Table of Contents


```
clc, clear, close all
load 'constants.mat'
```

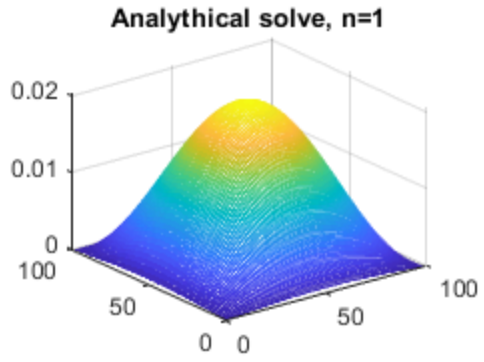
Handle-functions

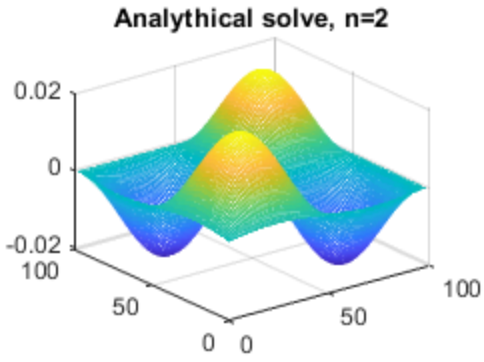
```
Energy = @ (n, L, meff) (hbar * pi * n / (L * 1e-9))^2 / (2 * meff * m0) *
 J2eV * 1e3;
```

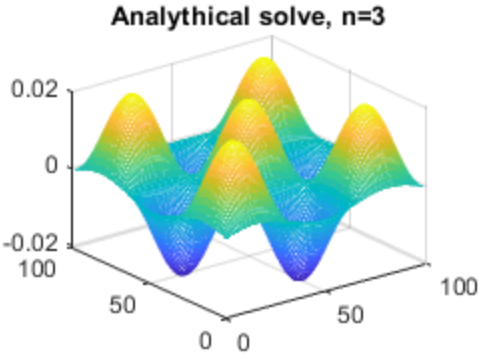
Ex.1 Analythical solve

In ex.1 we need to compare analythical and numeric solves for n = 1, 2, 3 by plotting.

```
figure('Name', 'Analythical solve')
hold on
L = 98;
h = 1;
for n = 1:3
    for x = 1:L
        for y = 1:L
            Psin(x, y) = (2 / L) * (sin(pi * n * x / L)) * (sin(pi * n * y /
 L));
        end
    end
    nexttile
    mesh(1:L, 1:L, Psin)
    title(strcat('Analythical solve, n= ', num2str(n)))
end
hold off
```





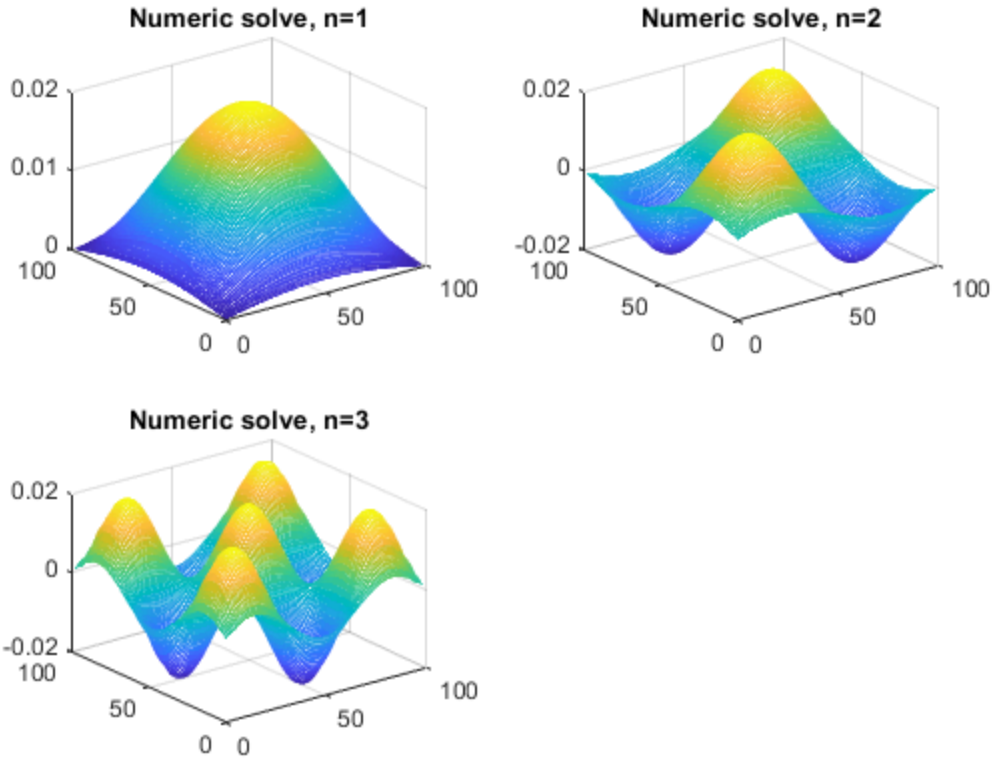


Ex.1 Numeric solve

```
figure('Name', 'Numeric solve')
N = 98;
I = eye(N, N);
T = I;
for i = 1:N
    for j = 1:N
        if i == j
            T(i, j) = 4;
            continue
        end
        if i == j - 1 || i == j + 1 || i == j - 5 || i == j + 5
            T(i, j) = -1;
        end
    end
end
[psi, D] = eig(T);
psi = psi ./ sqrt(h);

for n = 1:3
    vals = psi(:, n);
    for x = 1:L
        for y = 1:L
            psi_r(x, y) = vals(x) * vals(y);
```

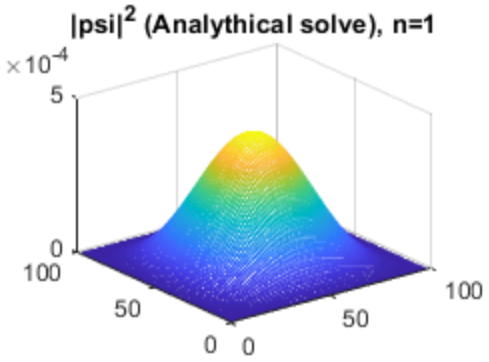
```
        end
    end
    nexttile
    mesh(1:L, 1:L, psi_r)
    title(strcat('Numeric solve, n= ', num2str(n)))
end
```

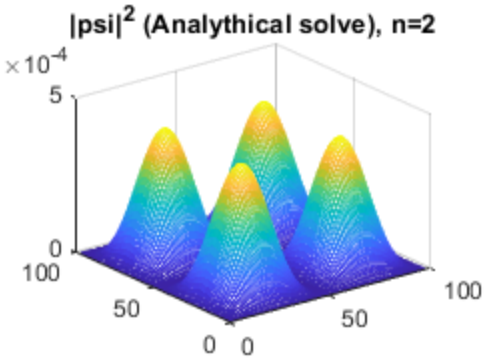


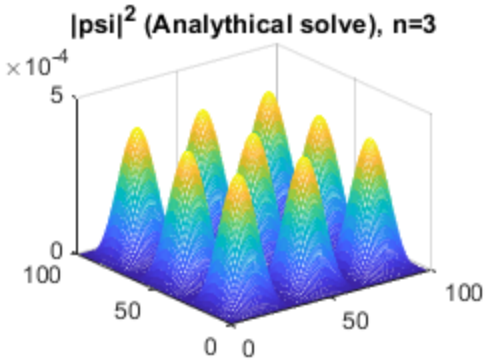
Ex.2 abs(psi)^2 (Analythical solve)

In ex.2 we need the same thing we had in ex.1 but for abs(psi)^2.

```
figure('Name', 'abs(psi)^2 (Analythical solve)')
hold on
for n = 1:3
    for x = 1:L
        for y = 1:L
            Psin(x, y) = (2 / L) * (sin(pi * n * x / L)) * (sin(pi * n * y /
 L));
        end
    end
    nexttile
    mesh(1:L, 1:L, abs(Psin).^2)
    title(strcat('|psi|^2 (Analythical solve), n= ', num2str(n)))
end
hold off
```

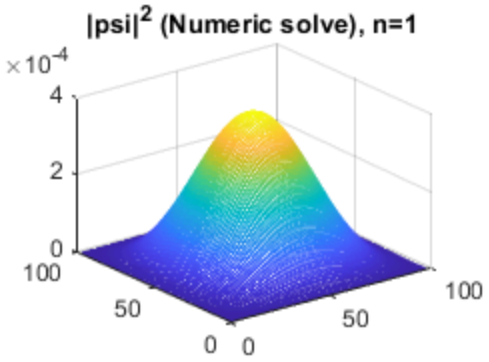


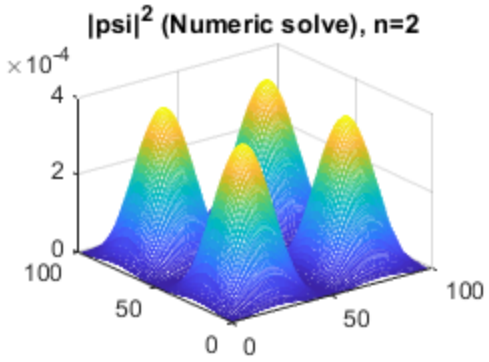


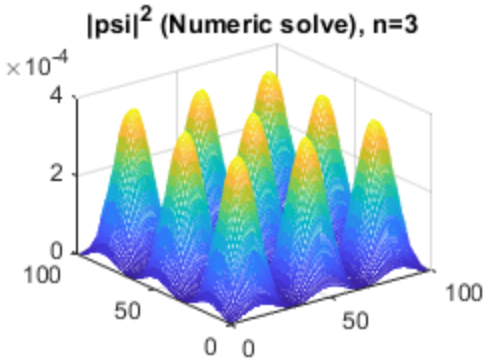


Ex.2 abs(psi)^2 (Numeric solve)

```
figure('Name', 'abs(psi)^2 (Numeric solve)')
for n = 1:3
    vals = psi(:, n);
    for x = 1:L
        for y = 1:L
            psi_r(x, y) = abs(vals(x) * vals(y))^2;
        end
    end
    nexttile
    mesh(1:L, 1:L, psi_r)
    title(strcat('|psi|^2 (Numeric solve), n= ', num2str(n)))
end
```



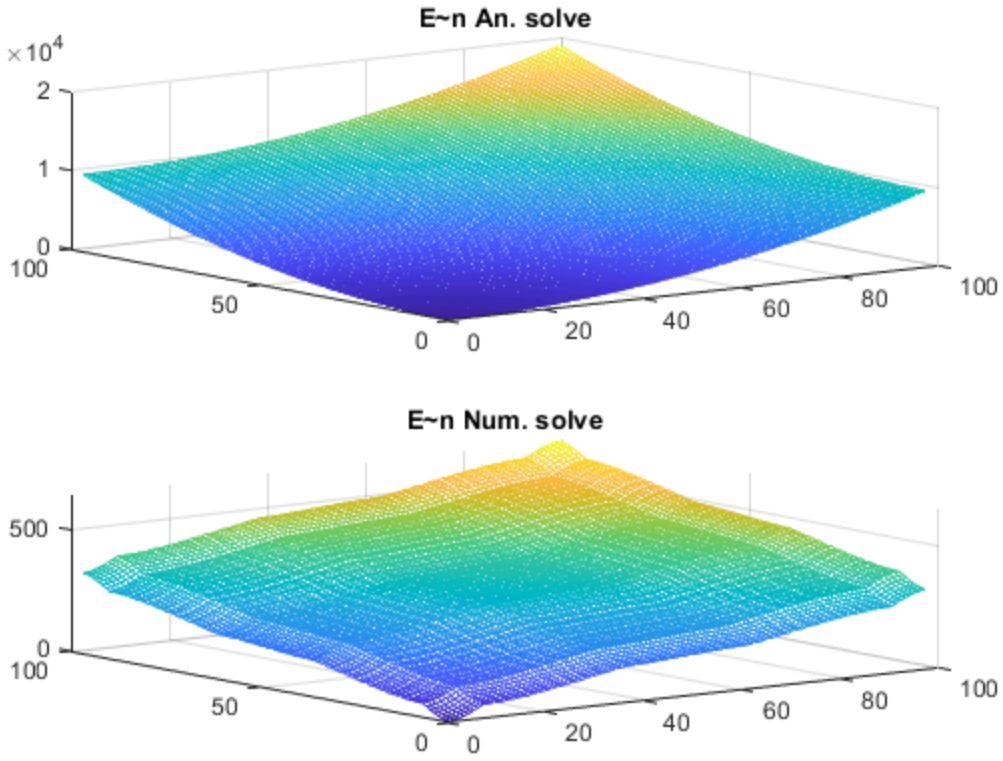




Ex.3 E~n An. and Num. solves

In ex.3 we need to compare analythical and numeric connection between E and n by plotting. Also we need to try high n values (n = 10, 50, 90).

```
figure('Name', 'E~n Num. and An. solves')
E = hbar^2 * diag(D) / (h^2 * 2 * m0);
for nx = 1:98
    for ny = 1:98
        Enum(nx, ny) = E(nx)/E(1) + E(ny)/E(1);
        Ean(nx, ny) = Energy(nx, 98, 0.07)/Energy(1, 98, 0.07) + ...
            Energy(ny, 98, 0.07)/Energy(1, 98, 0.07);
    end
end
hold on
nexttile
mesh(1:98, 1:98, Ean)
title('E~n An. solve')
nexttile
mesh(1:98, 1:98, Enum)
title('E~n Num. solve')
hold off
```



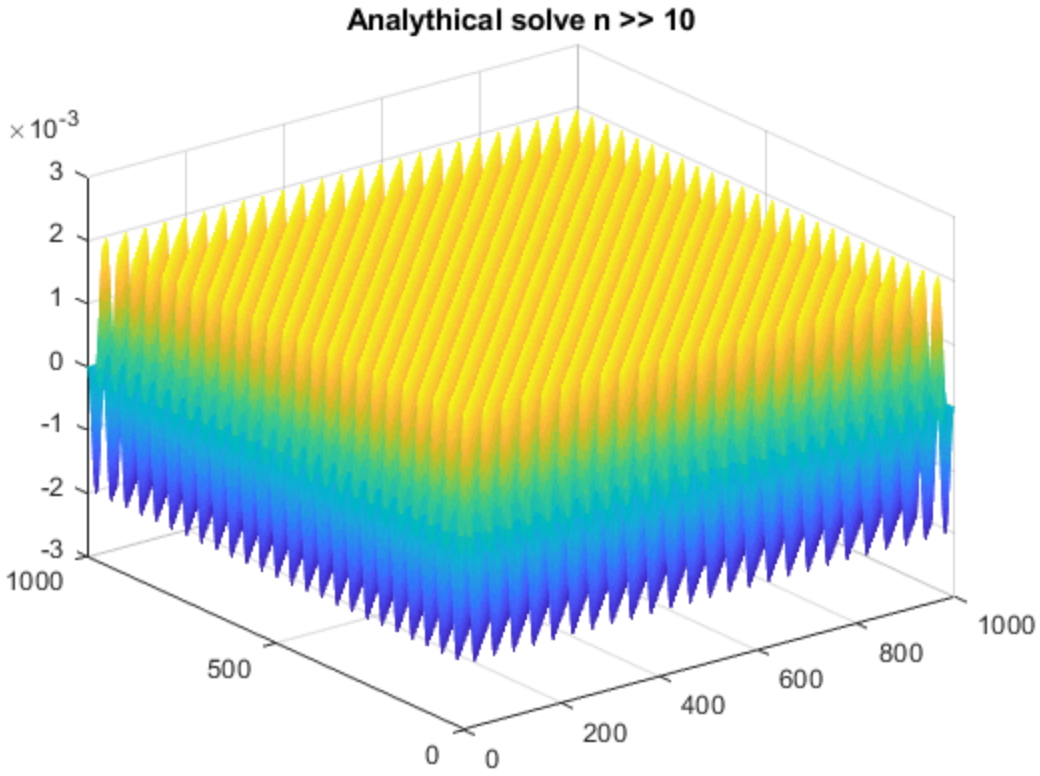
Ex. 4 An. and Num. solves for n >> 10

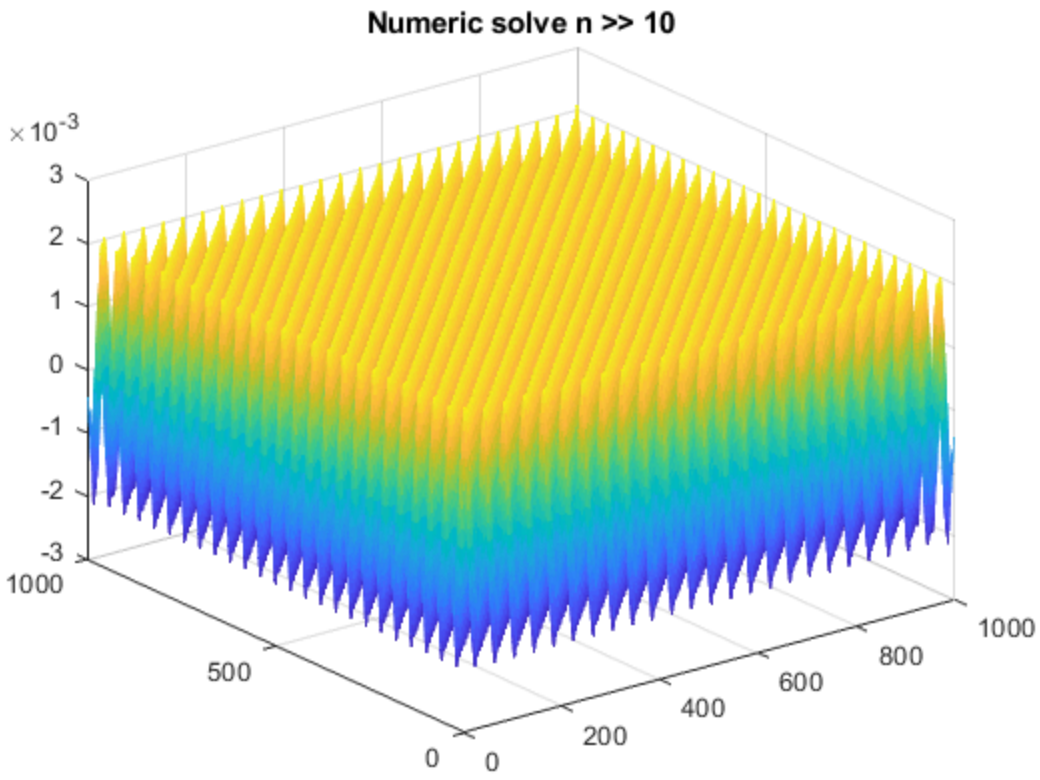
In ex.4 we need the same thing we had in ex.1 but for n >> 10 (for example, n = 50).

```
figure('Name', 'Analythical solve n >> 10')
L = 998;
h = 1;
n = 50;
for x = 1:L
    for y = 1:L
        Psin(x, y) = (2 / L) * (sin(pi * n * x / L)) * (sin(pi * n * y / L));
    end
end
mesh(1:L, 1:L, Psin)
title('Analythical solve n >> 10')

figure('Name', 'Numeric solve n >> 10')
N = 998;
I = eye(N, N);
T = I;
for i = 1:N
    for j = 1:N
        if i == j
            T(i, j) = 4;
            continue
        end
```

```
        if i == j - 1 || i == j + 1 || i == j - 5 || i == j + 5
            T(i, j) = -1;
        end
    end
end
[psi, D] = eig(T);
psi = psi ./ sqrt(h);
vals = psi(:, n);
    for x = 1:N
        for y = 1:N
            psi_r(x, y) = vals(x) * vals(y);
        end
    end
    mesh(1:N, 1:N, psi_r)
title('Numeric solve n >> 10')
```





Ex. 5 (Advanced) En ~ n by numeric solve

In ex.5 we need to confirm an equality of energies on levels like (21, 12), (13, 31), (23, 32).

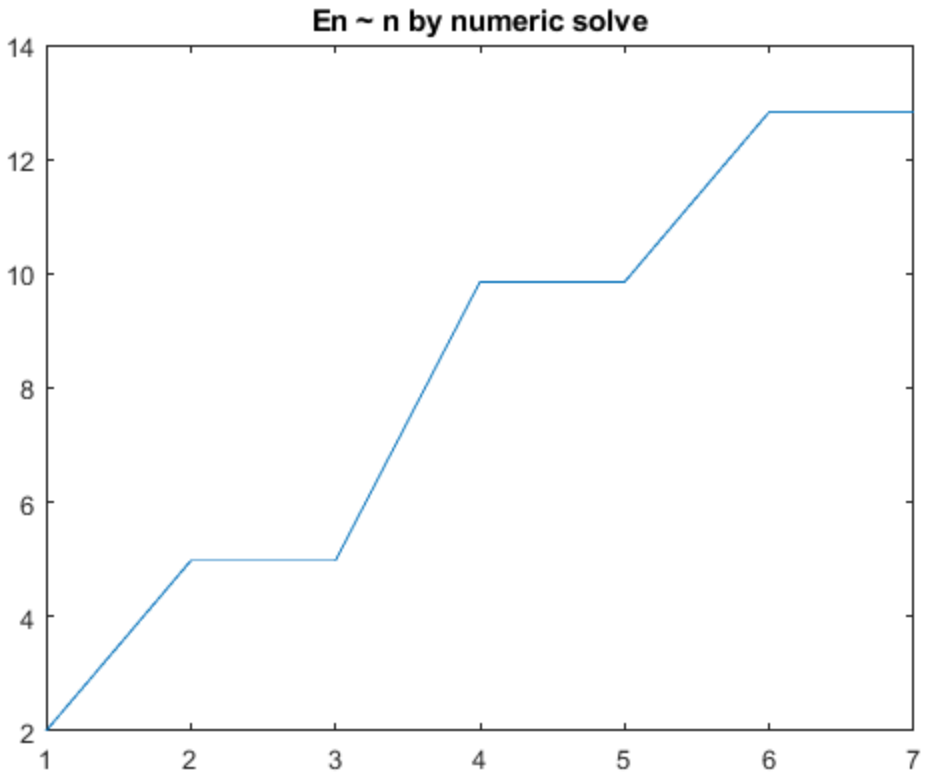
```
figure('Name', 'En ~ n by numeric solve')
En = [Enum(1, 1), Enum(2, 1), Enum(1, 2), Enum(1, 3), ...
    Enum(3, 1), Enum(2, 3), Enum(3, 2),];
plot(1:7, En)
title('En ~ n by numeric solve')

datetime(clock)


ans =

  datetime

   09-Nov-2023 23:58:46
```



Published with MATLAB® R2023a