TESTING OF ALGORITHM 1

→ To test Algorithm 1 of ten microphones using single finger snap paper, it takes $(7!)^{10}$ iterations according to the code, $\begin{bmatrix} 7 \rightarrow \text{no. of peaks per mic} \\ 10 \rightarrow \text{no. of mics} \end{bmatrix}$ as we use permutation of peaks for every mic to get right combination of θ/t_{ij}; which is very huge.

→ So, we designed a room configuration with a source & 10 mics & 6 walls in such a way that all mics has same order of peaks, which in turn will help us detect the right combination of θ in first iteration itself. This code is run for only 10 iterations, & we get result in 1st iteration itself. (due to permutations function in MATLAB)

⇓

I/P :- 2 4 6

O/P :-

6	6	4	4	2	2
4	2	6	2	6	4
2	4	2	6	4	6

⇓

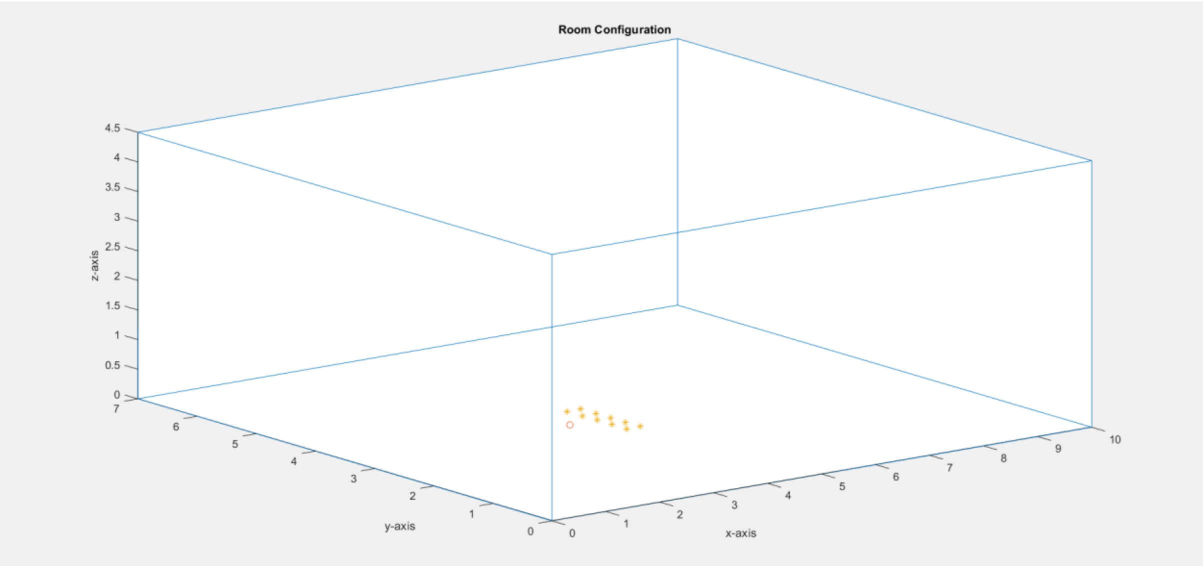
first combination (descending order).

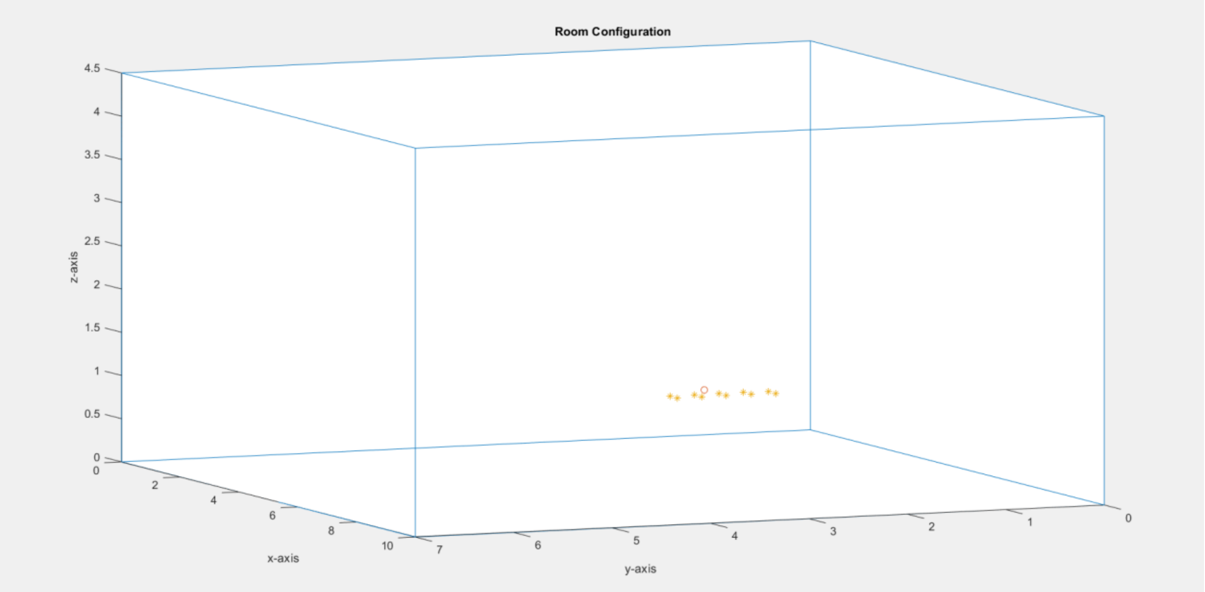
Room Configuration :

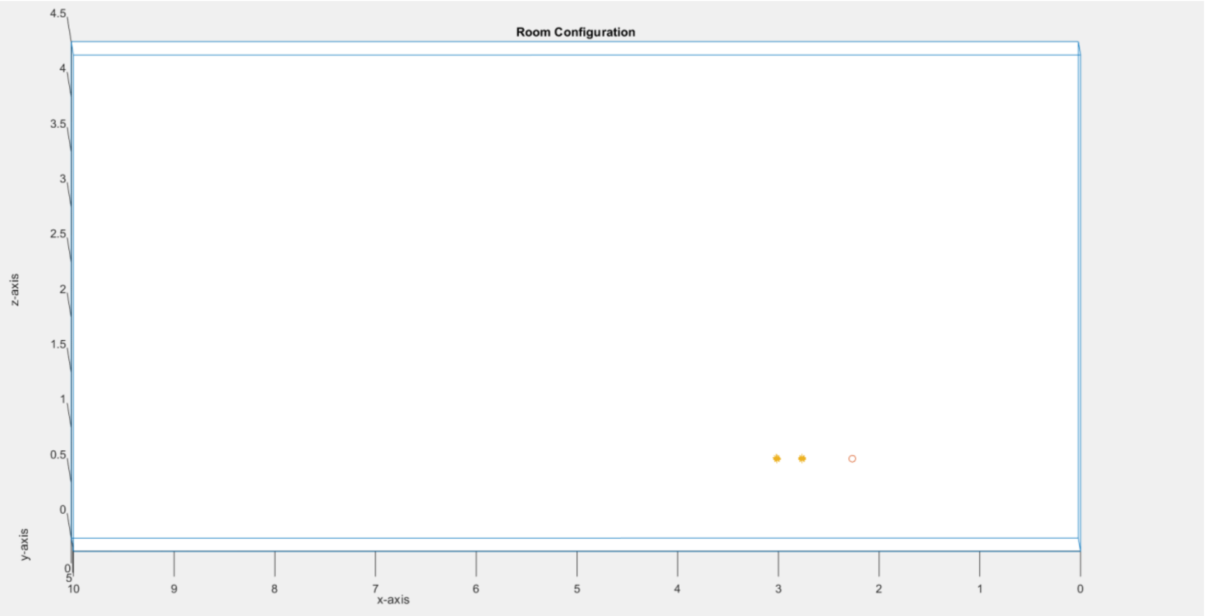
Length, Breadth, Height, source position, mics positions

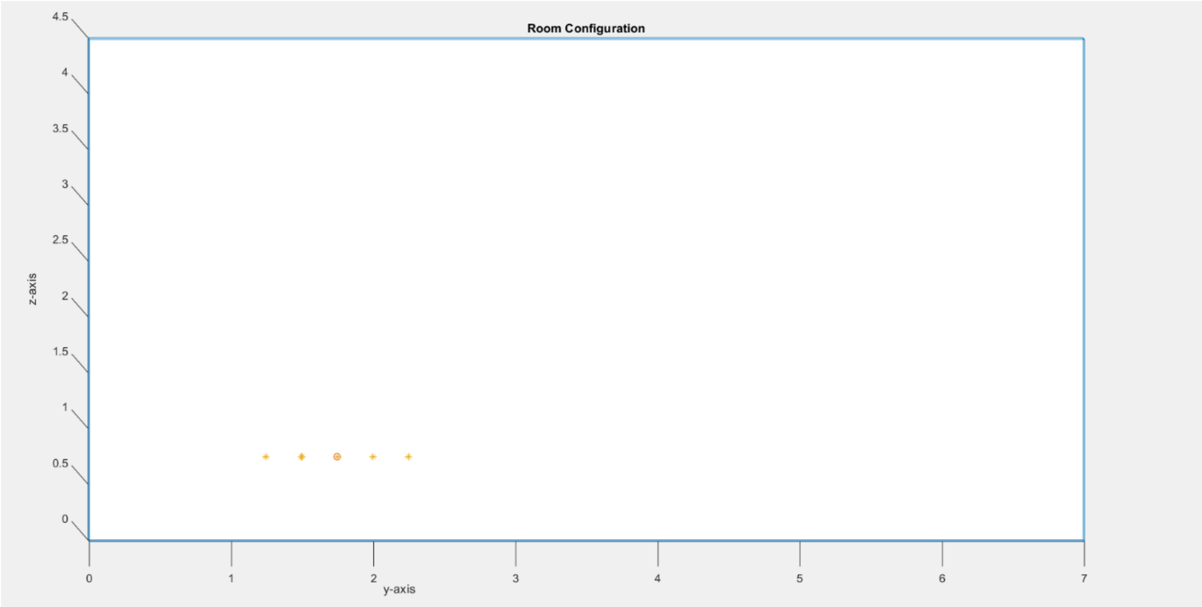
```
>> dim = [10 7 4.5]

dim =

   10.0000    7.0000    4.5000

>> s = [2.25 1.75 0.75]

s =

    2.2500    1.7500    0.7500

>> M = [2.75 1.25 0.75; 2.75 1.5 0.75; 2.75 1.75 0.75; 2.75 2 0.75; 2.75 2.25 0.75; 3 1.25 0.75; 3 1.5 0.75; 3 1.75 0.75; 3 2 0.75; 3 2.25 0.75]

M =

    2.7500    1.2500    0.7500
    2.7500    1.5000    0.7500
    2.7500    1.7500    0.7500
    2.7500    2.0000    0.7500
    2.7500    2.2500    0.7500
    3.0000    1.2500    0.7500
    3.0000    1.5000    0.7500
    3.0000    1.7500    0.7500
    3.0000    2.0000    0.7500
    3.0000    2.2500    0.7500
```

Different views of Room :




Room Configuration
z-axis
x-axis
y-axis


Room Configuration
z-axis
y-axis
x-axis



<u>Expected order of peaks</u> :

Direct Peak, wall 6, wall 1, wall 2, wall 5, wall 3, wall 4.

Where,

Wall 1 – wall on x-axis

Wall 2 – wall on y-axis

Wall 3 – parallel to wall on x-axis

Wall 4 – parallel to wall on x-axis

Wall 5 – top wall

Wall 6 – bottom wall .

Basic theory:-

→ We calculate a case of source & Mic & a wall considered at once.

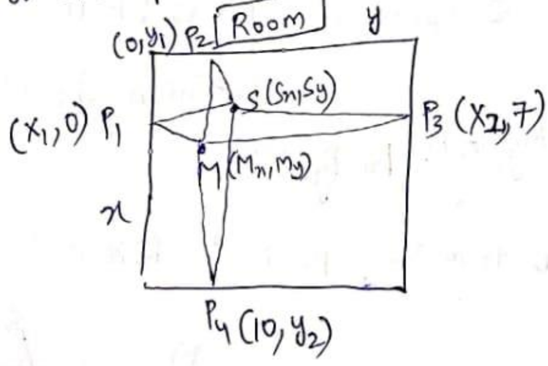
→ Total calculations :- 1 × 10 × 7 = 70
(Source) (Mics) (Direct peak + 6 walls)

→ Direct peak to mic distance can be easily calculated by distance formula.

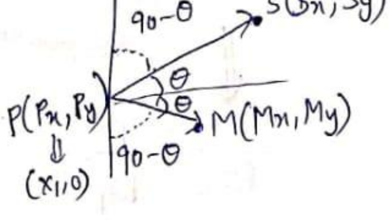
$$d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2 + (z_2-z_1)^2}$$

→ In case of side walls (1, 2, 3, 4),

- z-coordinate is same for Source, Mic, point of contact
(S) (M) (P).
- So, we only deal with x & y coordinates.
- For example



⇓ For 1 wall



slope of PS = $\tan\theta$.

slope of PM = $\tan(360-\theta)$
$= -\tan\theta$.

$\Rightarrow \text{slope}_{PS} = -\text{slope}_{PM}$

$$\text{slope}_{PS} = \frac{S_y - P_y}{S_x - P_x} = \frac{S_y - 0}{S_x - x_1} = \frac{S_y}{S_x - x_1}$$

$$\text{slope}_{PM} = \frac{M_y - P_y}{M_x - P_x} = \frac{M_y}{M_x - x_1} = \frac{M_y}{M_x - x_1}$$

$$\Rightarrow \frac{S_y}{S_x - x_1} = \frac{-M_y}{M_x - x_1}$$ (where S_x, S_y, M_x, M_y are known)

$\Downarrow$

we get x_1 & thus, $\underline{P}$.

→ same thing applies for walls 1,2,3,4 for our room configuration.

→ In case of bottom & top walls,

- Here both source & Mics are at same height (z), but differs in x & y coordinates.;
- Point of contact P's z-coordinate is 0 (bottom)
  is full height (top).;

Bottom:- $\underbrace{(S_x, S_y, z)\,;\,(M_x, M_y, z)}$; $(P_x, P_y, 0)$

$\Downarrow$

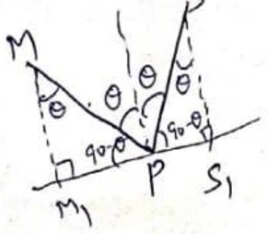
drop normals from these points to bottom surface

$\Downarrow$

$\underbrace{(S_x, S_y, 0)\,;\,(M_x, M_y, 0)\,;\,(P_x, P_y, 0)}$

These points are collinear.

⇒ ~~$M_x = S$~~



$\Downarrow$

$$\frac{PM}{PM_1} = \frac{PS}{PS_1}$$

$\Downarrow$

$$\frac{\sqrt{(M_x-P_x)^2+(M_y-P_y)^2+(z-0)^2}}{\sqrt{(M_x-P_x)^2+(M_y-P_y)^2}} = \frac{\sqrt{(S_x-P_x)^2+(\cancel{M_y}S_y-P_y)^2+(z-0)^2}}{\sqrt{(S_x-P_x)^2+(S_y-P_y)^2}}$$

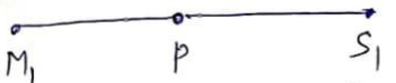
$$\Rightarrow \frac{(M_x-P_x)^2+(M_y-P_y)^2+z^2}{\underbrace{(M_x-P_x)^2+(M_y-P_y)^2}_{a}} = \frac{(S_x-P_x)^2+(S_y-P_y)^2+z^2}{\underbrace{(S_x-P_x)^2+(S_y-P_y)^2}_{b}}$$

$$\Rightarrow \frac{a+z^2}{a} = \frac{b+z^2}{b}$$

$$\cancel{ab}+bz^2 = \cancel{ab}+az^2$$

$$\Rightarrow a=b.$$

$$\Rightarrow PM_1 = PS_1 \Longrightarrow$$



$(M_x, M_y, 0)$ $(P_x, P_y, 0)$ $(S_x, S_y, 0)$.

$$\Rightarrow PM = PS$$

$$\Rightarrow P_x = \frac{M_x+S_x}{2}$$

(mid point)

$$P_y = \frac{M_y+S_y}{2}$$

Top:- same procedure
i.e, mid point for x & y;
full height for z. co-ordinate.

→ For our configuration, formula to calculate points of contact on walls (1, 2, 3, 4, 5, 6).

- **Wall 1:-**

$$P = \left(\frac{S_y M_x + M_y S_x}{S_y + M_y},\ 0,\ 0.75 \right)$$

- **Wall 2:-**

$$P = \left(0,\ \frac{S_y M_x + M_y S_x}{S_x + M_x},\ 0.75 \right)$$

- **Wall 3:-**

$$P = \left(\frac{M_x (S_y - 7) + S_x (M_y - 7)}{(S_y - 7) + (M_y - 7)},\ 7,\ 0.75 \right)$$

- **Wall 4:-**

$$P = \left(10,\ \frac{S_y (M_x - 10) + M_y (S_x - 10)}{(M_x - 10) + (S_x - 10)},\ 0.75 \right)$$

- **Wall 5:-**

$$P = \left(\frac{M_x + S_x}{2},\ \frac{M_y + S_y}{2},\ 4.5 \right)$$

- **Wall 6:-**

$$P = \left(\frac{M_x + S_x}{2},\ \frac{M_y + S_y}{2},\ 0 \right)$$

→ Now, all the points of contacts are calculated & distances from source to point of contact & point of contact to mics are also calculated.

Direct ~~...~~ peak : d_{S-M}

echoes : $d_{S-P} + d_{P-M}$.

→ P_1 : points of contacts on wall 1 corresponding to 10 Mics.

→ P_2 : " wall 2 " .

→ similarly P_3, P_4, P_5, P_6.

→ Now, we have S, P_s & M_s.

→ So we get distance matrix D $\left(d_{ij_s} \rightarrow \begin{matrix} i \rightarrow \text{peak} \\ j \rightarrow \text{mic} \end{matrix}\right)$

→ $\frac{D}{V} \Rightarrow t_{ij}$ $(v = 340\,m/s)$.

→ This t_{ij} is taken as input to algorithm 1 testing code, & we got the result in 1st iteration, as expected (as all the mics has same order of peaks).

→ The ε_{best} is ~~...~~ 1.7×10^{-16}, which gives us the correct choice of $\hat{R}, \hat{S}, \hat{\tau}$.

<u>Note</u>:- H_Q is real, only for the right t_{ij} combination.

Details of Algorithm 1 – testing code:-

Name:- ten_mic_algo1_sumanth_testing

No of iterations:- 10.

Result occurs at:- iteration 1.

→ The following are the simulation results.


```
>> dim = [10 7 4.5]

dim =

   10.0000    7.0000    4.5000

>> s = [2.25 1.75 0.75]

s =

    2.2500    1.7500    0.7500

>> M = [2.75 1.25 0.75; 2.75 1.5 0.75; 2.75 1.75 0.75; 2.75 2 0.75; 2.75 2.25 0.75; 3 1.25 0.75; 3 1.5 0.75; 3 1.75 0.75; 3 2 0.75; 3 2.25 0.75]

M =

    2.7500    1.2500    0.7500
    2.7500    1.5000    0.7500
    2.7500    1.7500    0.7500
    2.7500    2.0000    0.7500
    2.7500    2.2500    0.7500
    3.0000    1.2500    0.7500
    3.0000    1.5000    0.7500
    3.0000    1.7500    0.7500
    3.0000    2.0000    0.7500
    3.0000    2.2500    0.7500
```

```
>> [P1, P2, P3, P4, P5, P6, D] = peak_order_dist_1(dim, s, M)

P1 =

    2.5417         0    0.7500
    2.5192         0    0.7500
    2.5000         0    0.7500
    2.4833         0    0.7500
    2.4688         0    0.7500
    2.6875         0    0.7500
    2.6538         0    0.7500
    2.6250         0    0.7500
    2.6000         0    0.7500
    2.5781         0    0.7500


P2 =

         0    1.5250    0.7500
         0    1.6375    0.7500
         0    1.7500    0.7500
         0    1.8625    0.7500
         0    1.9750    0.7500
         0    1.5357    0.7500
         0    1.6429    0.7500
         0    1.7500    0.7500
         0    1.8571    0.7500
         0    1.9643    0.7500
```

```
P3 =

    2.4886    7.0000    0.7500
    2.4942    7.0000    0.7500
    2.5000    7.0000    0.7500
    2.5061    7.0000    0.7500
    2.5125    7.0000    0.7500
    2.6080    7.0000    0.7500
    2.6163    7.0000    0.7500
    2.6250    7.0000    0.7500
    2.6341    7.0000    0.7500
    2.6437    7.0000    0.7500


P4 =

   10.0000    1.4917    0.7500
   10.0000    1.6208    0.7500
   10.0000    1.7500    0.7500
   10.0000    1.8792    0.7500
   10.0000    2.0083    0.7500
   10.0000    1.4873    0.7500
   10.0000    1.6186    0.7500
   10.0000    1.7500    0.7500
   10.0000    1.8814    0.7500
   10.0000    2.0127    0.7500


P5 =

    2.5000    1.5000    4.5000
    2.5000    1.6250    4.5000
    2.5000    1.7500    4.5000
    2.5000    1.8750    4.5000
    2.5000    2.0000    4.5000
    2.6250    1.5000    4.5000
    2.6250    1.6250    4.5000
    2.6250    1.7500    4.5000
    2.6250    1.8750    4.5000
    2.6250    2.0000    4.5000


P6 =

    2.5000    1.5000         0
    2.5000    1.6250         0
    2.5000    1.7500         0
    2.5000    1.8750         0
    2.5000    2.0000         0
    2.6250    1.5000         0
    2.6250    1.6250         0
    2.6250    1.7500         0
    2.6250    1.8750         0
    2.6250    2.0000         0


D =

    0.7071    0.5590    0.5000    0.5590    0.7071    0.9014    0.7906    0.7500    0.7906    0.9014
    3.0414    3.2882    3.5355    3.7832    4.0311    3.0923    3.3354    3.5795    3.8243    4.0697
    5.0249    5.0062    5.0000    5.0062    5.0249    5.2738    5.2559    5.2500    5.2559    5.2738
   11.0114   10.7616   10.5119   10.2622   10.0125   11.0255   10.7761   10.5268   10.2774   10.0281
   15.0083   15.0021   15.0000   15.0021   15.0083   14.7585   14.7521   14.7500   14.7521   14.7585
    7.5333    7.5208    7.5166    7.5208    7.5333    7.5540    7.5416    7.5374    7.5416    7.5540
    1.6583    1.6008    1.5811    1.6008    1.6583    1.7500    1.6956    1.6771    1.6956    1.7500

>> peak_time_matrix = (D/340)'

peak_time_matrix =

    0.0021    0.0089    0.0148    0.0324    0.0441    0.0222    0.0049
    0.0016    0.0097    0.0147    0.0317    0.0441    0.0221    0.0047
    0.0015    0.0104    0.0147    0.0309    0.0441    0.0221    0.0047
    0.0016    0.0111    0.0147    0.0302    0.0441    0.0221    0.0047
    0.0021    0.0119    0.0148    0.0294    0.0441    0.0222    0.0049
    0.0027    0.0091    0.0155    0.0324    0.0434    0.0222    0.0051
    0.0023    0.0098    0.0155    0.0317    0.0434    0.0222    0.0050
    0.0022    0.0105    0.0154    0.0310    0.0434    0.0222    0.0049
    0.0023    0.0112    0.0155    0.0302    0.0434    0.0222    0.0050
    0.0027    0.0120    0.0155    0.0295    0.0434    0.0222    0.0051

>> [R_hat, S_hat, T_hat, Thet_best, eps_best, count_best] = ten_mic_algo1_sumanth_testing(peak_time_matrix)
```

```
R_hat =

   -0.4270   -0.1815    0.0640    0.3095    0.5550   -0.5481   -0.3026   -0.0570    0.1885    0.4340
   -0.4048   -0.3576   -0.3104   -0.2632   -0.2160    0.2249    0.2722    0.3194    0.3666    0.4138
         0         0         0         0         0         0         0         0         0         0


S_hat =

    0.0495    0.0495    0.0495   -1.0914   10.3606    0.3807   -3.3875
   -0.2348   -0.2348   -0.2348    5.6989    1.7478   -1.9575   -0.8957
    0.4940    1.5793    7.5163   13.6950    0.4940    4.7103    0.4940


T_hat =

     0     0     0     0     0     0     0
```

```
Thet_best =

    0.0021    0.0016    0.0015    0.0016    0.0021    0.0027    0.0023    0.0022    0.0023    0.0027
    0.0049    0.0047    0.0047    0.0047    0.0049    0.0051    0.0050    0.0049    0.0050    0.0051
    0.0222    0.0221    0.0221    0.0221    0.0222    0.0222    0.0222    0.0222    0.0222    0.0222
    0.0441    0.0441    0.0441    0.0441    0.0441    0.0434    0.0434    0.0434    0.0434    0.0434
    0.0324    0.0317    0.0309    0.0302    0.0294    0.0324    0.0317    0.0310    0.0302    0.0295
    0.0148    0.0147    0.0147    0.0147    0.0148    0.0155    0.0155    0.0154    0.0155    0.0155
    0.0089    0.0097    0.0104    0.0111    0.0119    0.0091    0.0098    0.0105    0.0112    0.0120


eps_best =

   1.7220e-16


count_best =

     1
```