Electronic Supplementary Material (ESI) for Digital Discovery.

Supporting information for the article:
Robotically automated 3D printing and testing of thermoplastic material specimens

Miguel Hernández-del-Valle,[a,b] Christina Schenk,[a] Lucía Echevarría-Pastrana,[a] Burcu Ozdemir,[a,c] Enrique Dios-Lázaro,[a,b] Jorge Ilarraza-Zuazo,[a,b] De-Yi Wang,[a] and Maciej Haranczyk*[a]

[a] IMDEA Materials Institute, c/Eric Kandel, 2, 28906 Getafe, Madrid, Spain. Tel: +34 915 49 34 22; E-mail: maciej.haranczyk@imdea.org
[b] Universidad Carlos III de Madrid, 28911 Leganés, Madrid, Spain.
[c] Universidad Politécnica de Madrid, 28040 Madrid, Spain.

Contents:

Section 1: Example of code for running Bayesian Optimization

This section contains the excerpts of the code used for running the Bayesian Optimization with a single printer. For clarity, only the lines directly related with the Bayesian Optimization have been included, so the code is not executable by itself.

We begin by importing the GPOptimizer from the GPcam library, that can be installed with "pip install gpcam"

```
from gpcam.gp_optimizer import GPOptimizer
```

First we define the parameter bounds

```
s_lb = 50
s_ub = 150
f_lb = 1000
f_ub = 5000
lh_lb = 0.2
lh_ub = 0.6
lw_lb = 0.6
lw_ub = 1.0

parameter_bounds = np.array([[s_lb, s_ub],[f_lb,f_ub], [lh_lb, lh_ub],
[lw_lb, lw_ub]])
```

Secondly, we define the bounds for the hyperparameters

```
hyperparameter_bounds =
np.array([[0.001,1000],[0.001,1000],[0.001,1000],[0.001,1000],[0.001,1000]]
)
guess = (0.001+1000.)/2.
hps_guess = np.array([guess, guess, guess, guess, guess])
```

We can now initialize and train a GP with those parameters and hyperparams

```
gp = GPOptimizer(dim, parameter_bounds)

gp.init_gp(hps_guess)
gp.train_gp(hyperparameter_bounds, pop_size = 20, tolerance = 1e-6,
max_iter = 20)
```

We define an acquisition function, in this case UCB with lambda=1

```
def optional_acq_func(x,obj):
    a = 1.0
    m = obj.posterior_mean(x)["f(x)"]
    v = obj.posterior_covariance(x)["v(x)"]
    return m + a * np.sqrt(v)
```

We ask GPcam for a first suggestion to start printing

```
nextp = gp.ask(position = None, n = 1, acquisition_function =
```

```
          optional_acq_func,
          bounds = parameter_bounds,
          method = "global", pop_size = 50, max_iter = 20,
          tol = 1e-6, dask_client = False)
```

Generate a gcode using the parameters suggested by GPcam

```
speed, flow, lh, lw = extract_parameters(nextp)
gcode = set_print_parameters(speed, flow, lh, lw)
start_print_with_gcode(gcode, printer) #sends the gcode to the printer and
starts the printing
```

We start a loop to do 20 experiments

```
counter = 0

while counter < 20:

    if printer.printing == False: # The next part is only executed once the
printing has finished
```

The following line executes all the experimental workflow to measure weight and impact strength

```
        weights, impacts = after_print_workflow_charpy()
        mean_weight_diff = np.mean(weights)/expected_weight
```

We save the parameters as "points" and the results as "values"

```
        points = np.vstack((points, [speed, flow, lh, lw]))
        values = np.vstack((values, -abs(1-mean_weight_diff)))
```

We update the GP by introducing the new points and values

```
        gp.tell(points, values)
        gp.train_gp(hyperparameter_bounds, pop_size = 20, tolerance = 1e-6,
max_iter = 20)
```

We ask for the next point to experiment with

```
        nextp = gp.ask(position = None, n = 1,
               acquisition_function = optional_acq_func,
               bounds = parameter_bounds, method = "global",
               pop_size = 50, max_iter = 20,
               tol = 1e-6, dask_client = False)
```

Generate a gcode using the parameters suggested by GPcam

```
        speed, flow, lh, lw = extract_parameters(nextp)
        gcode = set_print_parameters(speed, flow, lh, lw)
        start_print_with_gcode(gcode, printer) #sends the gcode to the
printer and starts the printing

        counter = counter+1
```

Section 2: Example of code for running Latin Hypercube Sampling

Example of the usage of the LatinHypercube function from Scipy to generate 10 LHS points in 4 dimensions:

```
from scipy.stats import qmc
sampler = qmc.LatinHypercube(d=4, centered=True, seed = 42)
sample = sampler.random(n=10)
```

Now, we define the lower and upper bounds of our parameters to rescale the points:

```
s_lb = 50
s_ub = 150
f_lb = 1000
f_ub = 5000
lh_lb = 0.2
lh_ub = 0.6
lw_lb = 0.6
lw_ub = 1.0

l_bounds = [s_lb, f_lb, lh_lb, lw_lb}
u_bounds = [s_ub, f_ub, lh_ub, lw_ub]

lhs_samples = qmc.scale(sample, l_bounds, u_bounds)
```

Section 3: Data used to generate the figures of the article.

Table 1: Data for Bayesian Optimization of HPLA used in Figures 6, 7, 8, 9 and 10.

Speed	Flow	Layer Height	Layer Width	Weight	Impact
94,5903693	1041,65434	0,53675802	0,69589768	0,60393145	0,41009389
108,436089	2975,89192	0,2431057	0,83511068	0,87726815	0,72662379
109,763889	4993,89602	0,49855709	0,6035247	0,93644153	0,84169986
111,956245	2691,81433	0,49870066	0,73357306	0,79601815	0,69892512
82,6626075	4999,46923	0,50198577	0,63066122	1,06507056	0,84279126
57,2433828	4938,40829	0,45838455	0,90071234	0,93795363	0,59681953
107,401138	4243,31906	0,27547764	0,77933314	1,01498656	0,77171416
107,542951	4184,68348	0,2108603	0,99248713	1,05887097	0,82199876
149,987655	4142,87596	0,53442363	0,90438772	0,8078629	0,32381678
50,0292811	3876,45742	0,34025611	0,68351653	0,95811492	0,73891521
149,982262	4514,98198	0,2642811	0,89164408	1,0733871	0,84229443
50,0408255	4470,01475	0,37867271	0,92179829	0,95861895	0,66741776
50,0061998	4369,65177	0,392102	0,84695193	1,07867944	0,89553473
50,0133918	4384,84741	0,30264408	0,98608627	1,08165323	0,92784289
50,0088971	4208,84476	0,217873	0,89826837	0,93825605	0,78912434
94,0673728	4535,03852	0,21639402	0,64074535	1,03190524	0,91213151
96,6597181	4552,92591	0,22044973	0,64277159	1,08884409	0,86208484
50,0033571	3497,84369	0,38406129	0,78444204	0,9624328	0,7528745
96,7048519	4552,07931	0,20831767	0,8223425	0,93331653	0,66065502
50,007354	3670,56038	0,31107213	0,63012847	0,96586022	0,83161934

Table 2: Data for LHS sampling of HPLA used in Figures 6, 7, 8.

Flow	Speed	Layer Height	Layer Width	Weight	Impact
2800	105	0,22	0,86	1,08971774	0,99790416
4400	115	0,58	0,66	1,11606183	1,02996916
2000	55	0,54	0,7	0,98462702	0,89569572
3600	125	0,38	0,9	1,11149194	1,02184707
3200	85	0,5	0,98	1,08998656	0,97664081
4800	75	0,3	0,82	1,12530242	1,01381798
4000	95	0,34	0,94	1,12636089	1,03008784
2400	145	0,42	0,78	1,07419355	0,95857903
1200	65	0,46	0,62	0,81236559	0,55235621
1600	135	0,26	0,74	0,97011089	0,78930356

Table 3: Data for Bayesian Optimization of LPLA used in Figure 11.

Speed	Flow	Layer Height	Layer Width	Weight	Impact
94,5787594	1001,85126	0,25048521	0,93907228	0,4563004	0,23329012
123,003642	3268,65458	0,21959061	0,87964956	0,97898185	0,73050592
75,1124844	3221,13604	0,29426776	0,69975638	0,98109879	0,53842761
146,727426	4999,80138	0,4639792	0,93109783	1,14314516	0,89651697
50,0364714	3909,63176	0,53026139	0,83252488	1,02560484	0,581533
50,0012254	3787,78131	0,38458283	0,6670455	1,12464718	0,8885435
50,0024015	3065,48232	0,41528671	0,87182408	1,06895161	0,87227732
50,0056943	4288,44831	0,33875642	0,88064299	1,00010081	0,76696961
149,9796	2816,26051	0,57023334	0,73228711	1,08402218	0,92683076
149,986199	4294,17736	0,2843375	0,92547587	1,16078629	0,82140682
149,999979	4201,37046	0,37140349	0,94955506	0,97091734	0,72311592
50,0045205	4580,55488	0,55873636	0,92633094	1,17083333	0,94551878
50,0063583	4108,98658	0,50320312	0,68616002	0,97726815	0,73761287
149,995383	4220,84465	0,49876826	0,86690372	1,1594086	1,02012966
50,016024	4114,46494	0,46482161	0,7395485	1,13561828	0,96663776
149,979315	3189,38516	0,58937482	0,90936139	0,7609879	0,73031794
149,993287	3168,41926	0,34822869	0,76311552	1,17399194	1,05165579
149,99752	3328,86887	0,41125511	0,93782956	1,17172379	0,99827223
99,1819121	2951,43232	0,56961699	0,85709064	0,9453125	0,75283104
136,403963	3221,44574	0,51973209	0,95578727	1,18830645	1,00846525

Table 4: Data for LHS sampling of LPLA used in Figure 11.

Speed	Flow	Layer Height	Layer Width	Weight	Impact
105	2800	0,22	0,86	1,00947581	0,91910473
115	4400	0,58	0,66	1,03261089	0,93440285
55	2000	0,54	0,7	1,07896505	0,95562446
85	3200	0,5	0,98	0,99146505	0,90318142
125	3600	0,38	0,9	1,11507056	1,02987616
75	4800	0,3	0,82	1,09309476	0,93440285
95	4000	0,34	0,94	1,13172043	0,96593215
145	2400	0,42	0,78	0,90141129	0,78156169
65	1200	0,46	0,62	0,88004032	0,49175611
135	1600	0,26	0,74	0,78608871	0,61723864

Table 5: Data of LHS repeated in the two printers for two runs each to test repeatability (Figure 12)

Printer run	Speed	Flow	LH	LW	Weight
1	1.05e+02	2.80e+03	2.20e-01	8.60e-01	3.97
1	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.04
1	1.05e+02	2.80e+03	2.20e-01	8.60e-01	3.81
1	1.05e+02	2.80e+03	2.20e-01	8.60e-01	3.98
2	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.29
2	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.29
2	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.31
2	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.32
1	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.14
1	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.22
1	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.15
1	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.19
2	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.50
2	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.47
2	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.48
2	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.44
1	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.52
1	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.57
1	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.53
1	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.64
2	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.13
2	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.08
2	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.07
2	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.12
1	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.23
1	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.22
1	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.21
1	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.26
2	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.39
2	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.44
2	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.44
2	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.42
1	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.19
1	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.21
1	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.16
1	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.20
2	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.40
2	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.36

2	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.38
2	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.38
1	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.34
1	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.36
1	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.32
1	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.36
2	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.39
2	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.37
2	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.39
2	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.41
1	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.20
1	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.20
1	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.21
1	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.19
2	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.44
2	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.43
2	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.46
2	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.44
1	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.02
1	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.88
1	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.85
1	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.91
2	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.37
2	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.35
2	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.33
2	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.32
1	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.53
1	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.74
1	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.76
1	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.73
2	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.53
2	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.55
2	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.49
2	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.56
1	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.39
1	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.31
1	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.38
1	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.44
2	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.93
2	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.93
2	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.91
2	1.35e+02	1.60e+03	2.60e-01	7.40e-01	4.00

3	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.13
3	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.09
3	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.12
3	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.19
4	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.33
4	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.35
4	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.32
4	1.05e+02	2.80e+03	2.20e-01	8.60e-01	4.36
3	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.29
3	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.28
3	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.23
3	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.21
4	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.49
4	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.48
4	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.49
4	1.15e+02	4.40e+03	5.80e-01	6.60e-01	4.45
3	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.83
3	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.77
3	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.75
3	5.50e+01	2.00e+03	5.40e-01	7.00e-01	3.76
4	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.12
4	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.10
4	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.12
4	5.50e+01	2.00e+03	5.40e-01	7.00e-01	4.09
3	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.19
3	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.14
3	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.12
3	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.18
4	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.43
4	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.44
4	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.38
4	1.25e+02	3.60e+03	3.80e-01	9.00e-01	4.44
3	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.04
3	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.09
3	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.17
3	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.14
4	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.38
4	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.37
4	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.38
4	8.50e+01	3.20e+03	5.00e-01	9.80e-01	4.35
3	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.36
3	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.34

3	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.40
3	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.36
4	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.42
4	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.33
4	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.36
4	7.50e+01	4.80e+03	3.00e-01	8.20e-01	4.48
3	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.13
3	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.20
3	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.30
3	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.27
4	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.45
4	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.48
4	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.50
4	9.50e+01	4.00e+03	3.40e-01	9.40e-01	4.46
3	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.80
3	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.95
3	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.93
3	1.45e+02	2.40e+03	4.20e-01	7.80e-01	3.88
4	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.32
4	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.34
4	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.34
4	1.45e+02	2.40e+03	4.20e-01	7.80e-01	4.31
3	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.74
3	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.77
3	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.84
3	6.50e+01	1.20e+03	4.60e-01	6.20e-01	2.59
4	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.63
4	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.54
4	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.67
4	6.50e+01	1.20e+03	4.60e-01	6.20e-01	3.60
3	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.42
3	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.26
3	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.32
3	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.44
4	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.96
4	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.99
4	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.92
4	1.35e+02	1.60e+03	2.60e-01	7.40e-01	3.95

Table 6: Data for Bayesian Optimization of HPLA in a single printer (Figure 13)

Flow	Speed	Layer Height	Layer Width	Weight	Impact
4991,62016	137,445129	0,40883653	0,671793	1,10241935	0,93440285
1000,5315	52,1313499	0,2707154	0,67169354	0,6188004	0,36397312
4821,84146	75,467274	0,36410931	0,93649548	1,09764785	0,94551878
4129,60814	149,991028	0,45356054	0,80265424	1,06441532	0,88772937
3530,29496	50,0095311	0,53981602	0,7882906	1,05685484	0,86495958
3095,3563	149,99716	0,57170345	0,82408563	1,04482527	0,94532309
2824,3401	50,0348354	0,28113368	0,83709138	1,00665323	0,93447982
2615,91355	149,986158	0,20868301	0,99019419	0,9749496	0,84950986
2779,95271	50,0069854	0,33205365	0,85637678	1,05120968	0,91153496
2851,38039	149,982472	0,35882803	0,64186208	1,02268145	0,79157875
2798,00399	149,990007	0,28927794	0,97257634	1,03462702	0,94243673
2958,87771	50,0325984	0,28849753	0,7217728	1,03361895	0,74808247
2552,77932	50,0076875	0,31171302	0,79319913	0,98981855	0,87251997
2436,15713	50,0286186	0,45313225	0,9731165	0,96606183	0,79164522
2622,49107	50,0003147	0,35863958	0,72309482	1,00806452	0,91879214
2619,60609	50,0086551	0,49248131	0,77885676	1,02701613	0,88805136
2557,55358	149,987843	0,29969475	0,62628683	0,99388441	0,79152722
2544,87418	149,996106	0,22789782	0,89792753	1,02363911	0,8283349
2607,08274	50,0081727	0,23788634	0,60355119	1,00302419	0,66754689
2607,14055	50,0003415	0,30560658	0,92893133	0,99791667	0,89298583