

```

1 def
2   ↪ gen_NorSand_par_2(dict_ranges_material,dict_ranges_test,n_samples,n_samples_2):
3     lhs = Lhs(lhs_type="centered", criterion='maximin')
4     lhsinner = Lhs(criterion="ratio")
5     space_material = Space([(0, 1.) for x in range(len(dict_ranges_material))])
6     space_test = Space([(0, 1.) for x in range(len(dict_ranges_test))])
7     x_mat = lhs.generate(space_material.dimensions, n_samples,random_state=11)
8     data_inp_mat = (np.array(x_mat).T)
9     data_expand_mat = []
10    for ind_vals in range(len(dict_ranges_material)):
11        vlow,vup = list(dict_ranges_material.values())[ind_vals]
12        data_pts = data_inp_mat[ind_vals]
13        data_expand_mat.append((vup-vlow)*data_pts + vlow)
14    data_expand_mat = np.round(np.array(data_expand_mat),4)
15    data_expand_tst_corretos=[]
16    for pbb,yv in enumerate(data_expand_mat.T):
17        x_tst = lhsinner.generate(space_test.dimensions,
18            ↪ n_samples_2,random_state=int(11+2*pbb))
19        data_inp_tst = (np.array(x_tst).T)
20        data_expand_tst = []
21        for ind_vals in range(len(dict_ranges_test)):
22            if ind_vals==0:
23                data_expand_tst.append(data_inp_tst[ind_vals])
24            else:
25                vlow,vup = list(dict_ranges_test.values())[ind_vals]
26                data_pts = data_inp_tst[ind_vals]
27                data_expand_tst.append((vup-vlow)*data_pts + vlow)
28        data_expand_tst = np.array(data_expand_tst)
29        data_expand_tst_prov = data_expand_tst.copy()
30        data_expand_tst_prov[0] = np.array([(np.clip(yv[2]/(yv[4]*(1+yv[3])),0,
31            ↪ yv[2]/(5*yv[4]*(1+yv[3])))+0.2)*lhsv-0.2 for lhsv in
32            ↪ data_expand_tst_prov[0]])
33        data_expand_tst_corretos.append(data_expand_tst_prov)
34    data_expand_tst_corretos = np.round(np.array(data_expand_tst_corretos),4)
35    final_comp=[]
36    for mat_vals,tst_vals in zip(data_expand_mat.T,data_expand_tst_corretos):
37        for ti_vals in tst_vals.T:
38            final_comp.append(np.concatenate((mat_vals,ti_vals),axis=0))
39    return final_comp

```