```matlab
function [rho,rho_a] = FPE_1D(Nelem,h)
% $Author : Vignesh Ramakrishnan$
% $RIN : 662028006$  $Date : November 10, 2021$
% $Code Version: 1.0$
% This is script to solve the 1D Fokker Planck equation using Finite Elements
% 1D FPE: Drho/Dt = -a*rho + h^2/2 (D2rho/Dx^2)
% Assume boundary conditions = 0 on either end of it
% Inputs: Nelem : Number of elements on the mesh
%         h     : The diffusion coefficient parameter
% Output: rho   : FEM solution to FPE
%         rho_a : Analytical solution of FPE

x1_lim1 = -3;
x1_lim2 = 3;
mesh = generate1Dmesh(Nelem,x1_lim1,x1_lim2);

order = 1;
c_coeff = 1;
fespace = FiniteElementSpace(mesh,order);

% Go through each element and compute Bilinear forms - diffusion and
% convection
% mass integrator choice -1 , convection - 2, diffusion - 3

Nnodes = fespace(end).ElemDOF(end);

D = single(zeros(Nnodes)); % diffusion overall matrix
C = single(zeros(Nnodes)); % convection overall matrix
W = single(zeros(Nnodes));

for i=1:Nelem
    d = Diffusion_Integrator(-h^2/2,order,fespace(i));
    c = Convection_Integrator(c_coeff,order,fespace(i));
    e = fespace(i).ElemDOF(1);
    D = Assemble(D,d,e);
    C = Assemble(C,c,e);
end



for i=1:Nnodes
    if i == 1 || i == Nnodes
        w(1,i) = 0.5 * (x1_lim2-x1_lim1)/Nnodes;
    else
        w(1,i) = (x1_lim2-x1_lim1)/Nnodes;
    end
end

for i=1:Nnodes
    W(i,:) = w;
end

A = D+C+W;
X = ones(Nnodes,1);

rho = A\X;

% plotting
R = @(x) sqrt(3/(pi*h^2))*exp((-3*x.^2)/(h^2));
```

```
x = (x1_lim1:(x1_lim2-x1_lim1)/(length(rho)-1):x1_lim2)' ;
rho_a = R(x);
plot(x,rho,'r','lineWidth',2);
hold on;
grid on;
plot(x,rho_a,'k--','lineWidth',2);
xlabel('x');
ylabel('$\rho$','Interpreter','latex');
legend('FEM approximate solution','Exact solution');
title('FEM solution to 1D FPE','a = 3, h = 0.8');

end
```