
delta = (x(length(x)) - x(1)) / (length(x)+1);
I = sum(y) * delta;
xfactor = area/(I*yfactor)
y = y .* yfactor;
x = x .* xfactor;
x4 = x;
y4 = y;

yNormal = exp( - x4 .* x4 );

plot(x1,y1,x2,y2,x3,y3,x4,y4,x4,yNormal)