円柱の軸出し精度のpcaとfittingの比較

test_cylinder_fitting.pyの中身

In [1]:
```
# from test_cylinder_fitting import *
import random
random.seed(0)
from geo import *
from face_fit import *

import numpy
import numpy.linalg
```

In [2]:
```
def distort(pos,sigma) :
    return map(lambda x: x+random.normalvariate(0,sigma), pos)

def make_cylinder_point_cloud(n,r,l_range,angle_range,sigma=None,transform=None)
    rslt = []
    for i in range(n) :
        zz=random.uniform(l_range[0],l_range[1])
        th=random.uniform(angle_range[0],angle_range[1])
        if sigma :
            tmp=VECTOR(vec=list(distort([r*cos(th),r*sin(th),zz],sigma)))
        else :
            tmp=VECTOR(r*cos(th),r*sin(th),zz)
        if transform :
            tmp=transform*tmp
        rslt.append(tmp)
    return rslt
```

In [3]:
```
def calc_pca(point_cloud) :
    tmp=numpy.array(point_cloud).T
    tmp=numpy.cov(tmp,bias=True)
    return numpy.linalg.eig(tmp)
```

In [4]:
```
def calc_axis_center_pca(point_cloud) :
    tmp=calc_pca(point_cloud)
    idx=numpy.argmax(tmp[0])
    cc=numpy.average(point_cloud,axis=0)
    return VECTOR(tmp[1][0][idx], tmp[1][1][idx], tmp[1][2][idx]),VECTOR(cc[0],cc
```

In [5]:
```
def make_cylinder_face_data(point_cloud,radius,transform=None) :
    if not transform :
        transform=FRAME()
    return [["cylinder",transform,radius],point_cloud]
```

In [6]:
```
def outlier(p_list, sigma, n=1) :
    rslt=[]
    for pp in p_list :
        for i in range(n) :
            rslt.append( VECTOR(pp[0]+random.normalvariate(0,sigma),
                                pp[1]+random.normalvariate(0,sigma),
```

```
                                    pp[2]+random.normalvariate(0,sigma)))
    return rslt
```

In [7]:
```
def cmp_with_tv(tv, axis, center) :
    tv_axis=tv.mat.col(2)
    tmp1=abs(axis*tv_axis)
    tmp2=(-tv)*center
    return tmp1, tmp2
```

関数の単体テスト

円柱表面上のpoint_cloudの生成

In [8]:
```
a=make_cylinder_point_cloud(100,10,[-100,100],[-pi,pi])
```

共分散行列を生成してpcaを行う.

In [9]:
```
aa=numpy.array(a).T
```

In [10]:
```
b=numpy.cov(aa,bias=True)
```

In [11]:
```
c=numpy.linalg.eig(b)
c
```

Out[11]:
```
(array([3213.27768195,   52.2790474 ,   46.53225288]),
 array([[-0.00378424,  0.9671312 , -0.25424973],
        [ 0.00996903, -0.25420243, -0.96709966],
        [ 0.99994315,  0.00619436,  0.0086794 ]]))
```

第１主成分の軸を取り出す.

In [12]:
```
tmp=numpy.argmax(c[0])
VECTOR(c[1][0][tmp],c[1][1][tmp],c[1][2][tmp])
```

Out[12]:
```
v:[-0.003784241077756072, 0.009969031333209265, 0.9999431473507596]
```

pcaを求める部分をまとめた関数のテスト

In [13]:
```
calc_pca(a)
```

Out[13]:
```
(array([3213.27768195,   52.2790474 ,   46.53225288]),
 array([[-0.00378424,  0.9671312 , -0.25424973],
        [ 0.00996903, -0.25420243, -0.96709966],
        [ 0.99994315,  0.00619436,  0.0086794 ]]))
```

重心を求める

In [14]:
```
numpy.average(a,axis=0)
```

Out[14]:
```
array([ 0.72877932, -0.54565138,  7.18142443])
```

pca を使って軸と重心（軸の通過位置）を求める

In [15]:
```
tmp=calc_axis_center_pca(a)
tmp
```

Out[15]:
```
(v:[-0.003784241077756072, 0.009969031333209265, 0.9999431473507596],
 v:[0.7287793178926456, -0.5456513832318244, 7.181424433499858])
```

In [16]:
```
cmp_with_tv(FRAME(),tmp[0],tmp[1])
```

Out[16]:
```
(0.010663117098535707,
 v:[0.7287793178926456, -0.5456513832318244, 7.181424433499858])
```

face_fit.py用のデータ

In [17]:
```
f=make_cylinder_face_data(a,10)
f
```

Out[17]:

```
[['cylinder',
  f:(m:[[1.0, 0.0, 0.0], [0.0, 1.0, 0.0], [0.0, 0.0, 1.0]],v:[0.0, 0.0, 0.0]),
  10],
 [v:[-0.49958183201467393, 9.987513103526865, 68.88437030500961],
  v:[0.5599628954437964, -9.98430976861827, -15.885683833830996],
  v:[8.268478984060739, -5.624255976584449, 2.254944273721705],
  v:[3.2874436688709583, -9.44418943710895, 56.75971780695451],
  v:[8.65872348439298, 5.002650059800457, -4.680609169528836],
  v:[9.995664276858658, 0.2944412748438794, 81.62257703906704],
  v:[-0.36460807002559614, 9.99335083719531, -43.63243112005924],
  v:[0.03181431242067512, -9.999949392348203, 23.67379933506632],
  v:[-9.941561866942632, 1.0795126890185804, 81.94925119364802],
  v:[-8.169410403740258, 5.7672119481826245, 62.04344719931791],
  v:[1.2638198419622642, 9.919816500674923, -37.970486136133474],
  v:[4.029988352674712, 9.15200491025361, 79.7676575935987],
  v:[-8.064194745925654, -5.913439193885828, -5.571456909457325],
  v:[7.6694891999539045, 6.41708153382754, -13.165632909243257],
  v:[-9.780687592103138, 2.0828226582404277, 82.60221064757965],
  v:[-6.627713270231825, 7.488218533642895, -4.5980446894566],
  v:[-3.389024208051284, 9.408215288631652, -47.90153792160812],
  v:[-9.961105570720434, -0.881121903576536, 9.739860767117861],
  v:[8.046500805041498, -5.937493140582694, 43.94093728079082],
  v:[4.918894817163553, 8.70657646711275, 64.96899542964661],
  v:[9.991859904962594, -0.4034050564890171, -99.77143613711435],
  v:[-0.3824975588565354, -9.992682103292829, 73.52055509855617],
  v:[-6.867024607087639, 7.269386015727385, -34.95912745052199],
  v:[9.113755033860212, 4.115758640006437, -61.78658169952189],
  v:[-9.79274064219856, 2.0253964339438695, -52.27681427695596],
  v:[9.470369270194654, -3.2112467806371994, 60.63589385597402],
  v:[4.260897047676061, -9.046809180540142, -83.91083628949292],
  v:[-9.122644141022635, 4.096018051262153, 1.5881285041147777],
  v:[9.485658954910551, 3.165797560035268, -78.18843081377926],
  v:[9.55902115826474, 2.9368545241204953, 41.312281973377935],
  v:[9.681384693365167, 2.5041546316221424, 62.8933726582672],
  v:[7.9709067930900925, 6.0385962686596475, 92.76770919476019],
  v:[9.408574107569445, -3.388027931167896, 17.52341283508727],
  v:[7.4970017214802995, -6.617776453471469, 19.257372316621257],
  v:[2.5069463435930404, -9.680662169001947, 15.130202832977702],
  v:[-3.871514470133251, -9.220161371014546, -62.121734289128774],
  v:[5.5342963564887695, 8.328959349075678, 22.554635973721332],
  v:[-8.44918734236645, -5.348946929405317, -4.693801598123855],
  v:[-7.149284232639465, 6.991976470278825, 51.52078439328736],
  v:[-5.4881391138020765, 8.359445500005114, 84.67620318925611],
  v:[-8.85472133690529, 4.646924794501645, 79.6346242715758],
  v:[7.756783876879009, -6.311283854129036, 8.119984989610884],
  v:[1.6036846130380187, -9.870572205394431, 41.05667997088125],
  v:[-5.851692474482003, 8.109111861609179, 62.32574170157571],
  v:[8.449966217559783, 5.347716421249205, 79.00779348533504],
  v:[8.772282569771097, 4.800735205787776, 89.95297464642411],
  v:[5.345244081810845, 8.451530376557026, -9.887378673768964],
  v:[-8.668866430757946, 4.9850531397045295, 99.25156787071455],
  v:[-8.69026632985408, -4.947653091740012, 58.66501682604485],
  v:[9.963749220149609, -0.8507064581676032, 22.556621008142443],
  v:[-5.624864485980316, 8.268065040525347, 26.029468082294557],
  v:[1.160446152749524, 9.932440018775287, -51.39287558762875],
  v:[-1.8453814987909967, -9.82825351341325, -76.5731413582964],
  v:[4.9565594987724015, -8.685189573931538, 58.91659434211519],
  v:[-8.067674342626917, -5.908691115748001, 63.18261930673191],
  v:[3.2290287360704544, 9.46432107557807, -70.72830221753924],
  v:[8.942185186527094, 4.4763069700206435, -90.95318642687752],
```

```
 v:[9.77003658061541, 2.132225413373722, 82.00320293980795],
 v:[-9.85964450117521, -1.6695539255877443, 36.1178265124513],
 v:[7.849728750809486, 6.19530132751547, 26.999981982291658],
 v:[7.753346985275101, -6.315505563763323, 15.190589606308151],
```

外れ値生成用の関数のテスト

In [18]:
```
outlier([VECTOR(1,0,0),VECTOR(0,1,0)],1,6)
```

Out[18]:
```
[v:[0.8734432458871026, -1.2430457051469561, -0.04980953110323319],
 v:[0.9864907466924226, -0.2974394211491618, 0.2491099349434455],
 v:[0.4760689588007443, -0.40097026649446166, 1.1738576792081525],
 v:[3.0267252195784047, -0.19291317951565695, -0.7065251276395103],
 v:[-0.3125276022543755, 0.39616207451501667, 0.8840790108339466],
 v:[1.7030658173795512, 1.4309783107514624, 0.926593721570572],
 v:[1.9198097871157425, 0.5264434923857274, -1.213931979243707],
 v:[-1.3584790755077172, 1.5664453481982865, 0.21567282882685382],
 v:[-0.5965976245359932, 1.1177143806219447, -0.04949489717599043],
 v:[0.042559178734451246, 0.752611851569589, -0.16244448470877107],
 v:[1.332831905209186, 0.485096771873879, 0.5970921641251684],
 v:[-1.2812138556969408, 1.3482055149702687, 0.4565809236978218]]
```

ガウス・ニュートン法の1回目

In [19]:
```
g=fit_face1(FRAME(xyzabc=[5,5,5,pi/6,pi/6,pi/6]),[f])
g
```

Out[19]:
```
(f:(m:[[0.7513591464411092, -0.6559914677357819, 0.07165631386795007], [0.651039
4389103459, 0.7546229311866306, 0.08180391623021199], [-0.10773616868345343, -0.
014813034299298256, 0.9940691585459527]],v:[3.61121123518089, 3.240957838154604
3, 4.910275710888731]),
 [-2.1729074363399317,
  -0.5563158110216354,
  -1.462560971049779e-15,
  -0.7286550688841125,
  -0.017384082812397485,
  3.903351958143094e-16])
```

２回目以降は以下を繰り返せば良い．

In [20]:
```
g=fit_face1(g[0],[f])
g
```

Out[20]:
```
(f:(m:[[0.7541735528239153, -0.6563321671183001, 0.02122118344363754], [0.655117
5590212064, 0.7542162063872316, 0.0444848051018374], [-0.04520216900823199, -0.0
1964689361319511, 0.9987846431980736]],v:[0.7291406506927856, 2.225258714871235,
5.201610290478198]),
 [-2.85811755307812,
  1.119828314100657,
  8.334222147277742e-16,
  -0.004862894570060115,
  -0.06274029109472604,
  -5.686591507849017e-16])
```

上記結果から軸と原点をだす.

In [21]:
```
print(g[0].mat.col(2))
print(g[0].vec)
```

```
v:[0.02122118344363754, 0.0444848051018374, 0.9987846431980736]
v:[0.7291406506927856, 2.225258714871235, 5.201610290478198]
```

In [22]:
```
cmp_with_tv(FRAME(),g[0].mat.col(2),g[0].vec)
```

Out[22]:
```
(0.04928728549734683,
 v:[0.7291406506927856, 2.225258714871235, 5.201610290478198])
```

比較

比較のための準備

In [23]:
```
x=numpy.array([100,200,500,1000,2000,5000,10000,20000,50000,100000])
n_x=len(x)
pca_acc=numpy.zeros(n_x,dtype=numpy.float64)
fit_acc=numpy.zeros(n_x,dtype=numpy.float64)
pca_center=numpy.zeros(n_x,dtype=numpy.float64)
fit_center=numpy.zeros(n_x,dtype=numpy.float64)
```

In [24]:
```
import time
pca_tm=numpy.zeros(n_x,dtype=numpy.float64)
fit_tm=numpy.zeros(n_x,dtype=numpy.float64)
```

In [25]:
```
import matplotlib.pyplot as plt

def plot_data(pca,fit,y="y", log=False) :
    plt.plot(x,pca)
    plt.plot(x,fit)
    plt.xlabel("n")
    plt.ylabel(y)
    plt.xscale('log')

    if log :
        plt.yscale('log')

    plt.legend(["pca","fit"])
    plt.show()
```

点群生成パラメタ

In [26]:
```
tv=FRAME(xyzabc=[50,50,50,pi/3,pi/3,pi/3])
tv_z=tv.mat.col(2)
t_offset=FRAME(xyzabc=[5,5,5,pi/6,pi/6,pi/6])  # for gauss-newton
```

In [27]:
```
def do_pca_fit() :
    s_tm=time.time()
    tmp1=calc_axis_center_pca(cpc)
```

```
    tmp2=cmp_with_tv(tv,tmp1[0],tmp1[1])
    pca_acc[i]=tmp2[0]
    ce=tmp2[1]
    ce[2]=0
    pca_center[i]=abs(ce)
    e_tm=time.time()
    pca_tm[i]=e_tm-s_tm

    print("time=", pca_tm[i])
    print(pca_acc[i])

    fit_data=make_cylinder_face_data(cpc,r)
    fit_init=tv*t_offset

    s_tm=time.time()
    fit=fit_face1(fit_init,[fit_data])
    tmp3=cmp_with_tv(tv,fit[0].mat.col(2),fit[0].vec)
    print("fit 0",tmp3[0])
    for j in range(5) :
        fit=fit_face1(fit[0],[fit_data])
        tmp3=cmp_with_tv(tv,fit[0].mat.col(2),fit[0].vec)
        print("fit",j+1,tmp3[0])
    tmp3=cmp_with_tv(tv,fit[0].mat.col(2),fit[0].vec)
    fit_acc[i]=tmp3[0]
    ce=tmp3[1]
    ce[2]=0
    fit_center[i]=abs(ce)
    e_tm=time.time()
    fit_tm[i]=e_tm-s_tm
    print("time=",fit_tm[i])
```

直径の10倍の長さの円柱，誤差なし

In [28]:

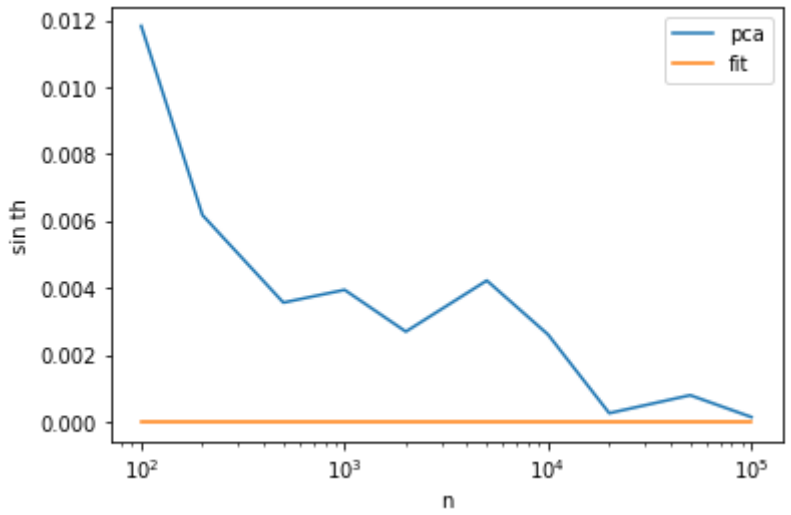
```
r=10
l_range=[-100,100]
angle_range=[-pi,pi]

for i in range(n_x) :
    cpc=make_cylinder_point_cloud(x[i],r,l_range,angle_range,transform=tv)
    print(len(cpc))

    do_pca_fit()
    '''
    s_tm=time.time()
    tmp1=calc_axis_center_pca(cpc)
    tmp2=cmp_with_tv(tv,tmp1[0],tmp1[1])
    pca_acc[i]=tmp2[0]
    ce=tmp2[1]
    ce[2]=0
    pca_center[i]=abs(ce)
    e_tm=time.time()
    pca_tm[i]=e_tm-s_tm

    print("time=", pca_tm[i])
    print(pca_acc[i])

    fit_data=make_cylinder_face_data(cpc,r)
    fit_init=tv*t_offset
```

```
    s_tm=time.time()
    fit=fit_face1(fit_init,[fit_data])
    tmp3=cmp_with_tv(tv,fit[0].mat.col(2),fit[0].vec)
    print("fit 0",tmp3[0])
    for j in range(5) :
        fit=fit_face1(fit[0],[fit_data])
        tmp3=cmp_with_tv(tv,fit[0].mat.col(2),fit[0].vec)
        print("fit",j+1,tmp3[0])
    tmp3=cmp_with_tv(tv,fit[0].mat.col(2),fit[0].vec)
    fit_acc[i]=tmp3[0]
    ce=tmp3[1]
    ce[2]=0
    fit_center[i]=abs(ce)
    e_tm=time.time()
    fit_tm[i]=e_tm-s_tm
    print("time=",fit_tm[i])
    '''
print("finish")
```

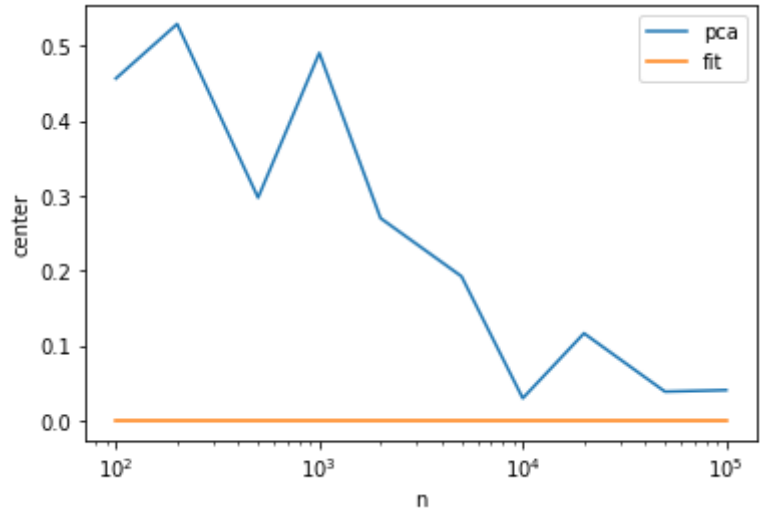
```
100
time= 0.001741170883178711
0.011824909624401777
fit 0 0.02735110602550341
fit 1 0.001706928967397715
fit 2 7.581762981890551e-07
fit 3 2.4720398937982376e-12
fit 4 1.1443916996305594e-16
fit 5 1.3597399555105182e-16
time= 0.07078862190246582
200
time= 0.0030083656311035156
0.006178819999900568
fit 0 0.050753110602416183
fit 1 0.0020508299679575866
fit 2 2.235375415091637e-06
fit 3 2.7544197387704025e-12
fit 4 2.391654643720554e-16
fit 5 2.391654643720554e-16
time= 0.09361815452575684
500
time= 0.002730131149291992
0.0035656394940263526
fit 0 0.044175614207152915
fit 1 0.003538637236783773
fit 2 1.2303851052927181e-05
fit 3 3.333078155120719e-11
fit 4 1.1443916996305594e-16
fit 5 1.1443916996305594e-16
time= 0.1940310001373291
1000
time= 0.005160093307495117
0.003945646306628271
fit 0 0.04180061236680305
fit 1 0.0015342973455288115
fit 2 1.1665402302762255e-06
fit 3 2.577738582666528e-13
fit 4 1.1443916996305594e-16
fit 5 1.1443916996305594e-16
time= 0.36090898513793945
2000
```

```
time= 0.010445117950439453
0.002699066961391213
fit 0 0.03122261650260033 8
fit 1 0.0006116969460386922
fit 2 2.0655396801447058e-07
fit 3 1.781424219857094e-14
fit 4 2.0062885096040707e-16
fit 5 1.8619006149354548e-16
time= 0.9533472061157227
5000
time= 0.03861188888549805
0.004224345984030205
fit 0 0.020880916591783153
fit 1 0.0007158709266112732
fit 2 1.1752579617879717e-07
fit 3 2.2206195148221764e-15
fit 4 1.760893866893587e-16
fit 5 1.760893866893587e-16
time= 2.307370662689209
10000
time= 0.057706356048583984
0.0026120925762041745
fit 0 0.03618173001523795
fit 1 0.0010154210124108968
fit 2 1.2933087838655152e-07
fit 3 2.1257944447923956e-15
fit 4 1.3092278833360675e-16
fit 5 1.3092278833360675e-16
time= 3.7816176414489746
20000
time= 0.16256046295166016
0.0002644940572096145
fit 0 0.02786655507451266
fit 1 0.000663392624482563
fit 2 5.467522116526223e-08
fit 3 3.1031676915590914e-16
fit 4 1.594436429147036e-16
fit 5 1.5515838457795457e-16
time= 7.771425008773804
50000
time= 0.3402845859527588
0.0008003148231100432
fit 0 0.02525034829980899
fit 1 0.0004875302287846279
fit 2 2.1386622016721155e-08
fit 3 1.0385185452638061e-16
fit 4 1.0385185452638061e-16
fit 5 1.0385185452638061e-16
time= 21.42798900604248
100000
time= 0.8876256942749023
0.00014711772479501125
fit 0 0.027182187233109542
fit 1 0.0007035430738131914
fit 2 2.496166956138084e-08
fit 3 2.3055512673781017e-16
fit 4 2.5626980138831173e-16
fit 5 2.5626980138831173e-16
time= 36.99507188796997
finish
```

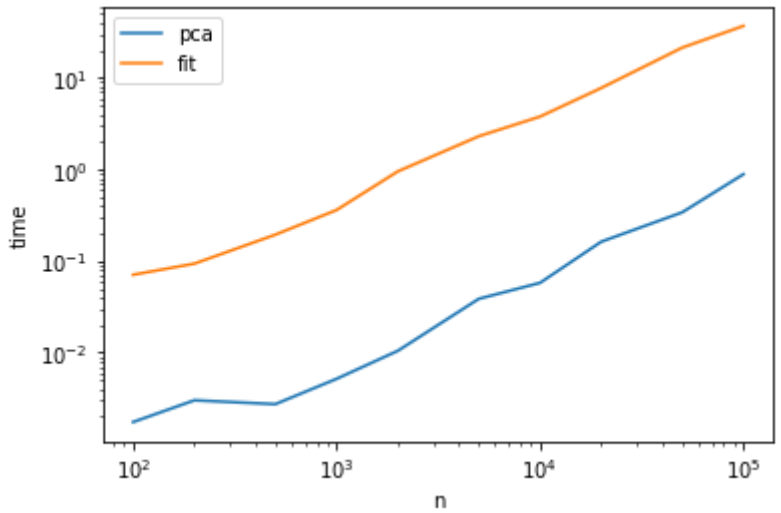
In [29]:
```
plot_data(pca_acc,fit_acc,y="sin th")
```



In [30]:
```
plot_data(pca_center,fit_center,y="center")
```



In [31]:
```
plot_data(pca_tm,fit_tm,y="time",log=True)
```



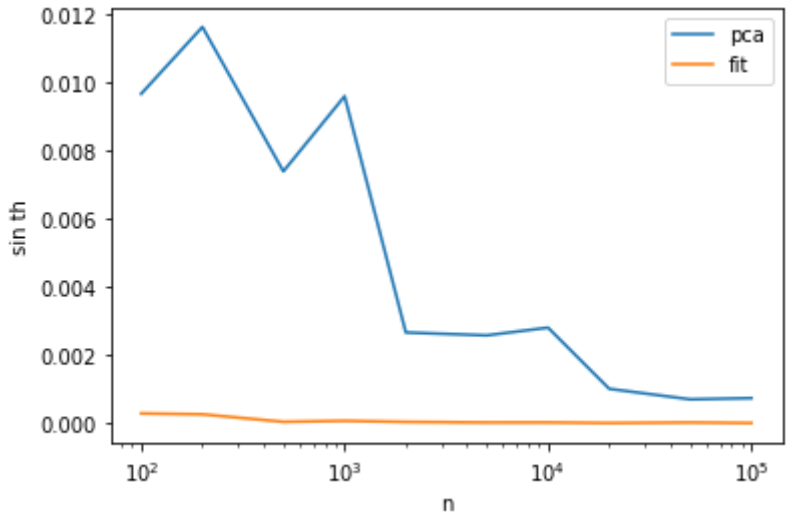
直径の10倍の長さの円柱，分散0.1の誤差を付加

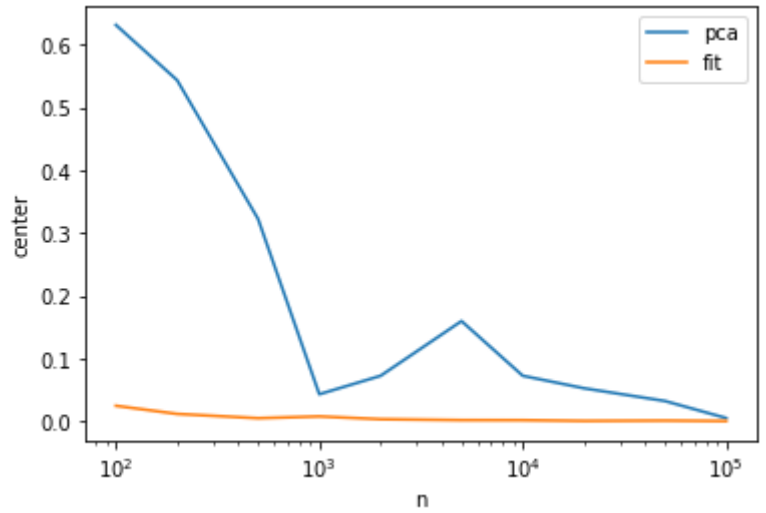
In [32]:
```
r=10
l_range=[-100,100]
angle_range=[-pi,pi]

for i in range(n_x) :
    cpc=make_cylinder_point_cloud(x[i],r,l_range,angle_range,sigma=0.1,transform=
    print(len(cpc))

    do_pca_fit()

print("finish")
```

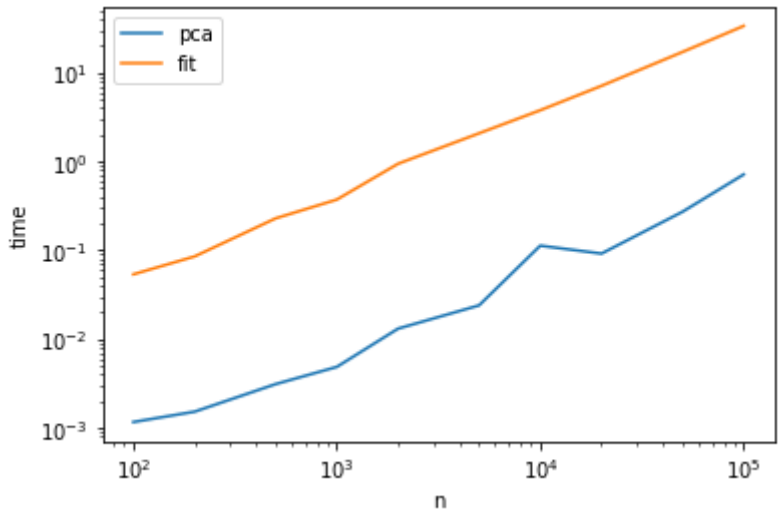
```
100
time= 0.0011696815490722656
0.009679178834631222
fit 0 0.022044690184994106
fit 1 0.0009261670743029337
fit 2 0.0002996688355614716
fit 3 0.0002968163158404396
fit 4 0.0002968335276194433
fit 5 0.0002968334034646771
time= 0.05369877815246582
200
time= 0.001529693603515625
0.011641265359057419
fit 0 0.061846087598874305
fit 1 0.010520999966495144
fit 2 0.0003483314035029418
fit 3 0.00027089239892401677
fit 4 0.000270944749524654
fit 5 0.000270944733407652
time= 0.08533453941345215
500
time= 0.003122091293334961
0.007402478847186788
fit 0 0.03148052320445422
fit 1 0.0005025079727529923
fit 2 5.2225770301705453e-05
fit 3 5.162698652716984e-05
fit 4 5.1627567260639644e-05
fit 5 5.1627566447253556e-05
time= 0.22875499725341797
1000
time= 0.004877567291259766
0.009607919707797358
fit 0 0.021444792613258403
fit 1 0.00085106893741678
fit 2 7.859221870511619e-05
fit 3 7.962393121772227e-05
fit 4 7.962337473303721e-05
fit 5 7.962337567978462e-05
time= 0.37218236923217773
2000
time= 0.013144254684448242
0.0026726000464605292
fit 0 0.031192655455152204
fit 1 0.0007432751751005842
fit 2 4.880434155609996e-05
fit 3 4.8188686525580213e-05
```

```
fit 4 4.818845275100015e-05
fit 5 4.8188452503680725e-05
time= 0.9416887760162354
5000
time= 0.024003267288208008
0.0025885892897210792
fit 0 0.029592765369617826
fit 1 0.00039996421243370213
fit 2 3.1791623239027424e-05
fit 3 3.1643603559175405e-05
fit 4 3.164352242115011e-05
fit 5 3.1643522367630334e-05
time= 2.070983409881592
10000
time= 0.11277484893798828
0.0028151038687437967
fit 0 0.03418614158785972
fit 1 0.0011497406909878004
fit 2 3.2614901685320325e-05
fit 3 3.2361531928165615e-05
fit 4 3.2361583183481e-05
fit 5 3.236158317093375e-05
time= 3.765211582183838
20000
time= 0.09184885025024414
0.0010192688853073646
fit 0 0.030253807710810866
fit 1 0.0009795203956619748
fit 2 1.5834450708736596e-05
fit 3 1.5573764660227737e-05
fit 4 1.5573706730313948e-05
fit 5 1.5573706718045353e-05
time= 7.08450722694397
50000
time= 0.271284818649292
0.0007107099492451402
fit 0 0.02397861265491761
fit 1 0.0004448342544771041
fit 2 3.111653712124608e-05
fit 3 3.10486236194055e-05
fit 4 3.10486165742151e-05
fit 5 3.104861657326069e-05
time= 17.0294930934906
100000
time= 0.7132124900817871
0.0007453643453318061
fit 0 0.02648867810717761
fit 1 0.0004828893049721514 5
fit 2 1.5504385198208365e-05
fit 3 1.5545503634635835e-05
fit 4 1.554550889607195e-05
fit 5 1.5545508896666897e-05
time= 33.41918158531189
finish
```

In [33]:
```
plot_data(pca_acc,fit_acc,y="sin th")
plot_data(pca_center,fit_center,y="center")
plot_data(pca_tm,fit_tm,y="time",log=True)
```







直径の3倍の長さの円柱．分散0.1の誤差を付加

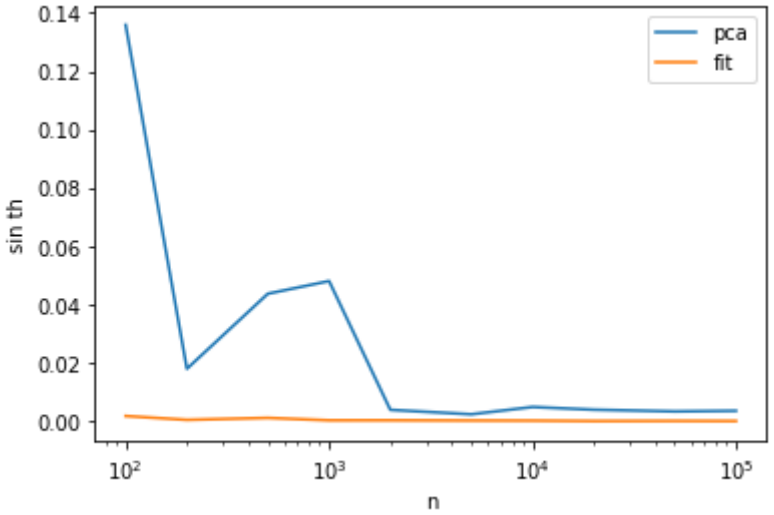
In [34]:
```
r=10
l_range=[-30,30]
```

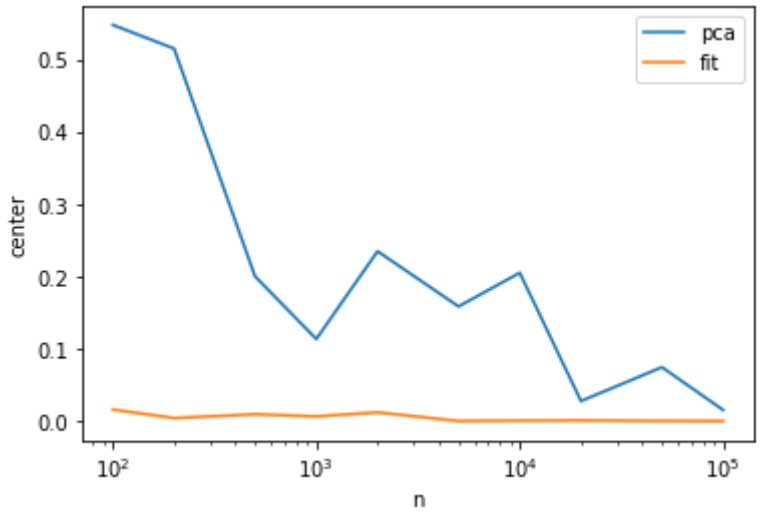
```
angle_range=[-pi,pi]

for i in range(n_x) :
    cpc=make_cylinder_point_cloud(x[i],r,l_range,angle_range,sigma=0.1,transform=
    print(len(cpc))

    do_pca_fit()

print("finish")
```

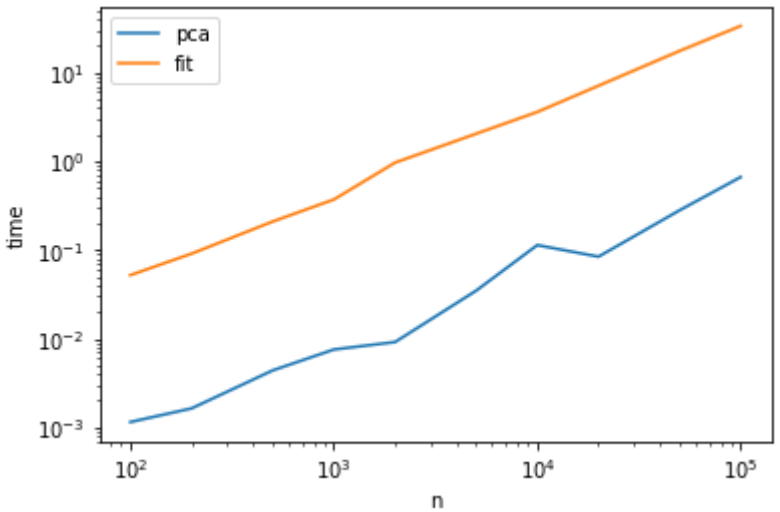
```
100
time= 0.0011515617370605469
0.13589006488002017
fit 0 0.4032018483129849
fit 1 0.17798225928120961
fit 2 0.009634248600482991
fit 3 0.0017152023328407676
fit 4 0.0017322867171929732
fit 5 0.0017322998773046936
time= 0.05208539962768555
200
time= 0.0016410350799560547
0.017945316373912903
fit 0 0.405927212979092
fit 1 0.1397584690538864
fit 2 0.0012402475281827119
fit 3 0.00043176184967588723
fit 4 0.00043115158407868256
fit 5 0.0004311521718835907
time= 0.09092283248901367
500
time= 0.00439143180847168
0.04369023894135549
fit 0 0.41803740309719983
fit 1 0.20519383347092116
fit 2 0.0021592132158241256
fit 3 0.0010876574081009195
fit 4 0.0010875083090365883
fit 5 0.0010875073515568573
time= 0.20894145965576172
1000
time= 0.007554292678833008
0.048042219712400386
fit 0 0.41018942042326584
fit 1 0.19903561909443374
fit 2 0.0010731323454715078
fit 3 0.00025841056023444216
fit 4 0.0002587550663861721
fit 5 0.0002587552261731321
time= 0.37014079093933105
2000
time= 0.009168863296508789
0.0038539752864527694
fit 0 0.39435157258195247
fit 1 0.13806767078590002
fit 2 0.000393921910504481
fit 3 0.00026897665377300864
fit 4 0.00026864932960397067
fit 5 0.00026864933957920236
time= 0.9640109539031982
5000
```

```
time= 0.03470778465270996
0.00233422093027804
fit 0 0.38012971037829735
fit 1 0.11989751639439475
fit 2 9.420423198627837e-05
fit 3 0.00018470113333408838
fit 4 0.00018472621011625224
fit 5 0.00018472622199372773
time= 2.037506580352783
10000
time= 0.11342287063598633
0.004837305510441032
fit 0 0.38689837190992
fit 1 0.13153020332619753
fit 2 0.00034747860536488433
fit 3 0.000152308236681058
fit 4 0.0001521914974436167
fit 5 0.00015219144826572942
time= 3.5865681171417236
20000
time= 0.08403635025024414
0.003902643785466730 6
fit 0 0.38210838303409866
fit 1 0.11574799421674539
fit 2 0.0004548985458780614
fit 3 2.1613387125154735e-05
fit 4 2.1637312812112974e-05
fit 5 2.1637310345626703e-05
time= 7.082995414733887
50000
time= 0.2793619632720947
0.0033232812758601737
fit 0 0.38027279369231815
fit 1 0.1118879718244029
fit 2 0.00017395170744626042
fit 3 5.875736848778042e-05
fit 4 5.875168727525679e-05
fit 5 5.8751685823007374e-05
time= 17.44931125640869
100000
time= 0.6627182960510254
0.0035244785787822033
fit 0 0.37807410943656594
fit 1 0.10911994264523922
fit 2 0.00020804817012921278
fit 3 2.5043487681016868e-05
fit 4 2.502407975228748e-05
fit 5 2.502407580735776e-05
time= 33.40200638771057
finish
```

In [35]:
```
plot_data(pca_acc,fit_acc,y="sin th")
plot_data(pca_center,fit_center,y="center")
plot_data(pca_tm,fit_tm,y="time",log=True)
```







直径の10倍の長さの円柱．分散0.1の誤差を付加. 上部に密度の偏りを入れる

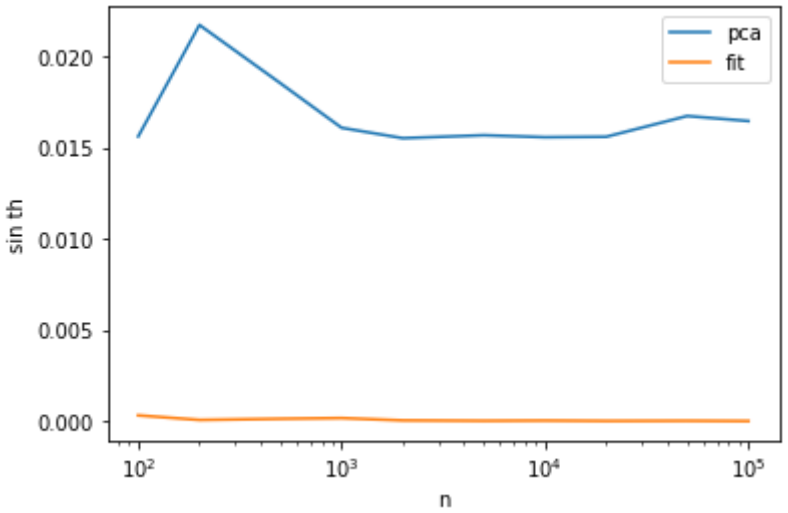
In [36]:
```
r=10
```

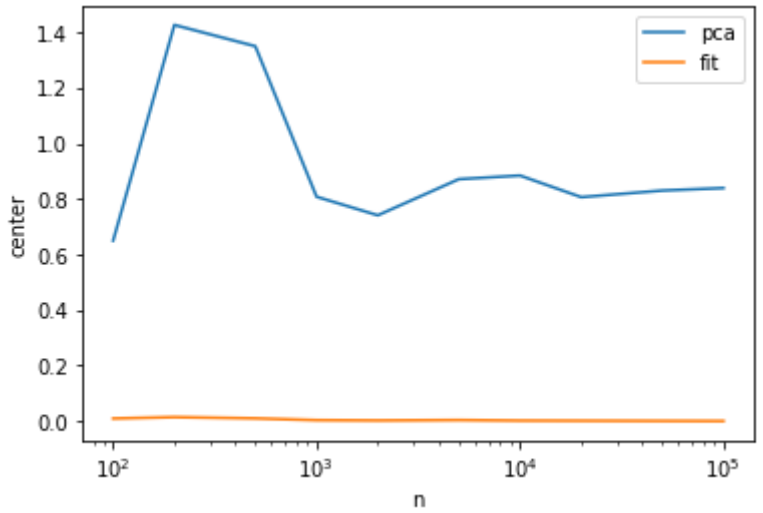
```
l_range=[-100,100]
angle_range=[-pi,pi]

for i in range(n_x) :
    cpc=make_cylinder_point_cloud(x[i],r,l_range,angle_range,sigma=0.1,transform=
    cpc1=make_cylinder_point_cloud(int(x[i]/10),r,[l_range[1]/2,l_range[1]],[0,pi,
    cpc=cpc+cpc1
    print(len(cpc))

    do_pca_fit()

print("finish")
```

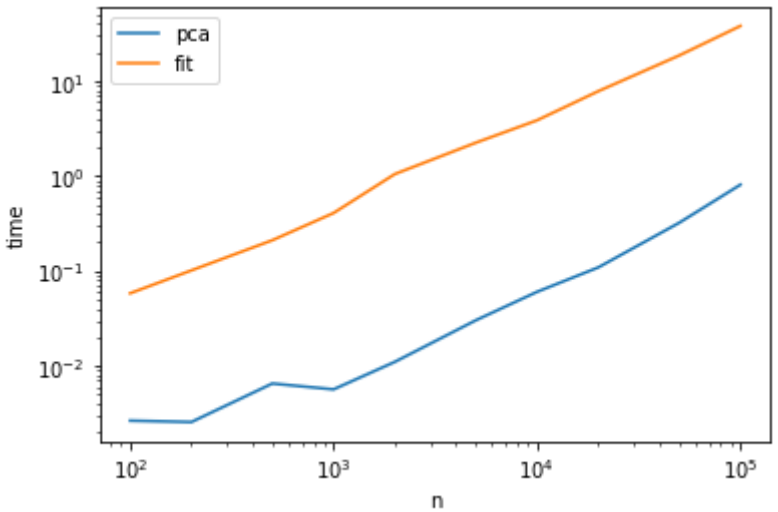
```
110
time= 0.002652406692504883
0.015636062411015277
fit 0 0.05468405841286654
fit 1 0.0021259326599056133
fit 2 0.0003265072440889298
fit 3 0.00032274281934646
fit 4 0.0003227496079552888
fit 5 0.0003227496256401879
time= 0.05822277069091797
220
time= 0.0025658607482910156
0.02175364738539068
fit 0 0.055475029462696786
fit 1 0.0019054708617332824
fit 2 7.342837782854515e-05
fit 3 7.292183873282038e-05
fit 4 7.292244036505221e-05
fit 5 7.292243988863647e-05
time= 0.10144519805908203
550
time= 0.0065267086029052734
0.01855576897494685
fit 0 0.027053857943640194
fit 1 0.00204681272751675
fit 2 0.00013202575457975192
fit 3 0.00013546301152625283
fit 4 0.00013546403045710994
fit 5 0.00013546403082773285
time= 0.21088409423828125
1100
time= 0.005664348602294922
0.016109019433223664
fit 0 0.027362302409600327
fit 1 0.0011996093734511955
fit 2 0.000171848175261297
fit 3 0.00017263837590090382
fit 4 0.00017263893356115186
fit 5 0.00017263893313963733
time= 0.40793943405151367
2200
time= 0.01103353500366211
0.015538616449106268
fit 0 0.03168681099923496
fit 1 0.0017174621518298754
fit 2 4.2588255217572044e-05
fit 3 4.5796372097404904e-05
fit 4 4.579458446655006e-05
```

```
fit 5 4.579458585597218e-05
time= 1.0493686199188232
5500
time= 0.030228376388549805
0.015701039558086202
fit 0 0.02424513486277235
fit 1 0.0011670946108520049
fit 2 2.5034265497405596e-05
fit 3 2.4416973410223992e-05
fit 4 2.4417011502632783e-05
fit 5 2.44170114246363e-05
time= 2.2438955307006836
11000
time= 0.060276031494140625
0.015592389883800259
fit 0 0.03589121200844846
fit 1 0.002191327725856873
fit 2 3.714052780853525e-05
fit 3 3.595427206000013e-05
fit 4 3.595401148452821e-05
fit 5 3.5954011432277476e-05
time= 3.86346173286438
22000
time= 0.10900354385375977
0.015616813137409064
fit 0 0.025183160708747918
fit 1 0.0015143632146937137
fit 2 1.739605715048189e-05
fit 3 1.795264154395738e-05
fit 4 1.7952469681355546e-05
fit 5 1.7952469644547085e-05
time= 7.821526765823364
55000
time= 0.32190918922424316
0.016756279572156148
fit 0 0.030898012053599124
fit 1 0.0017637222707519028
fit 2 2.4647879818615193e-05
fit 3 2.486856205368848e-05
fit 4 2.4868688386266318e-05
fit 5 2.4868688420365767e-05
time= 18.615026235580444
110000
time= 0.8107624053955078
0.01647984073692929
fit 0 0.028520885248200564
fit 1 0.0015625430596335598
fit 2 1.1248028287308342e-05
fit 3 1.2504644077897003e-05
fit 4 1.2504747260537986e-05
fit 5 1.2504747270072346e-05
time= 38.081907510757446
finish
```

In [37]:
```
plot_data(pca_acc,fit_acc,y="sin th")
plot_data(pca_center,fit_center,y="center")
plot_data(pca_tm,fit_tm,y="time",log=True)
```







直径の1倍の長さの円柱．分散0.1の誤差を付加. 上部に密度の偏りを入れる

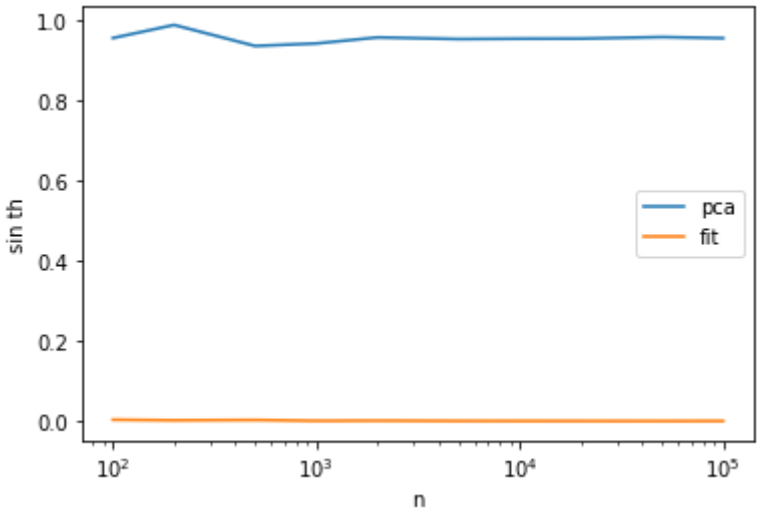
In [38]:
```
r=10
```

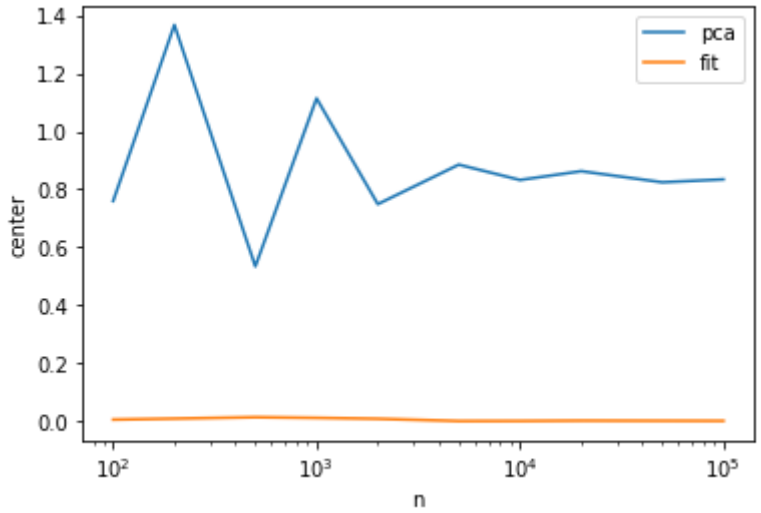
```
l_range=[-10,10]
angle_range=[-pi,pi]

for i in range(n_x) :
    cpc=make_cylinder_point_cloud(x[i],r,l_range,angle_range,sigma=0.1,transform=
    cpc1=make_cylinder_point_cloud(int(x[i]/10),r,[l_range[1]/2,l_range[1]],[0,pi,
    cpc=cpc+cpc1
    print(len(cpc))

    do_pca_fit()

print("finish")
```

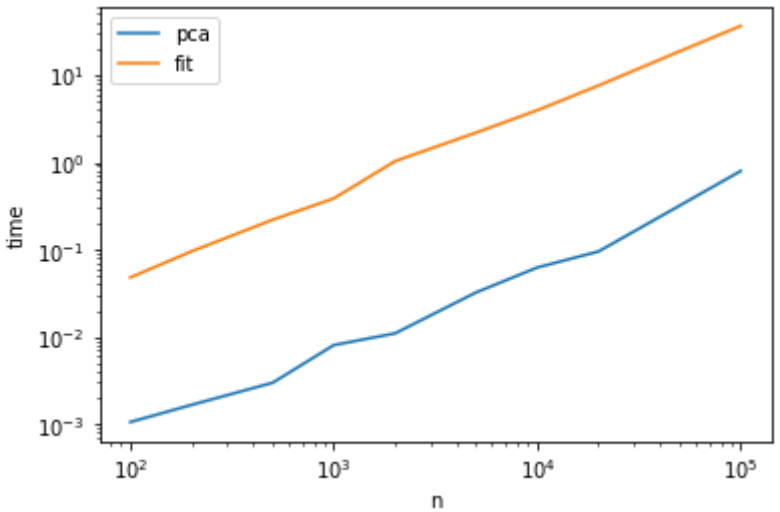
```
110
time= 0.0010590553283691406
0.9569320660906157
fit 0 0.5503623130122703
fit 1 0.14814152623121704
fit 2 0.0023255367589726547
fit 3 0.003261175610337885
fit 4 0.00325895123643025
fit 5 0.0032589591234981727
time= 0.048027753829956055
220
time= 0.001665353775024414
0.9896278339245334
fit 0 0.38093829338183294
fit 1 0.06788517952223729
fit 2 0.002242197410845431
fit 3 0.001832708764018593
fit 4 0.0018319729077520728
fit 5 0.0018319720216748547
time= 0.09569144248962402
550
time= 0.0029807090759277344
0.9371407566449815
fit 0 0.3467008450775932
fit 1 0.004684851382349711
fit 2 0.0025809950305668173
fit 3 0.00256469211012151
fit 4 0.002564726822006913
fit 5 0.0025647267178485955
time= 0.21953678131103516
1100
time= 0.00806117057800293
0.9433626460508893
fit 0 0.3982744242855714
fit 1 0.020586936954644912
fit 2 0.0004916430291111993
fit 3 0.0005414853727423022
fit 4 0.0005415155474518763
fit 5 0.0005415155598507555
time= 0.3862032890319824
2200
time= 0.010976552963256836
0.9585124414449816
fit 0 0.38154714230751
fit 1 0.023479169171103053
fit 2 0.0008218832670018909
fit 3 0.00087018682068865
fit 4 0.000870223055152841
```

```
fit 5 0.0008702231101498704
time= 1.0315792560577393
5500
time= 0.03229022026062012
0.9543964637976063
fit 0 0.3895882144567353
fit 1 0.021612564721960298
fit 2 0.00037475264462204347
fit 3 0.0003150735101333955
fit 4 0.000315020751917005
fit 5 0.00031502069361688084
time= 2.1874265670776367
11000
time= 0.06235766410827637
0.9552406227201237
fit 0 0.395511642884014
fit 1 0.024608853125585084
fit 2 0.00017952448212467246
fit 3 0.0001351123369235536
fit 4 0.00013508701140946733
fit 5 0.00013508699317477888
time= 3.949685573577881
22000
time= 0.0951848030090332
0.9554553441971906
fit 0 0.3967772962872707
fit 1 0.024357871791104522
fit 2 0.00010477679811532075
fit 3 8.747488768904533e-05
fit 4 8.747360837500495e-05
fit 5 8.747360711192796e-05
time= 7.580703496932983
55000
time= 0.31845903396606445
0.9594011000985919
fit 0 0.3931667895368488
fit 1 0.023769260898743667
fit 2 7.218152927215073e-05
fit 3 3.4341327456939516e-05
fit 4 3.432651828607456e-05
fit 5 3.432650944155719e-05
time= 18.671448230743408
110000
time= 0.7951221466064453
0.9565794618076701
fit 0 0.3912400002163232
fit 1 0.023211944198946014
fit 2 0.00017851891560789299
fit 3 0.00013432701241888136
fit 4 0.00013431031535684915
fit 5 0.00013431030958573973
time= 36.49756932258606
finish
```

In [39]:
```
plot_data(pca_acc,fit_acc,y="sin th")
plot_data(pca_center,fit_center,y="center")
plot_data(pca_tm,fit_tm,y="time",log=True)
```







直径の10倍の長さの円柱，分散0.1の誤差を付加. 寸法誤差あり.

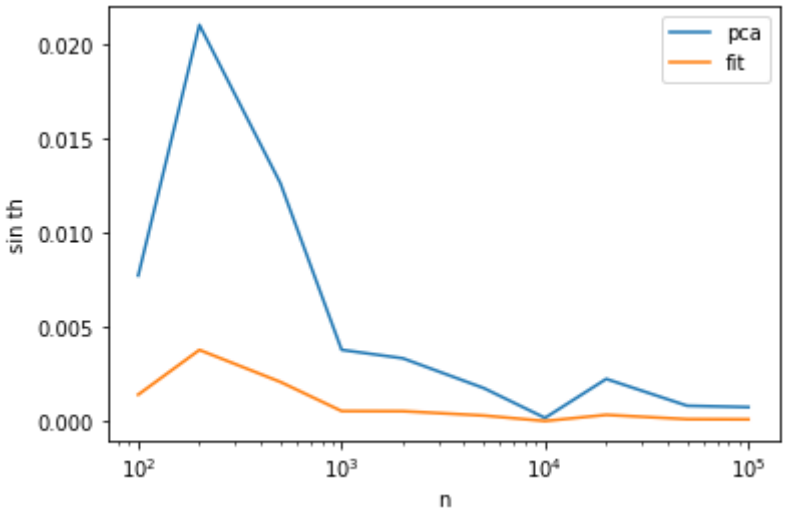
In [40]:
```
r=10
```

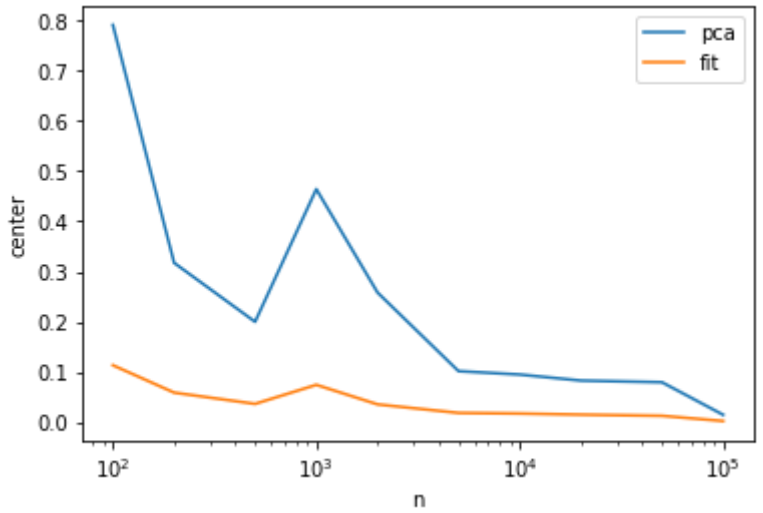
```
r_err=1.1
l_range=[-100,100]
angle_range=[-pi,pi]

for i in range(n_x) :
    cpc=make_cylinder_point_cloud(x[i],r*r_err,l_range,angle_range,sigma=0.1,tran
    print(len(cpc))
    tmp1=calc_pca(cpc)

    do_pca_fit()

print("finish")
```

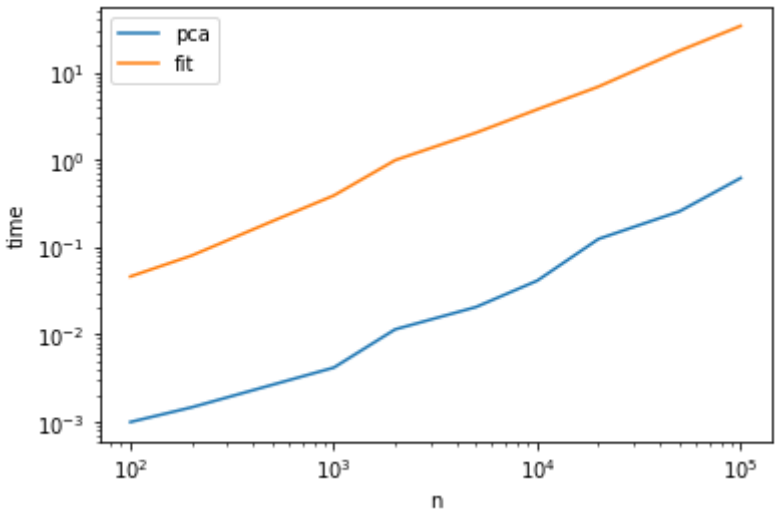
```
100
time= 0.0010008811950683594
0.007733625701393463
fit 0 0.12259178780952934
fit 1 0.01147061976881348
fit 2 0.002848571778925374
fit 3 0.0012694025683641376
fit 4 0.0014559634893641666
fit 5 0.0014329243671867735
time= 0.0461575984954834
200
time= 0.001474142074584961
0.020998657597690856
fit 0 0.1016893270995122
fit 1 0.004840235222159196
fit 2 0.003682086320758199
fit 3 0.003811854349608048
fit 4 0.003797762496328057
fit 5 0.00379927877472264
time= 0.08003973960876465
500
time= 0.0026721954345703125
0.012626168832128248
fit 0 0.039428789561947696
fit 1 0.0032694546510328862
fit 2 0.002357411520523973
fit 3 0.0020833562583793207
fit 4 0.002108972638755824
fit 5 0.0021063586221161503
time= 0.19849491119384766
1000
time= 0.004193544387817383
0.0038038503189395476
fit 0 0.028078083197331336
fit 1 0.0027678603369788815
fit 2 0.0005098650340405763
fit 3 0.0005737305517788287
fit 4 0.000563789784072521
fit 5 0.0005646566795976412
time= 0.3891012668609619
2000
time= 0.011465311050415039
0.003356163744904599
fit 0 0.05909773452783633
fit 1 0.0035486077489881107
fit 2 0.0009263880076854827
fit 3 0.0005251688584542087
fit 4 0.0005614423781581928
```

```
fit 5 0.000558146158696021
time= 0.9848668575286865
5000
time= 0.02065753936767578
0.0017786936277440807
fit 0 0.028501459246145125
fit 1 0.002366619528903089
fit 2 0.00036502556622214116
fit 3 0.00032844285934238254
fit 4 0.000326396582590961
fit 5 0.000326447880461821
time= 2.031876802444458
10000
time= 0.041368961334228516
0.00020141639120990735
fit 0 0.02663796102667565
fit 1 0.002114616307846541
fit 2 0.00019470981136642651
fit 3 3.74522720978446e-05
fit 4 3.731860978852945e-05
fit 5 3.695666567134352e-05
time= 3.7515506744384766
20000
time= 0.12392973899841309
0.0022642123416650676
fit 0 0.026971754616772158
fit 1 0.002104924236008191
fit 2 0.00047860606979644604
fit 3 0.00035645155968701216
fit 4 0.0003637378978444722
fit 5 0.00036305854950266454
time= 6.8372275829315186
50000
time= 0.25604939460754395
0.0008452607508748927
fit 0 0.026631992468217226
fit 1 0.0022421817647561367
fit 2 0.00011519780858835686
fit 3 0.0001484752503052357
fit 4 0.00013371656043392296
fit 5 0.00013499523357002032
time= 17.546934366226196
100000
time= 0.6161842346191406
0.0007717000300232385
fit 0 0.02537648000991867
fit 1 0.0021399098101025113
fit 2 5.231858420728613e-05
fit 3 0.0001403053108426609
fit 4 0.00012452490864924131
fit 5 0.00012586695438351824
time= 33.756627321243286
finish
```

In [41]:
```
plot_data(pca_acc,fit_acc,y="sin th")
plot_data(pca_center,fit_center,y="center")
plot_data(pca_tm,fit_tm,y="time",log=True)
```







In [ ]:
```
```