

```

1 def
  ↳ gen_NorSand_par_LD(dict_ranges_material,dict_ranges_test,n_samples,n_samples_2):
2     sampler = qmc.Sobol(d=len(dict_ranges_material), scramble=True,seed=11)
3     x_mat = sampler.random_base2(m=int(np.log2(n_samples)))
4     data_inp_mat = x_mat.T
5     data_expand_mat = []
6     for ind_vals in range(len(dict_ranges_material)):
7         vlow,vup = list(dict_ranges_material.values())[ind_vals]
8         data_pts = data_inp_mat[ind_vals]
9         data_expand_mat.append((vup-vlow)*data_pts + vlow)
10    data_expand_mat = np.round(np.array(data_expand_mat),4)
11    data_expand_tst_corretos=[]
12    for pbb,yv in enumerate(data_expand_mat.T):
13        samplerinner =
14        ↳ qmc.Halton(d=len(dict_ranges_test),scramble=True,seed=int(11+2*pbb))
15        x_tst = samplerinner.random(n=n_samples_2)
16        data_inp_tst = x_tst.T
17        data_expand_tst = []
18        for ind_vals in range(len(dict_ranges_test)):
19            if ind_vals==0:
20                data_expand_tst.append(data_inp_tst[ind_vals])
21            else:
22                vlow,vup = list(dict_ranges_test.values())[ind_vals]
23                data_pts = data_inp_tst[ind_vals]
24                data_expand_tst.append((vup-vlow)*data_pts + vlow)
25            data_expand_tst = np.array(data_expand_tst)
26            data_expand_tst_prov = data_expand_tst.copy()
27            data_expand_tst_prov[0] = np.array([(np.clip(yv[2]/(yv[4]*(1+yv[3])),0,
28                yv[2]/(5*yv[4]*(1+yv[3])))+0.2)*lhsv-0.2 for lhsv in
29                ↳ data_expand_tst_prov[0]])
30            data_expand_tst_corretos.append(data_expand_tst_prov)
31    data_expand_tst_corretos = np.round(np.array(data_expand_tst_corretos),4)
32    final_comp=[]
33    for mat_vals,tst_vals in zip(data_expand_mat.T,data_expand_tst_corretos):
34        for ti_vals in tst_vals.T:
35            final_comp.append(np.concatenate((mat_vals,ti_vals),axis=0))
36    return final_comp

```