

```

1. function nz = getNz(Ui, meff, dx, boundL, boundR)
2.     e = 1.6e-19; eVtoJ = e; JtoEv = e^(-1);
3.     hbar = 1.0551*1e-34; k_B = 1.38e-23;
4.     T=300;
5.
6.     EFermi=1.51e-20;
7.
8.     Nc3D = 4*pi*((2*meff(1)/((2*pi*hbar)^2))^(3/2));
9.     Nc3DActive = sqrt(2)*(meff(end)^(3/2))*k_B*T/(((2*pi)^2)*hbar^3);
10.
11.     function res = NEz(Ui, meff, dx, Ez, U1, Un)
12.         [waveL, waveR] = getWaveFunction(dx, meff, Ui, Ez);
13.
14.         waveL = abs(waveL).^2;
15.         waveR = abs(waveR).^2;
16.
17.         Ez = repmat(Ez', 1, length(Ui));
18.
19.         waveL = Nc3DActive*((waveL)./sqrt(Ez - Ui(1))).*log(1 + exp((EFermi + U1 - Ez)/(k_B*T)));
20.         waveL(Ez <= Ui(1)) = 0;
21.
22.         waveR = Nc3DActive*(waveR)./sqrt(Ez - Ui(end)).*log(1 + exp((EFermi + Un - Ez)/(k_B*T)));
23.         waveR(Ez <= Ui(end)) = 0;
24.
25.         res = waveL + waveR;
26.     end
27.
28.     foo = @(Ez) NEz( Ui(boundL: boundR), meff(boundL: boundR), dx, Ez, Ui(1), Ui(end) );
29.     nzA = integral(foo, Ui(end), 2*e, 'AbsTol', 1E-25, 'ArrayValued', true);
30.
31.     U1 = Ui(1: boundL-1);
32.     U2 = Ui(boundR + 1: end);
33.
34.     nzL = zeros(1, length(U1));
35.     nzR = zeros(1, length(U2));
36.
37.     for I = 1 : length(U1)
38.         foo = @(Ez) sqrt(Ez - U1(I))./(1 + exp((Ez - (EFermi + Ui(1)))/(k_B*T)));
39.         nzL(I) = Nc3D*integral(foo, U1(I), 2*e, 'AbsTol', 1E-25);
40.     end
41.
42.     for I = 1 : length(U2)
43.         foo = @(Ez) sqrt(Ez - U2(I))./(1 + exp((Ez - (EFermi + Ui(end)))/(k_B*T)));
44.         nzR(I) = Nc3D*integral(foo, U2(I), 2*e, 'AbsTol', 1E-25);
45.     end
46.
47.     nz = [nzL, nzA, nzR];
48. end

```