```
function [out1, out2] = Magnetoexziton(Input,Method,Potential)

[n, lambda, phi]    = deal(Input.n, Input.lambda, Input.phi) ;
dim                 = n+1 ;


switch Potential
    case 'Coulomb'
        const       =     (2*lambda/pi)^0.5 ;
        VC_ij       = @(n) - const
 *exp(gamma_prefactor(0:n)) .*F32(0:n) ;
    case 'Keldysh'
end

Hmx_ii      = @(n)      diag     (  lambda*(2*(0:n)+1) ) ;
Inh_ii      = @(n,phi)  eye(dim)*( -phi    -1i*0.2     ) ;

H           = Hmx_ii(n) + VC_ij(n) ;


switch Method
    case 'Spektrum'
        b           = ones (     dim    ,1) ;
        X           = zeros(length(phi) ,1) ;

        for i=1:length(phi)
            A       = H + Inh_ii(n,phi(i)) ;
            x_n     = linsolve(A,b) ;
            X(i)    = sum(x_n)*lambda/pi ;
        end
        out1        = X ;
        out2        = phi ;
    case 'Eigenwerte'
%          disp('Eigenwerte')
        % Bestimmung der Eigenwerte (eig_val) samt Normierung der
 Wellenfunktionen
        % (states). Beides beginnend mit dem Grundzustand (sort).
        [states, EW]   = eig(H,'vector');
        [EW, idx]      = sort(EW);
        states         = states(:,idx);
        EW             = EW(EW<50) ;
        states         = states(:,EW<50) ;
        [nn,g_nn]      = integrate([0 dim],dim,4) ;
        norm           = sqrt(2*pi*(states.^2)'*(nn.*g_nn));
        for i=1:length(EW); states(:,i) = states(:,i)*1/
norm(i)*sign(states(1,i)); end

        % Anzeigen der Grundzustandsenergie
%         disp(EW(1))
        out1            = EW ;
        out2            = states ;
end
```

Published with MATLAB® R2016a