Masterarbeit

Yannick Kees

$$\sum_{p \in P} \left| \int_{B_\rho(p)} u(x) \, \mathrm{d}x \right|$$

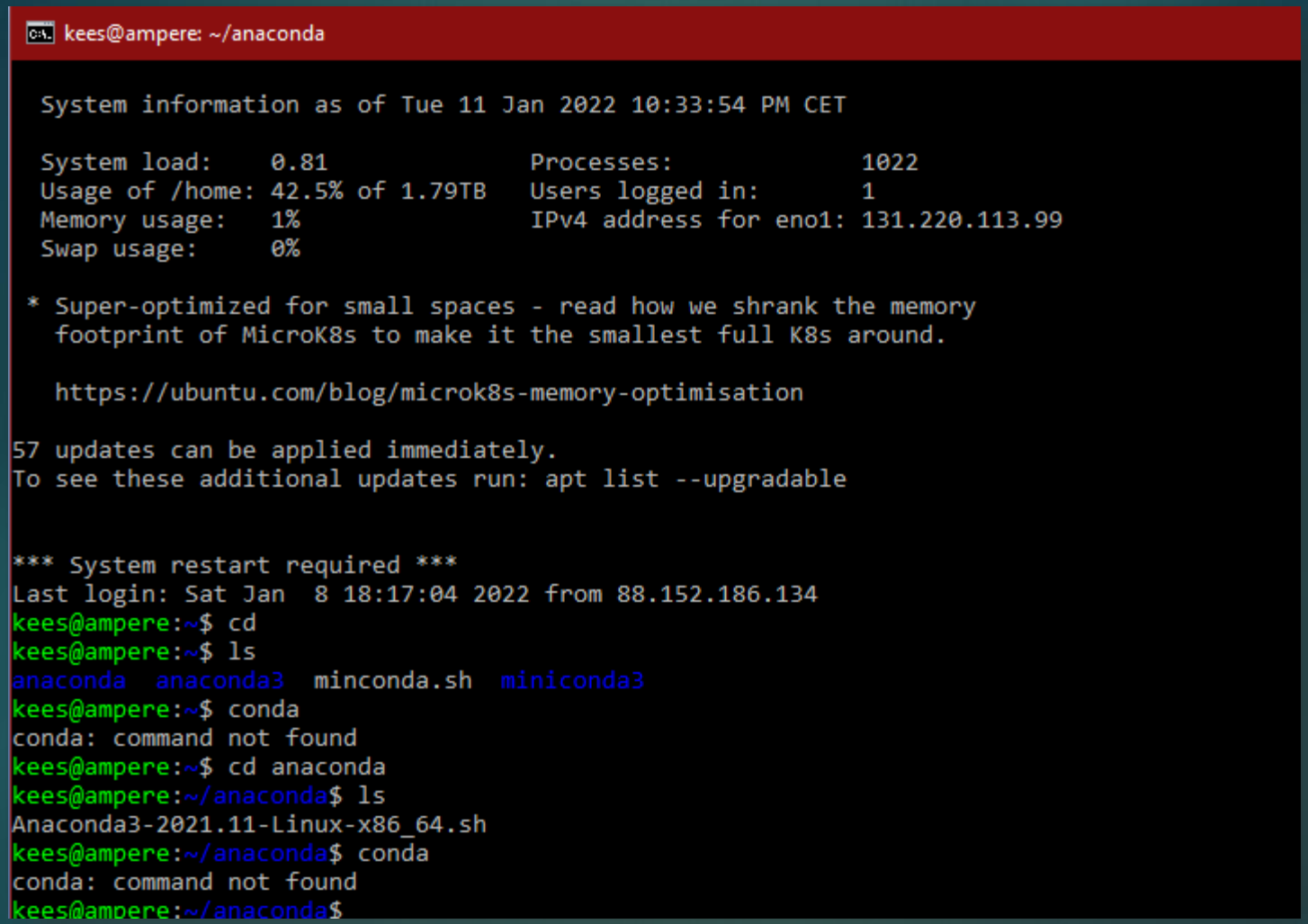
```
#   d:      Dimension of point cloud

loss = 0    # loss in the sum

for point in pc:
    # loop over all points in pointcloud, i.e. x\in X

    variation  = torch.normal(mean = torch.full(size=( n* d,1), fill_value=0.0) , std= torch.full(size=(n*d,1), fill_value=.001) )  # Ra
    start_point = point.repeat(n,1)+   torch.reshape( variation, (n, d) )                                                          #
    start_point = start_point.to(device)
    loss +=  torch.abs(f(start_point).mean())

return  c*eps**(1.0/3.0)/(len(pc)) *  loss      # returns C * eps^(1/3) * 1/|X| * \sum_{x\in X} |\dashint_{B_delta(x)} u(x) dx|
```


kees@ampere: ~/anaconda
System information as of Tue 11 Jan 2022 10:33:54 PM CET
System load: 0.81
Usage of /home: 42.5% of 1.79TB
Memory usage: 1%
Swap usage: 0%
Processes: 1022
Users logged in: 1
IPv4 address for eno1: 131.220.113.99
* Super-optimized for small spaces - read how we shrank the memory
footprint of MicroK8s to make it the smallest full K8s around.
https://ubuntu.com/blog/microk8s-memory-optimisation
57 updates can be applied immediately.
To see these additional updates run: apt list --upgradable
*** System restart required ***
Last login: Sat Jan 8 18:17:04 2022 from 88.152.186.134
kees@ampere:~$ cd
kees@ampere:~$ ls
anaconda anaconda3 minconda.sh miniconda3
kees@ampere:~$ conda
conda: command not found
kees@ampere:~$ cd anaconda
kees@ampere:~/anaconda$ ls
Anaconda3-2021.11-Linux-x86_64.sh
kees@ampere:~/anaconda$ conda
conda: command not found
kees@ampere:~/anaconda$

Loss?

$$\sum_{p \in P} \left| \int_{B_\rho(p)} f(u(x)) \, \mathrm{d}x \right|$$

$$f(s) = 2s - 1$$

$$\sum_{p \in P} |u(p)|$$

by a 3-*tuple* of real values. The manifold $\delta M'$ is represented as a set of planar triangles of $\mathbf{R}^3$

The algorithm is subdivided in the following steps:

1. Bounding box B calculation of P.
2. Generation of a cubical grid G inside B.
3. Calculate the minimum distance for each vertex $v_{i,j,k} \in G$ to P.
4. Calculate the intersection points I in the edges of the cells of G.
5. Identify for each cell of G the marching cube case which match the intersection points I.
6. Generate the simplexes of each cell of G given the calculated case.

3.1 Bounding Box Calculation

MARCHING CUBES IN AN UNSIGNED DISTANCE FIELD FOR SURFACE RECONSTRUCTION FROM UNORGANIZED POINT SETS

Algorithm 1: NDF: Dense PCs

$\mathcal{P}_{\text{init}}$: m points uniformly sampled in BB

$\mathcal{P}_{\text{init}} \leftarrow \{\mathbf{p} \in \mathcal{P}_{\text{init}} | f(\mathbf{p}) < \delta\}$

for $i = 1$ to num_steps **do**

$$\mathbf{p} \leftarrow \mathbf{p} - f(\mathbf{p}) \cdot \frac{\nabla_{\mathbf{p}} f(\mathbf{p})}{\|\nabla_{\mathbf{p}} f(\mathbf{p})\|}, \quad \forall \mathbf{p} \in \mathcal{P}_{\text{init}}$$

end for

$\mathcal{P}_{\text{dense}}$: n points drawn with replacement from $\mathcal{P}_{\text{init}}$

$\mathcal{P}_{\text{dense}} \leftarrow \{\mathbf{p}+\mathbf{d} | \mathbf{p} \in \mathcal{P}_{\text{dense}}, \mathbf{d} \sim \mathcal{N}(0, \delta/3)\}$

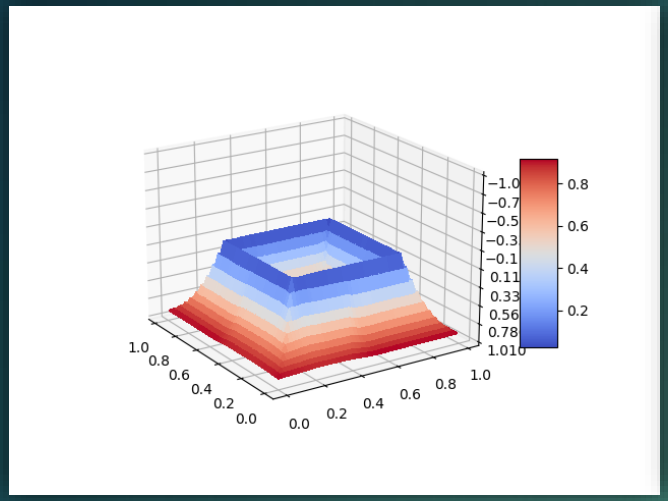
for $i = 1$ to num_steps **do**

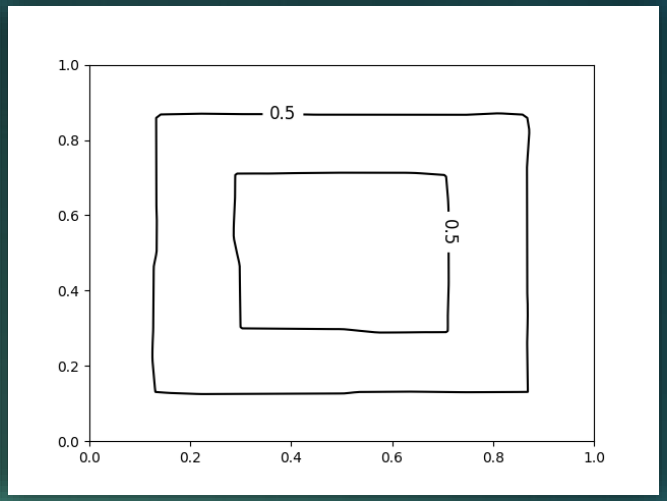
$$\mathbf{p} \leftarrow \mathbf{p} - f(\mathbf{p}) \cdot \frac{\nabla_{\mathbf{p}} f(\mathbf{p})}{\|\nabla_{\mathbf{p}} f(\mathbf{p})\|}, \quad \forall \mathbf{p} \in \mathcal{P}_{\text{dense}}$$

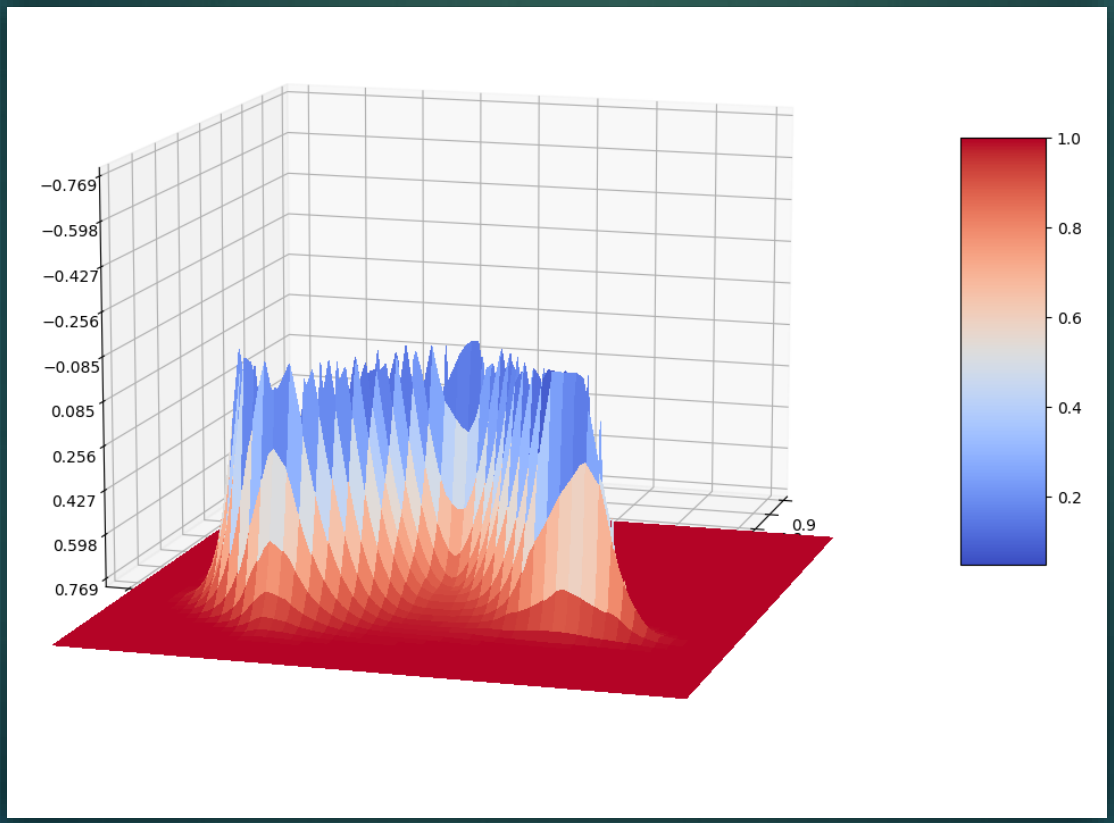
end for

return $\{\mathbf{p} \in \mathcal{P}_{\text{dense}} | f(\mathbf{p}) < \delta\}$

Neural Unsigned Distance Fields for Implicit Function Learning


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
−1.0
−0.7
−0.5
−0.3
−0.1
0.11
0.33
0.56
0.78
1.010
0.8
0.6
0.4
0.2


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
0.5
0.5

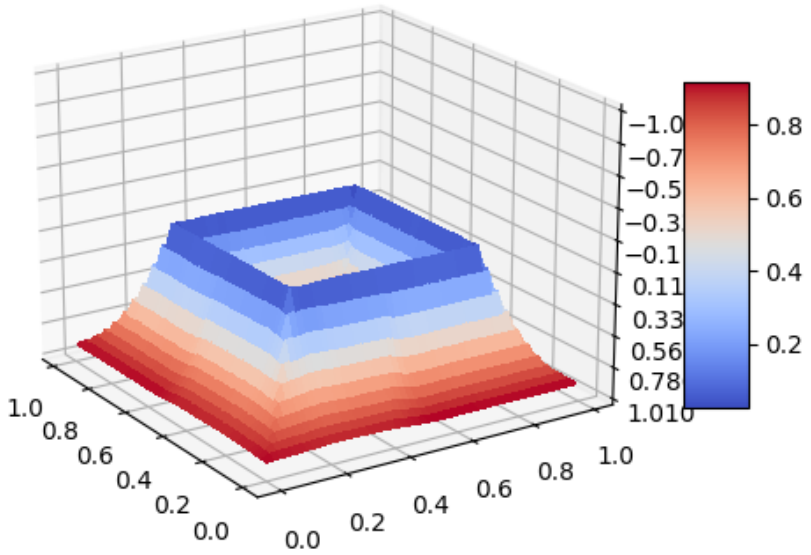

−0.769
−0.598
−0.427
−0.256
−0.085
0.085
0.256
0.427
0.598
0.769
0.9
1.0
0.8
0.6
0.4
0.2

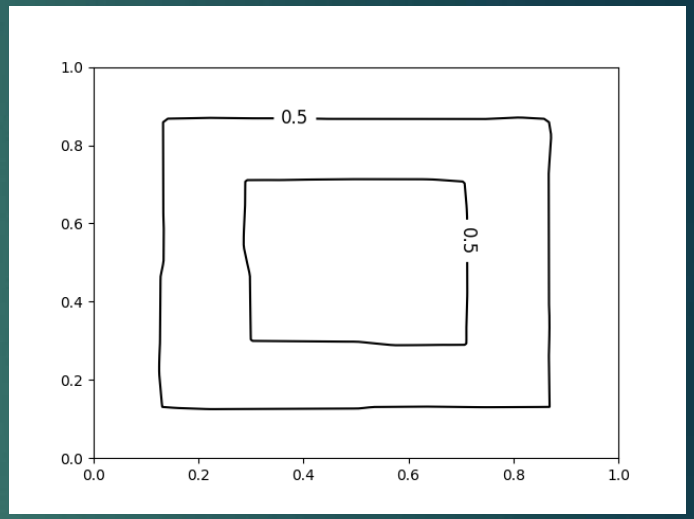
Ambrosio Tortorelli ist definiert als

$$\min_{u,v} \int_\Omega (v(x)^2 + k_\varepsilon)|\nabla u(x)|^2 + \varepsilon|\nabla v(x)|^2 + \frac{(1-v(x))^2}{4\varepsilon}\,\mathrm{d}x + \underbrace{\|u-g\|_{L^2}^2}_{=:\Phi} \tag{1}$$

$$\Phi \leftarrow \sum_{p\in\mathcal{P}} |u(p)|^2 \text{ oder } \Phi \leftarrow \sum_{p\in\mathcal{P}} \fint_{B_\delta(p)} u,$$

$$\min_{v} \int_\Omega \varepsilon|\nabla v(x)|^2 + \frac{(1-v(x))^2}{4\varepsilon}\,\mathrm{d}x + \sum_{p\in\mathcal{P}} |v(p)|^2$$


−1.0
−0.7
−0.5
−0.3
−0.1
0.11
0.33
0.56
0.78
1.010
0.8
0.6
0.4
0.2
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


0.5
0.5
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

[HQD02] Hugues Quynh Dinh, Greg S. Greg Turk T. Greg Turk: Reconstructing Surfaces By Volumetric Regularization Using Radial Basis Functions. In: *IEEE Trans. on Pattern Analysis and Machine Intelligence 24*, 2002, S. 1358–1371

[KH13] Kazhdan, Michael ; Hoppe, Hugues: Screened poisson surface reconstruction. In: *ACM Transactions on Graphics 32* (2013), jun, Nr. 3, S. 1–13. http://dx.doi.org/10.1145/2487228.2487237. – DOI 10.1145/2487228.2487237

[MK06] Michael Kazhdan, Hugues H. Matthew Bolitho B. Matthew Bolitho: Poisson surface reconstruction. In: *Eurographics Symposium on Geometry processing* (2006), S. 61–70

[PA07] P. Alliez, Y. Tong M. D. D. Cohen-Steiner: Voronoi-based Variational Reconstructionof Unoriented Point Sets. In: *Computer Graphics Forum (Proc. of the Symposium on Geometry Processing)*, 2007

[TO02] Turk, Greg ; O'Brien, James F.: Modelling with implicit surfaces that interpolate. In: *ACM Transactions on Graphics 21* Bd. 4, ACM Press, 2002, S. 855–873

[ZO02] Zhao, Hongkai ; Osher, Stanley: Visualization, analysis and shape reconstruction of unorganized data sets. In: *Geometric Level Set Methods in Imaging, Vision and Graphics* (2002)

[ZZW19] Zhong, Deyun ; Zhang, Ju ; Wang, Liguan: Fast Implicit Surface Reconstruction for the Radial Basis Functions Interpolant. In: *Applied Sciences 9* (2019), dec, Nr. 24, S. 5335. http://dx.doi.org/10.3390/app9245335. – DOI 10.3390/app9245335

Modica Mortola

Optimales Profil MM

$$\begin{cases} v' = \sqrt{W(v)} \\ v(0) = \dfrac{w+z}{2} \end{cases}$$

S. 106

Optimales Profil W(s)=(s^2-1)^2

$$\begin{cases} v' = \sqrt{W(v)} \\ v(0) = \dfrac{w+z}{2} \end{cases}$$

$$v' = 1 - v^2$$

$$\frac{1}{1-v^2}dv = 1dx$$

$$arctanh(v) = x$$

$$v = tanh(x)$$

Optimales Profil für W(s)=s^2-2|s|+1

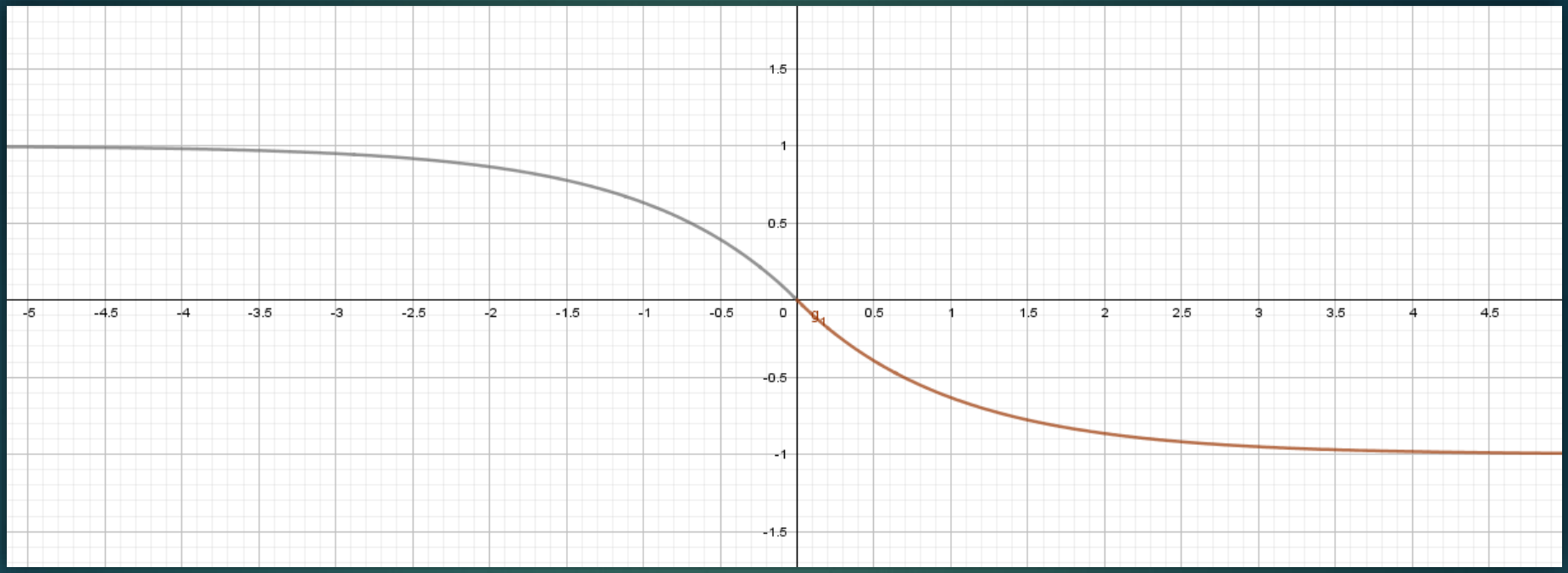
$$\begin{cases} v' = \sqrt{W(v)} \\ v(0) = \dfrac{w+z}{2} \end{cases}$$

$$v' = \sqrt{(v^2 - 2|v| + 1)}$$

$$\frac{1}{\sqrt{(v^2 - 2|v| + 1)}} dv = 1dx$$

$$\frac{v\ ln(||v| - 1|)}{|v|} = x$$

$$v = \begin{cases} 1 - e^x & x \leq 0 \\ e^{-x} - 1 & x > 0 \end{cases}$$


1.5
1
0.5
-0.5
-1
-1.5
-5
-4.5
-4
-3.5
-3
-2.5
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5

```python
def Zero_recontruction_loss_Lip(f, pc, eps, m, c, d):
    # Returns:
    #   Monte Carlo Estimation of C * eps^(1/3) * 1/|X| * \sum_{x\in X} |\dashint_{B_delta}(x) u(s) ds|

    # Parameters:
    #   f:      Function to evaluate
    #   pc:     Pointcloud X
    #   eps:    Epsilon
    #   c:      Constant
    #   n:      Number of samples drawn in the Monte Carlo Algorithm
    #   d:      Dimension of point cloud

    n = len(pc)

    matrix = pc.repeat(m,1)
    matrix = torch.reshape(matrix, (m,n,d)) # 3D Matrix containing the points
    variation  = torch.normal(mean = torch.full(size=( n*m *d,1), fill_value=0.0) , std= torch.full(size=(m*n*d,1), fill_value=.001) )
    error = torch.reshape( variation, (m,n, d) ) # 3D Matrix containing normal distribution
    matrix += error
    matrix = matrix.reshape(m*n,d)
    matrix = f(matrix)  # Apply network to targets
    matrix = torch.reshape( matrix, (m,n) ).mean(0)
    matrix = torch.abs(matrix).mean()

    return  c*eps**(1.0/3.0) *  matrix      # returns C * eps^(1/3) * 1/|X| * \sum_{x\in X} |\dashint_{B_delta(x)} u(x) dx|
```

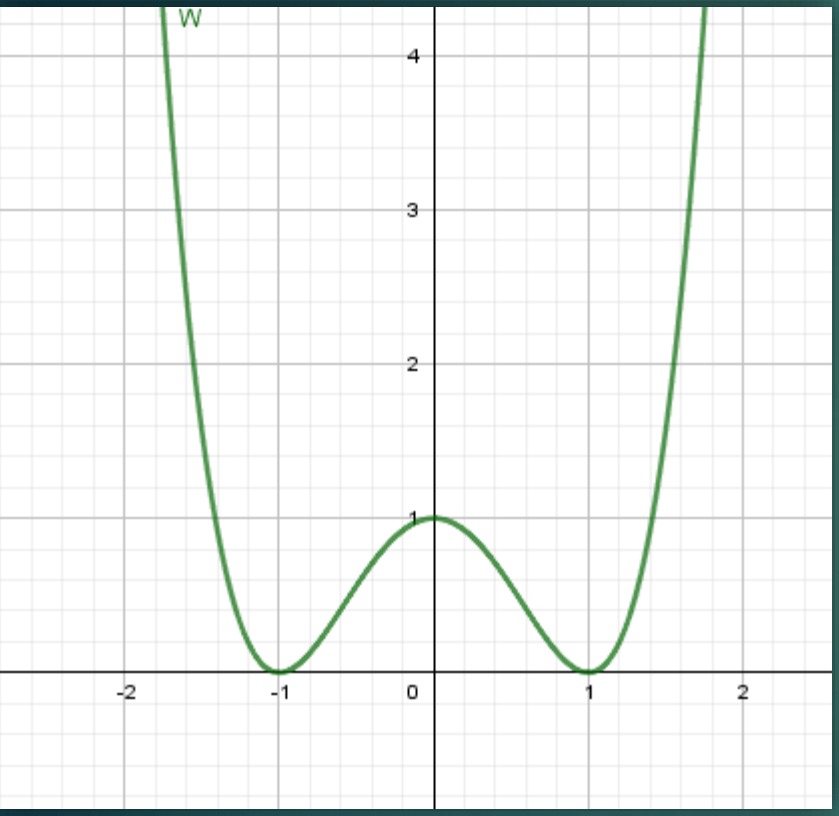
$$\sum_{p \in P} \left| \int_{B_\delta(p)} u(x) dx \right|$$

Fourier Features

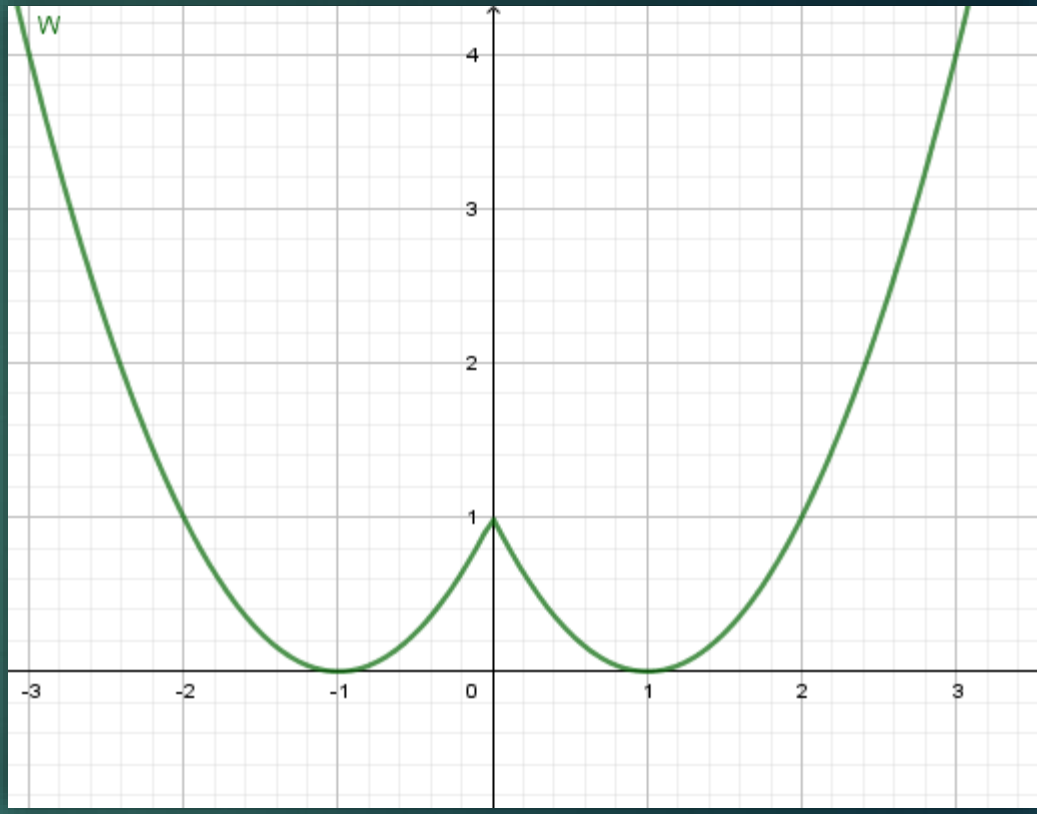
Gaussian: $\gamma(\mathbf{v}) = [\cos(2\pi \mathbf{B}\mathbf{v}), \sin(2\pi \mathbf{B}\mathbf{v})]^{\mathrm{T}}$, where each entry in $\mathbf{B} \in \mathbb{R}^{m \times d}$ is sampled from $\mathcal{N}(0, \sigma^2)$, and σ is chosen for each task and dataset with a hyperparameter sweep. In the absence of any strong prior on the frequency spectrum of the signal, we use an isotropic Gaussian distribution.

Our experiments show that all of the Fourier feature mappings improve the performance of coordinate-based MLPs over using no mapping and that the Gaussian RFF mapping performs best.

9/16(s^2-1)^2



s^2-2|s|+1



Max. in 0

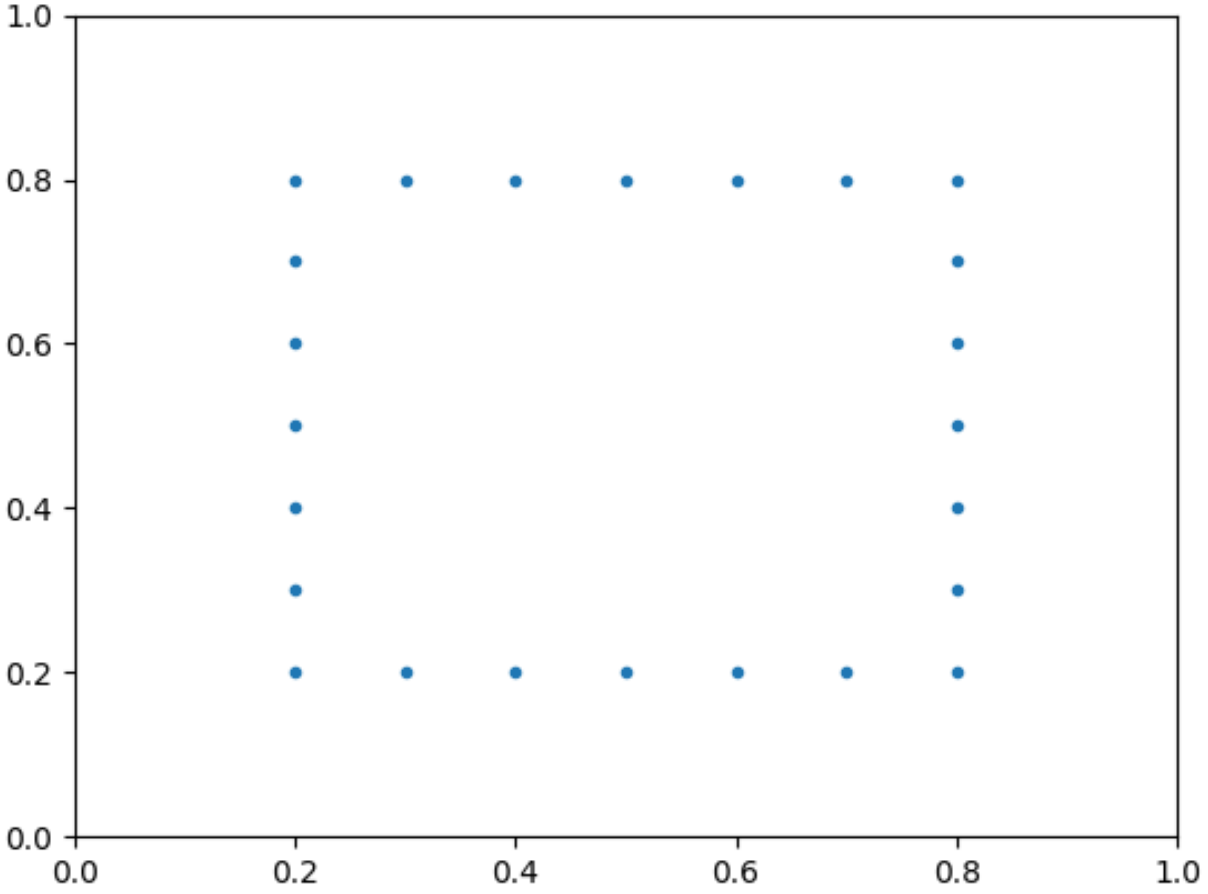
$$\int_\Omega \underbrace{\tfrac{1}{\varepsilon}W(u)}_{1} + \underbrace{\varepsilon|\nabla u|^2}_{2} + \underbrace{\frac{\lambda}{|P||B_\delta|}\sum_{p\in P}\left|\int_{B_\delta} u\right|}_{3=ZRL}$$

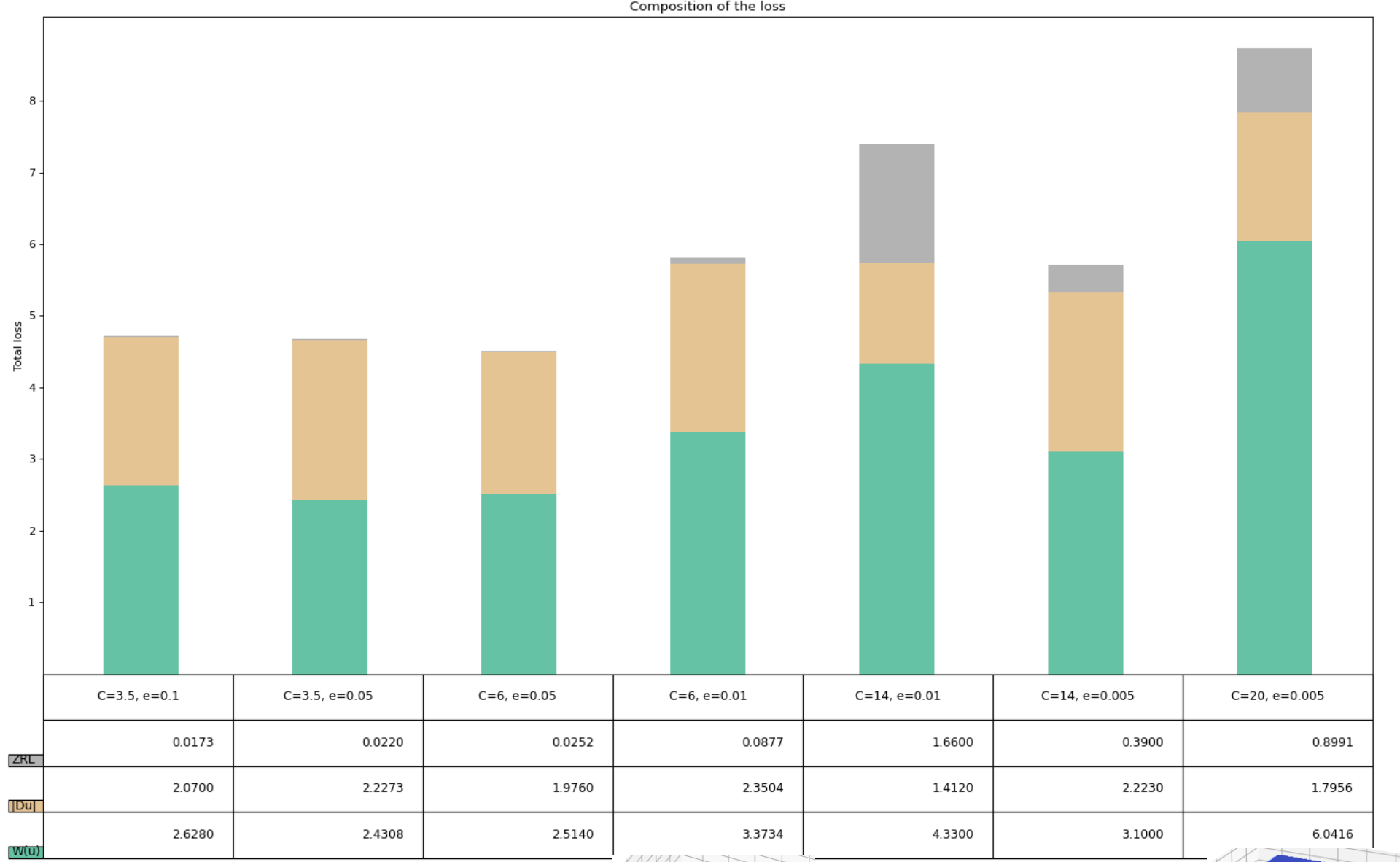
From the paper it follows:

$$\lambda \longrightarrow \infty,\ \lambda\sqrt{\varepsilon} \longrightarrow 0 \qquad (\varepsilon \to 0)$$

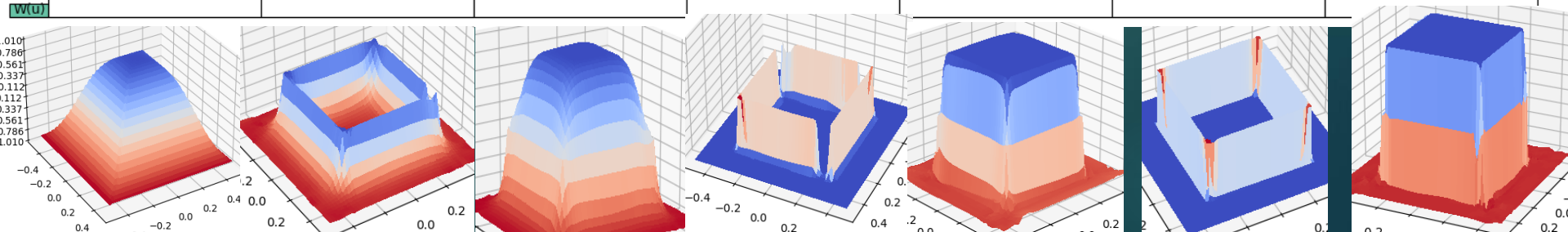
$$\lambda = C\varepsilon^{-\frac{1}{3}}$$

$$W(s) = \tfrac{9}{16}(s^2-1)^2$$

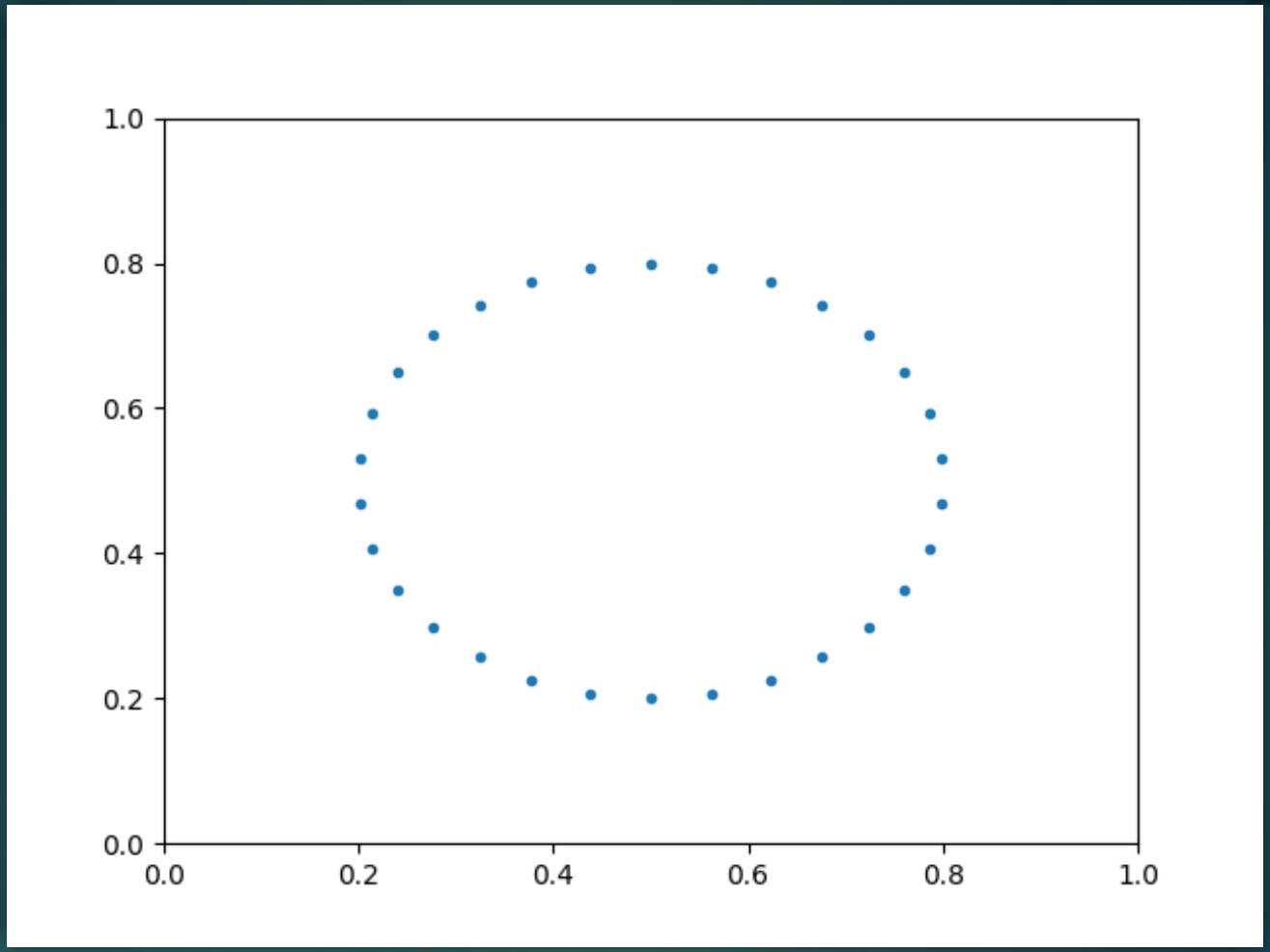

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

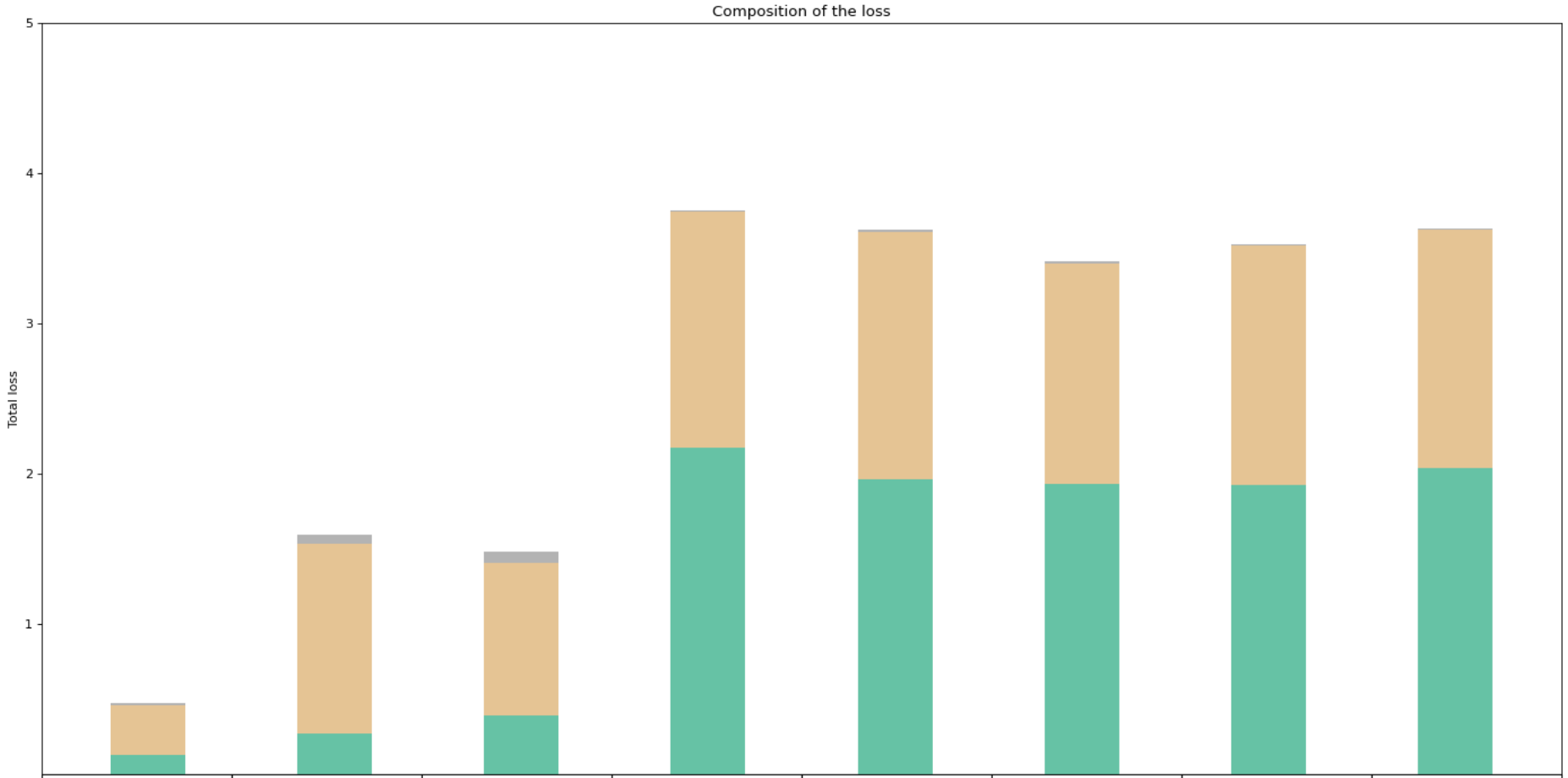


	C=3.5, e=0.1	C=3.5, e=0.05	C=6, e=0.05	C=6, e=0.01	C=14, e=0.01	C=14, e=0.005	C=20, e=0.005
ZRL	0.0173	0.0220	0.0252	0.0877	1.6600	0.3900	0.8991
|Du|	2.0700	2.2273	1.9760	2.3504	1.4120	2.2230	1.7956
W(u)	2.6280	2.4308	2.5140	3.3734	4.3300	3.1000	6.0416

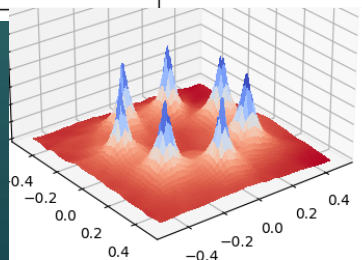


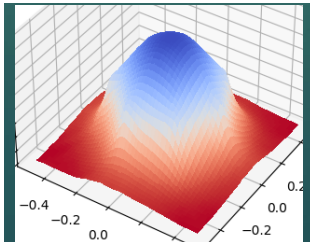
EPS	C
0,1	3,5
0,05	6
0,01	14
0,005	20
0,001	60
0,0005	120
0,0001	240


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

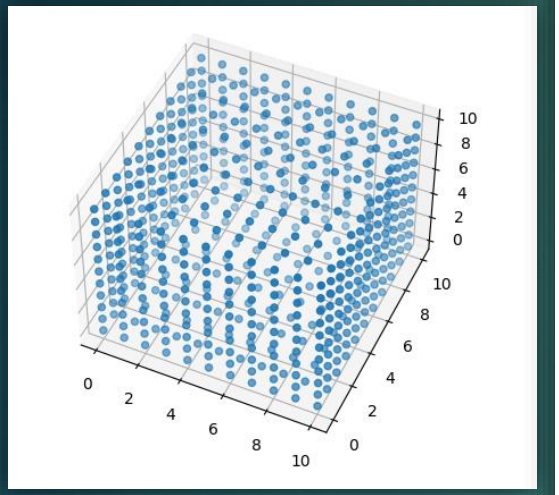


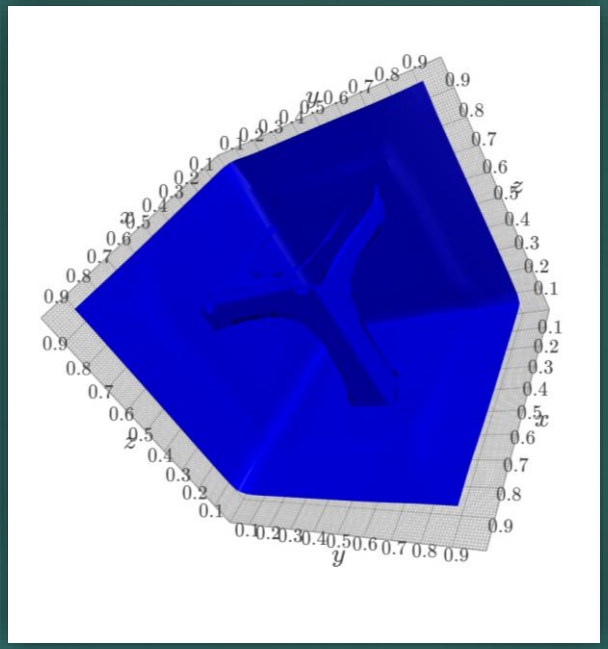
	5	6	7	8	9	25	100	1000
ZRL	0.0149	0.0635	0.0753	0.0117	0.0130	0.0099	0.0083	0.0090
|Du|	0.3345	1.2617	1.0142	1.5680	1.6497	1.4706	1.5970	1.5909
W(u)	0.1220	0.2697	0.3908	2.1744	1.9618	1.9303	1.9201	2.0348



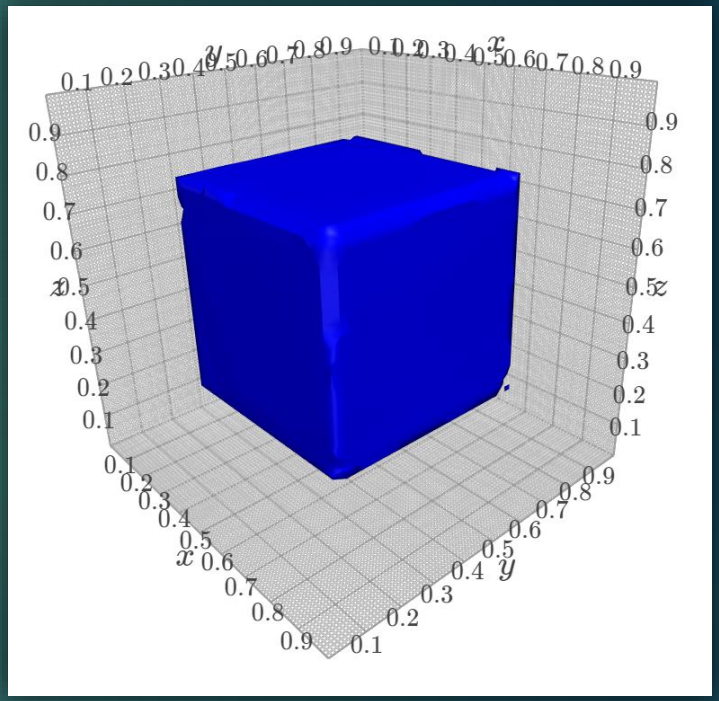


3D Beispiele

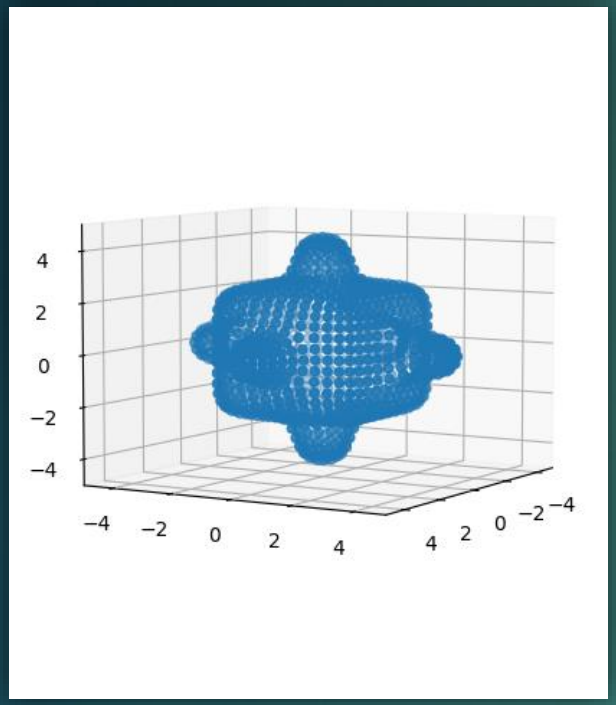


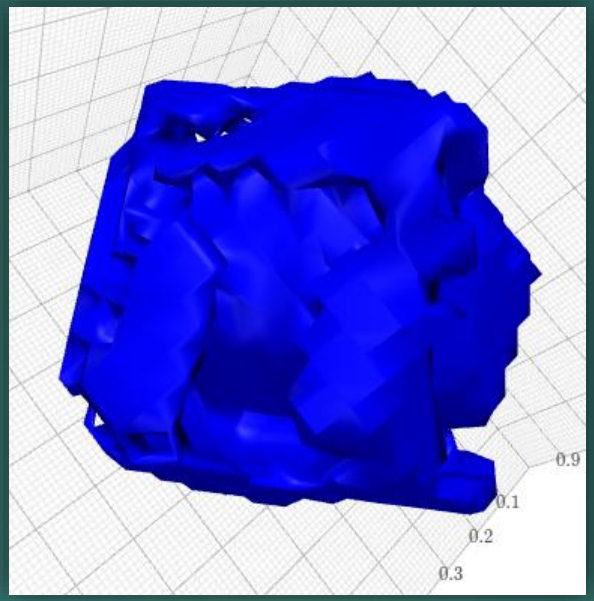


Ohne FF

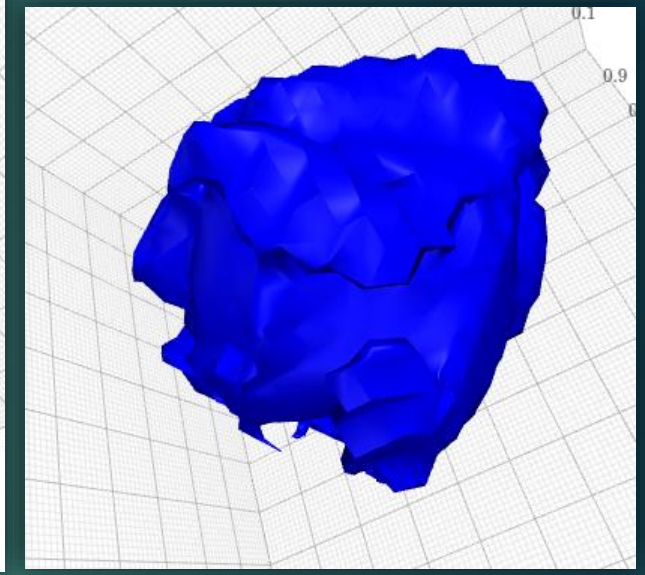


Mit FF

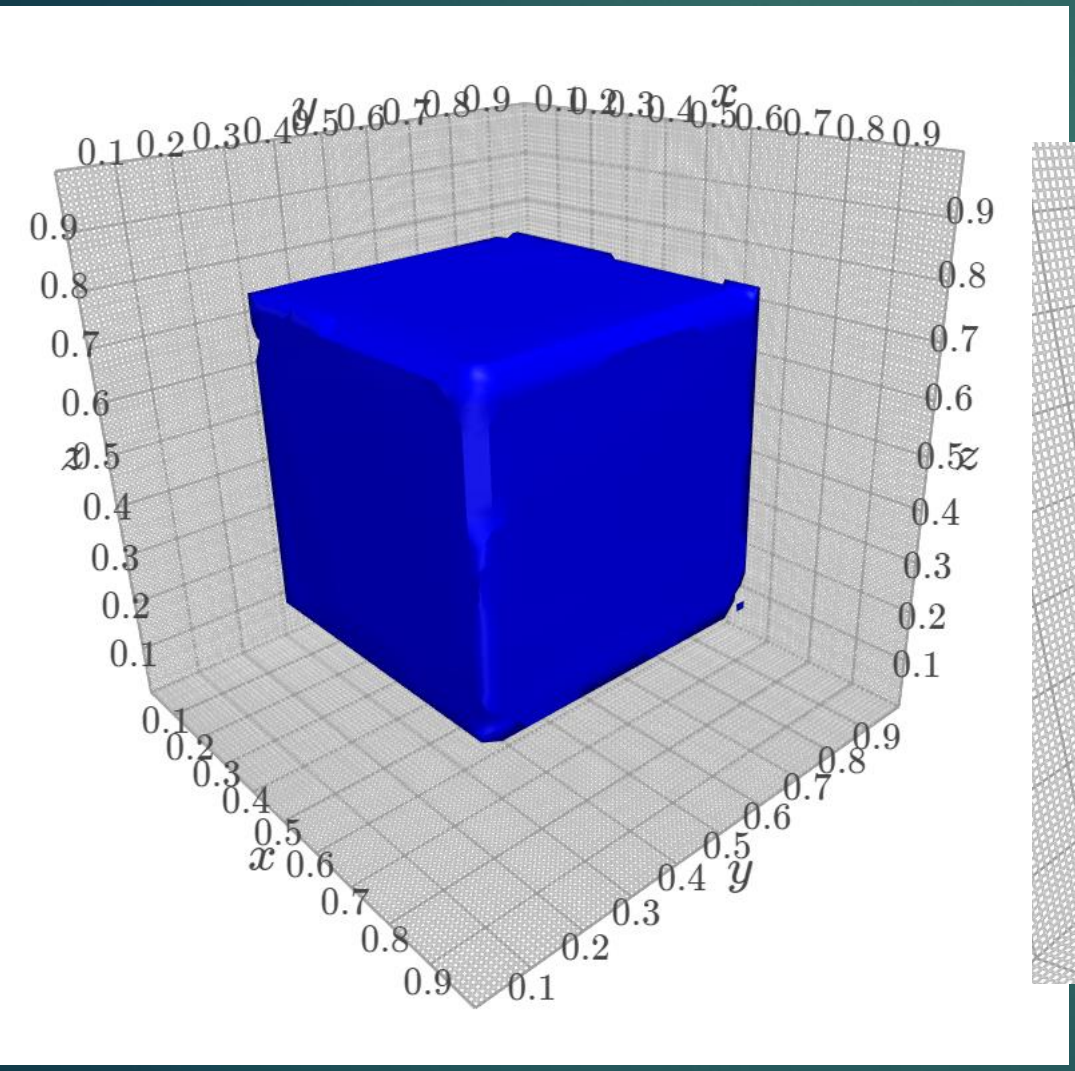


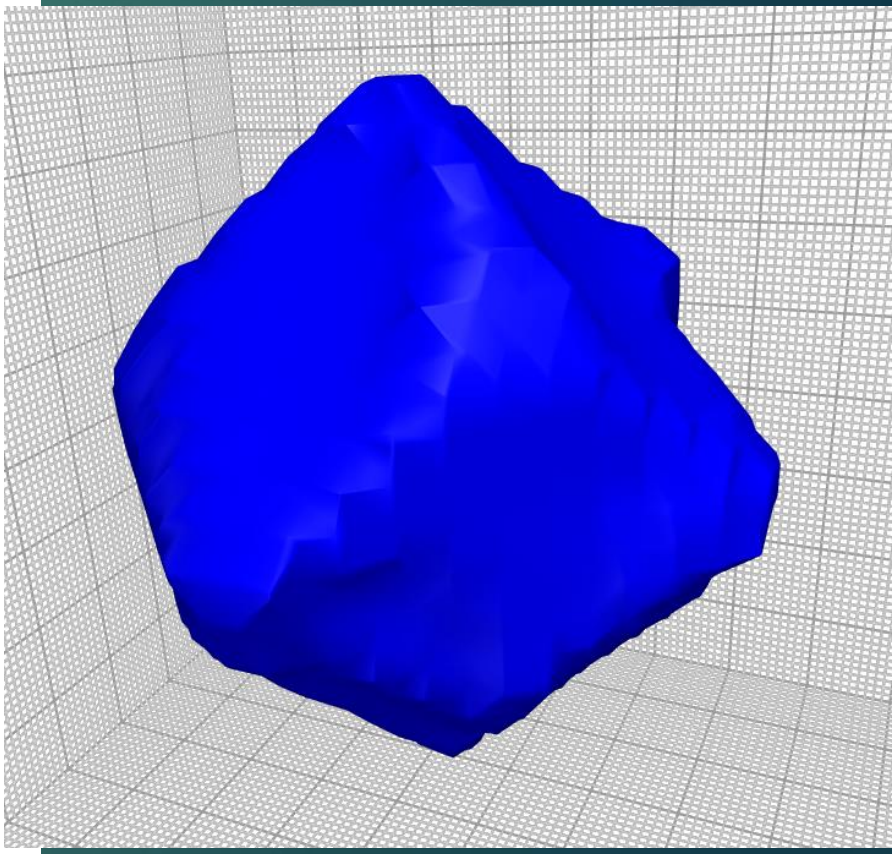


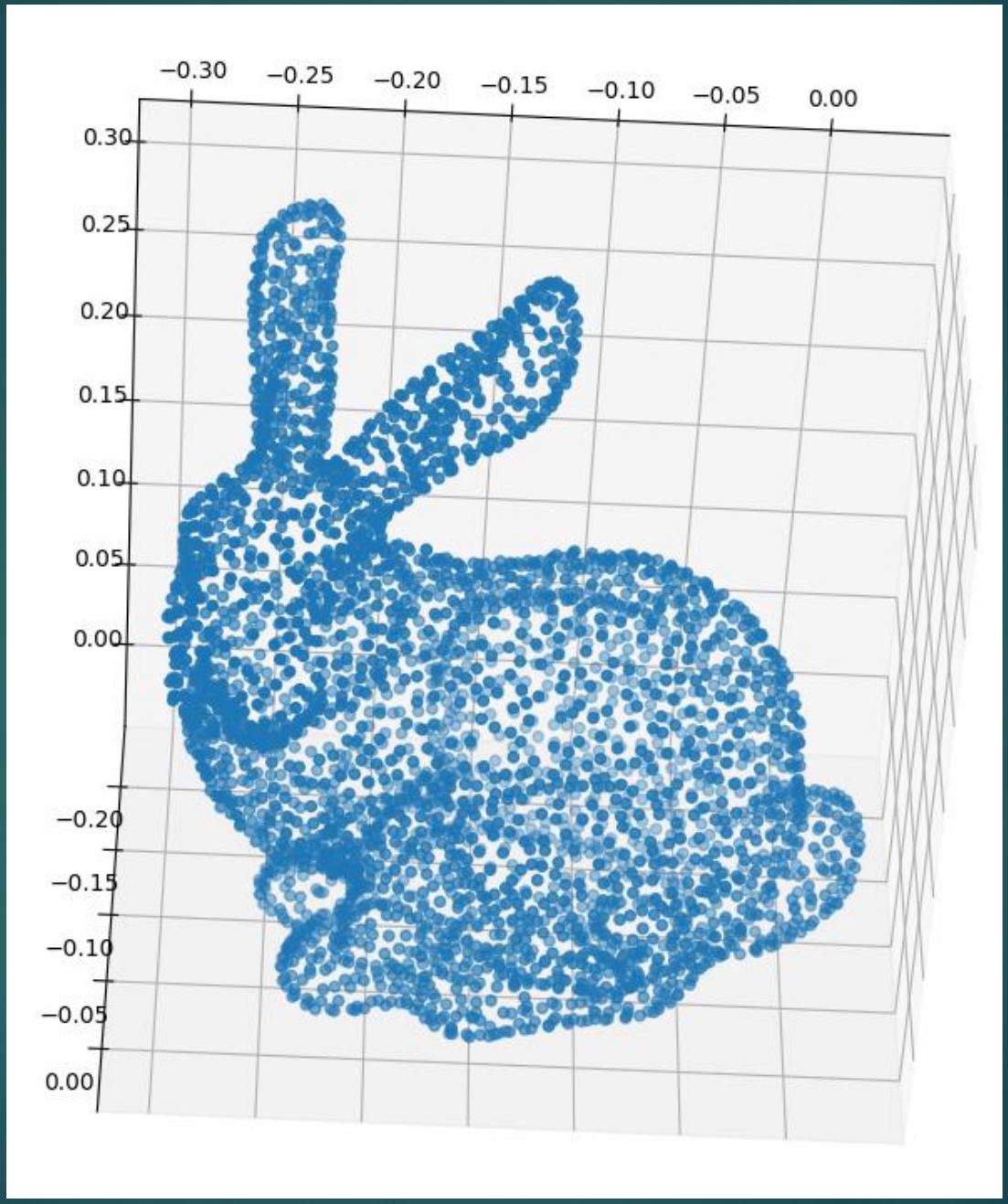
Ohne FF



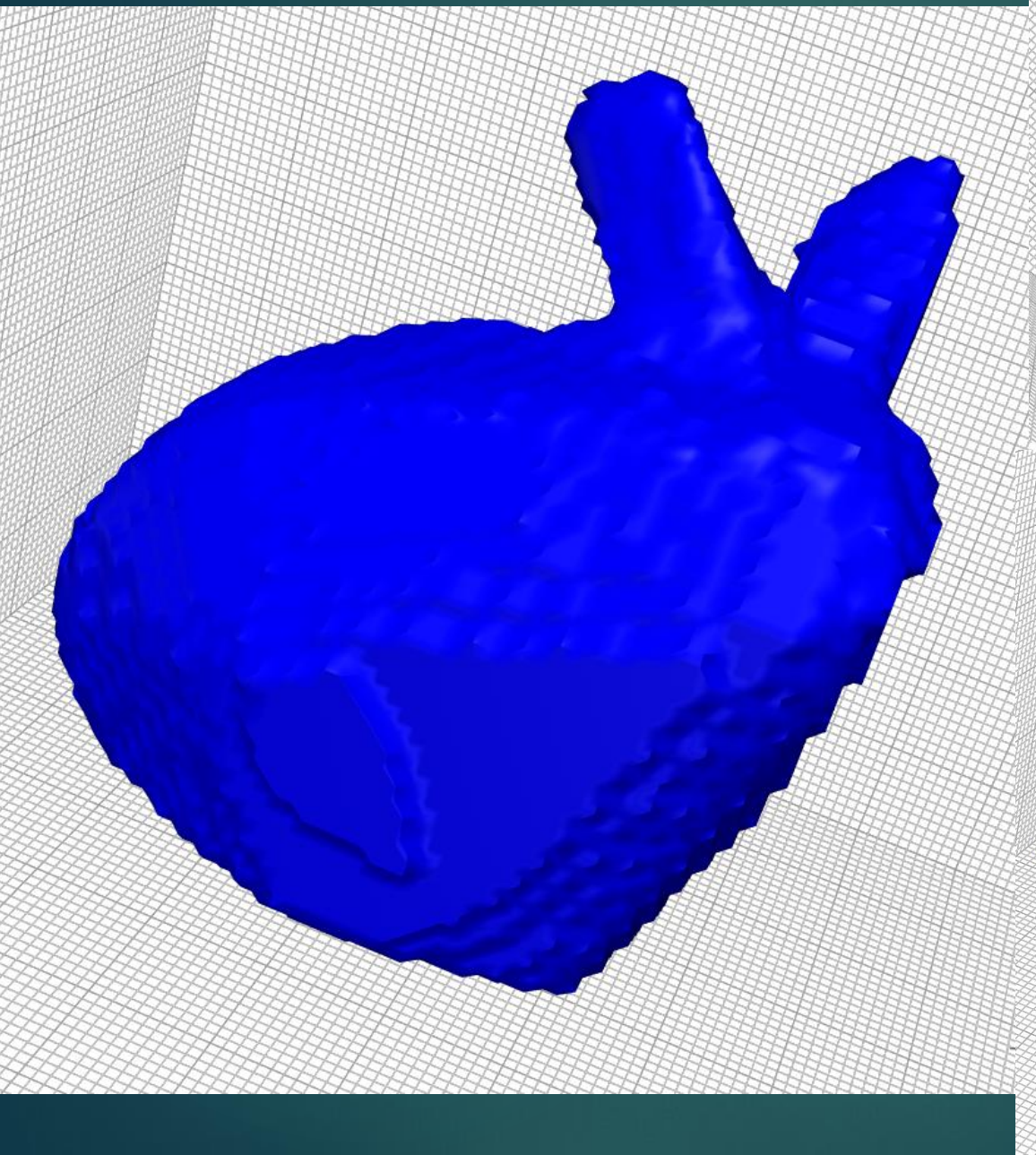
Mit FF

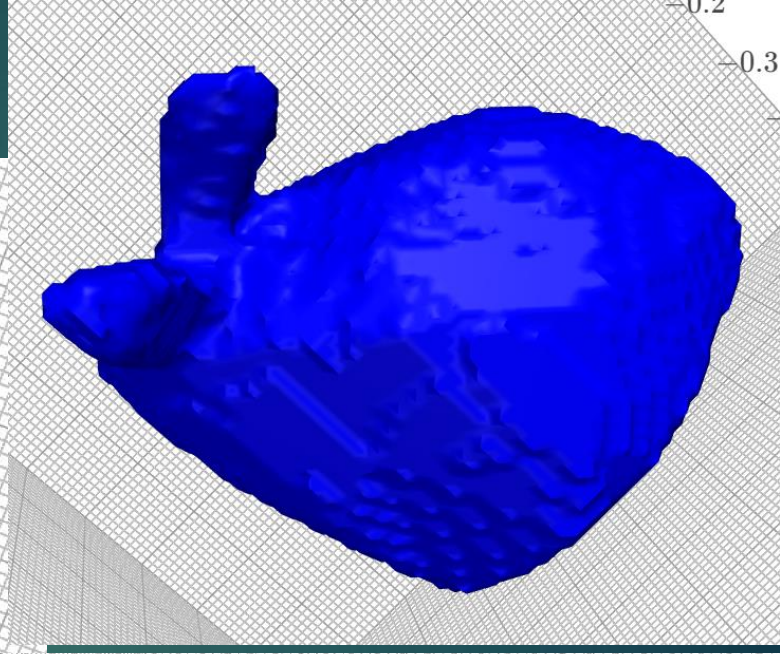

x
y
z
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9

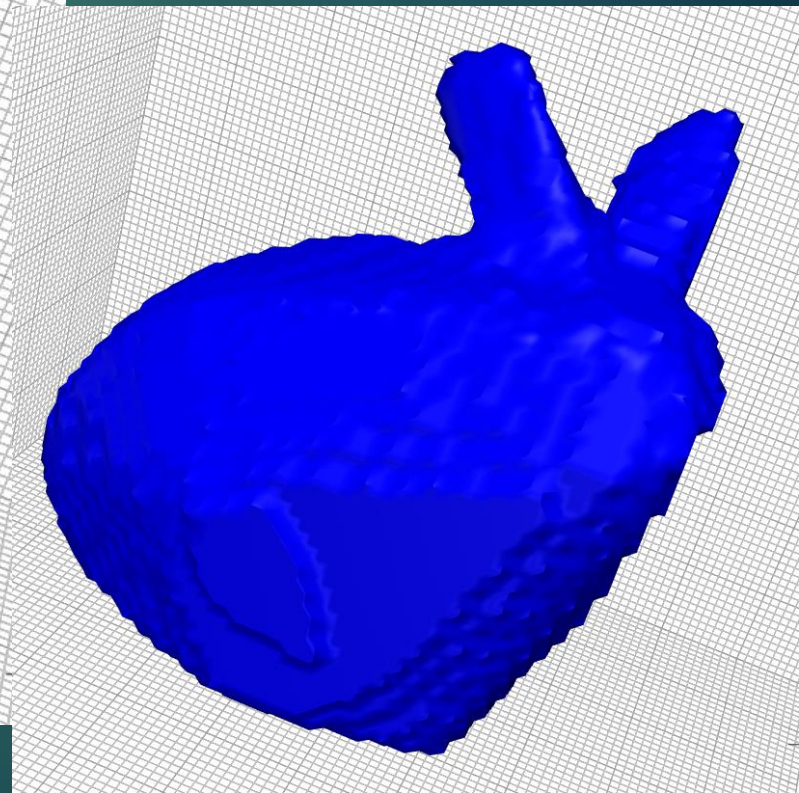



−0.30
−0.25
−0.20
−0.15
−0.10
−0.05
0.00
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−0.20
−0.15
−0.10
−0.05
0.00

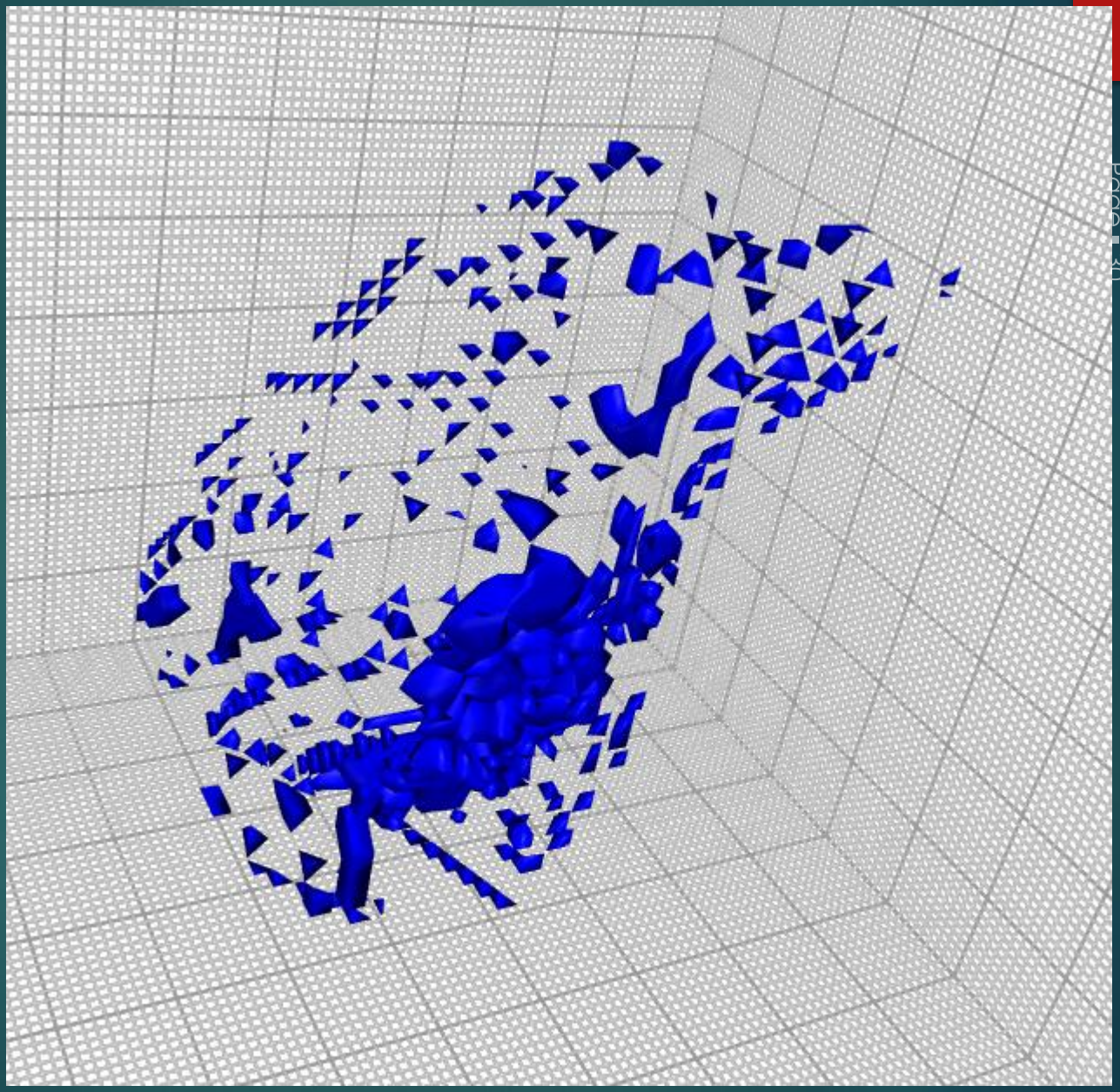
Eps = 0,0001, sigma =0,3

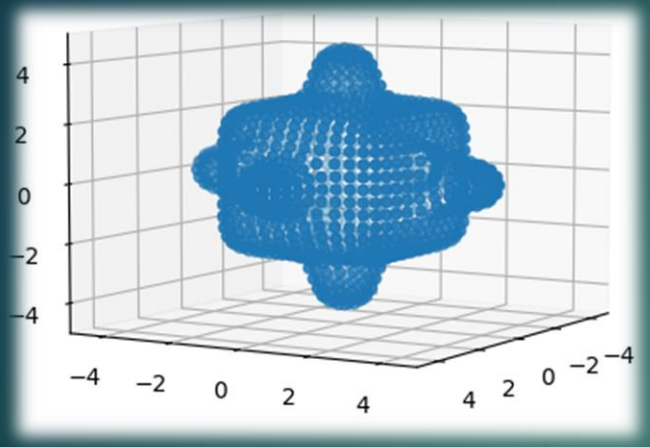


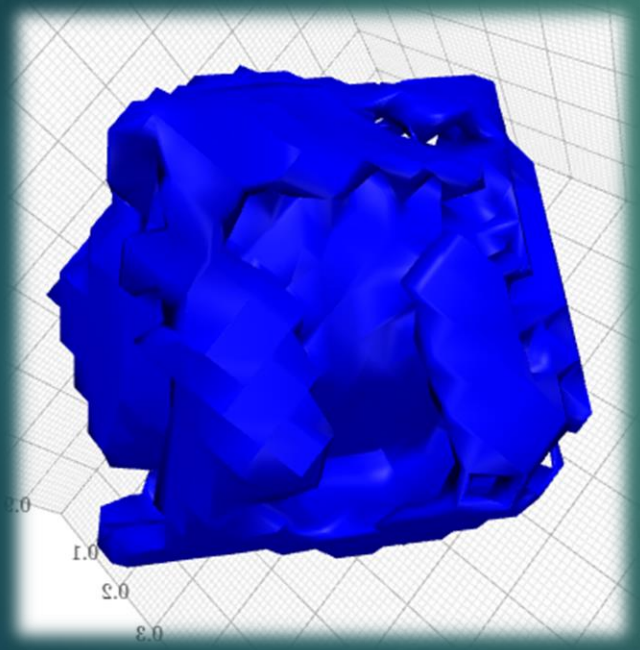


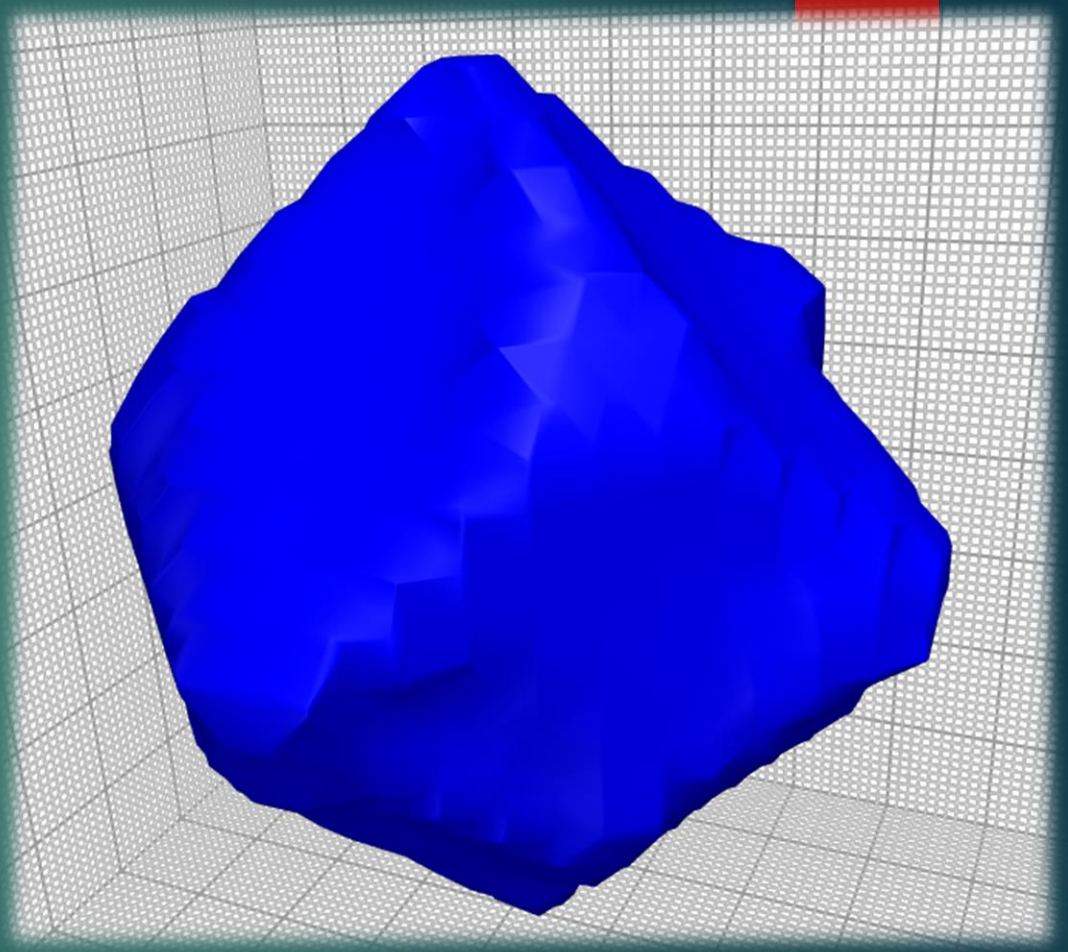


Eps = 0,00001, sigma =0,3




4
2
0
−2
−4
−4
−2
0
2
4
4
2
0
−2
−4





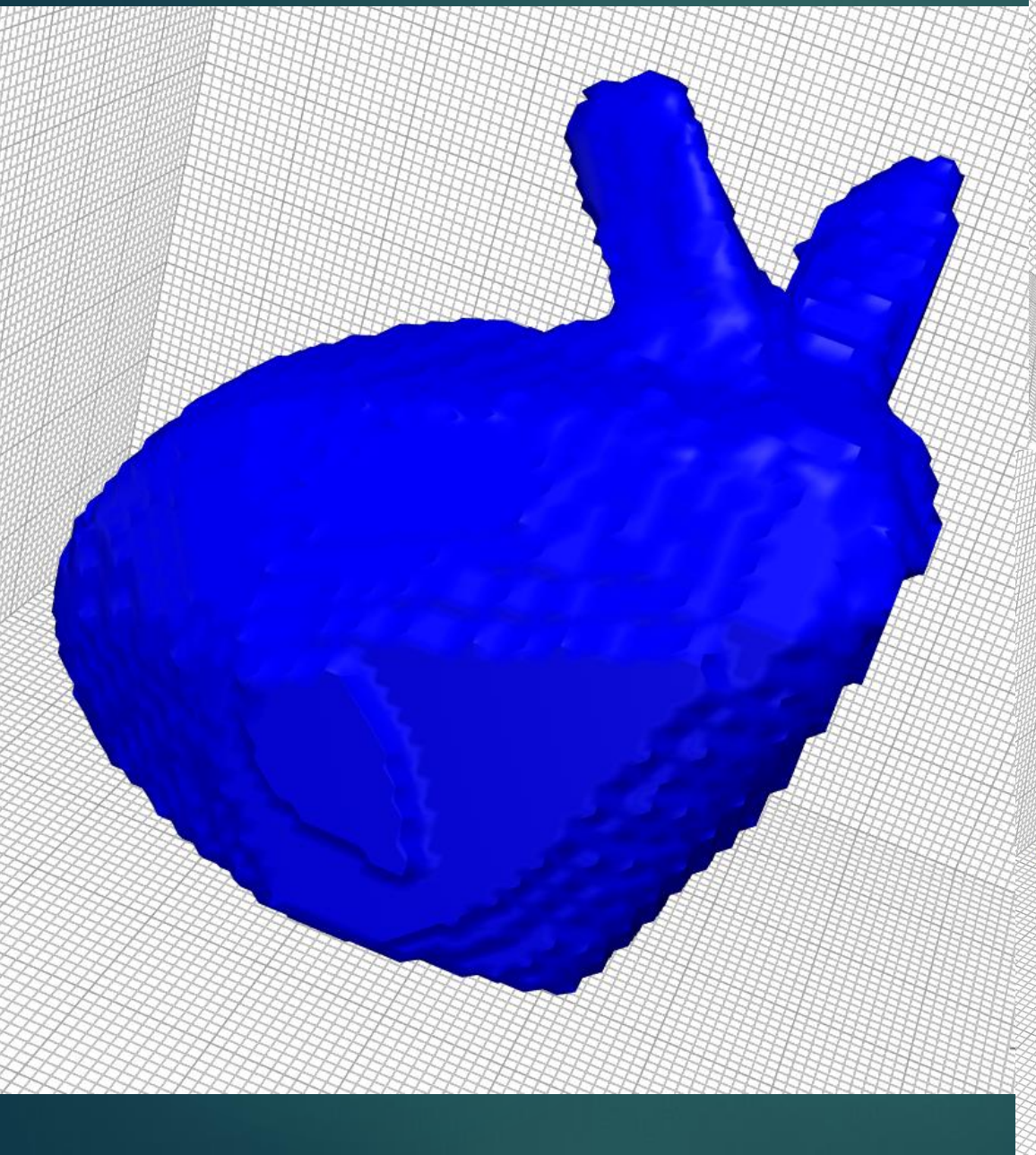
Modica Mortola:

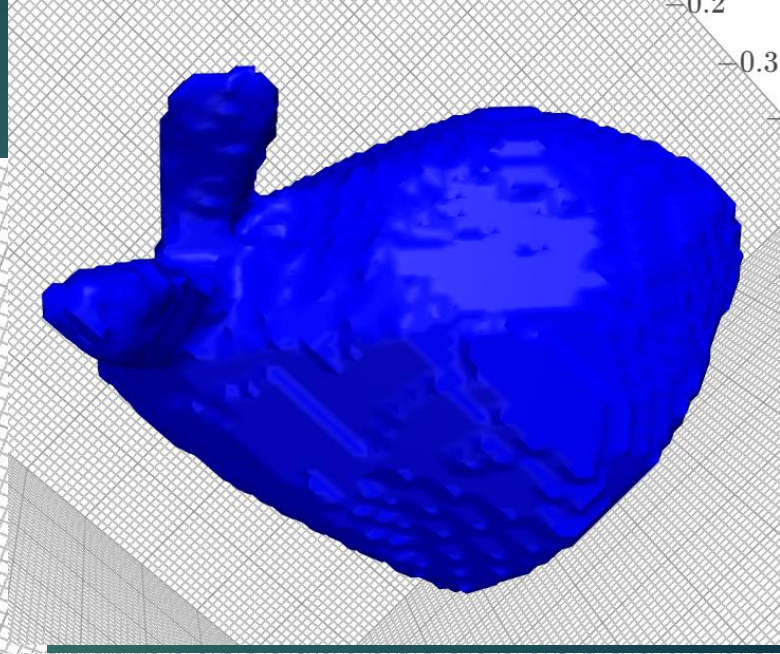
- AT Löst MM?

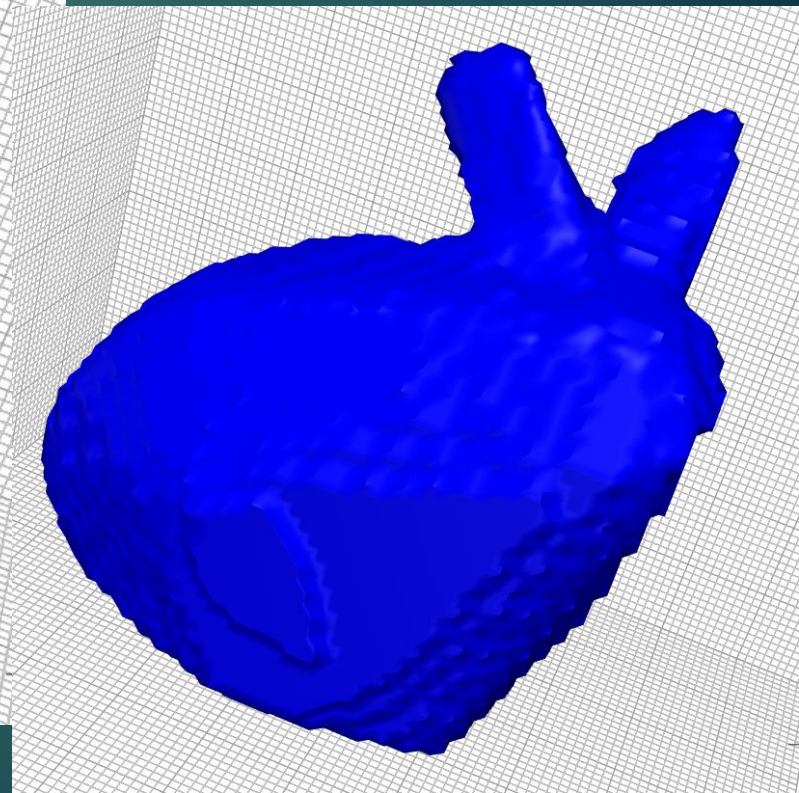
$$\int_\Omega W(u) + \varepsilon |\nabla u|^2 \dots$$

- Nabla u?
- Bei AT?

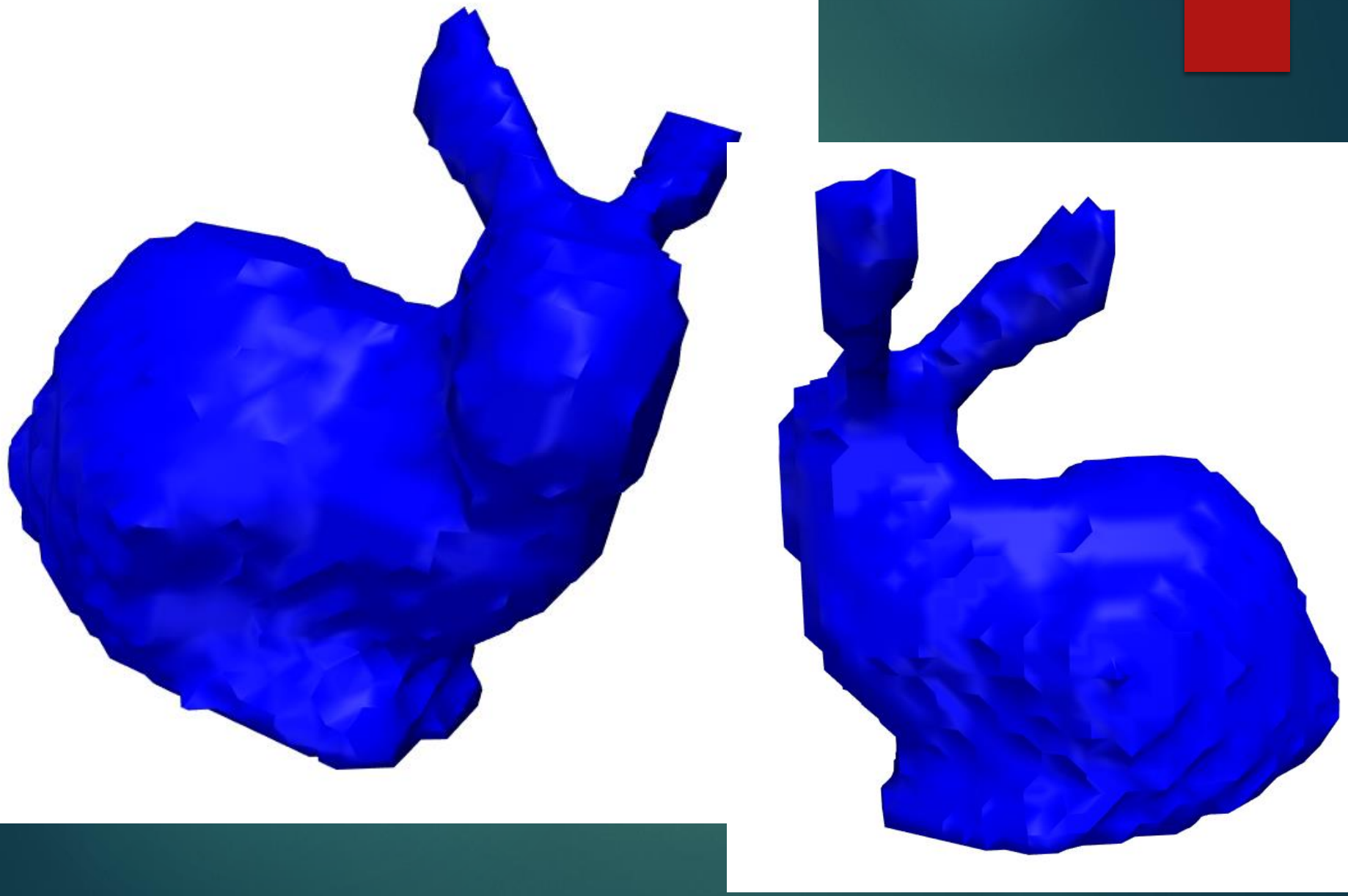
Eps = 0,0001, sigma =0,3



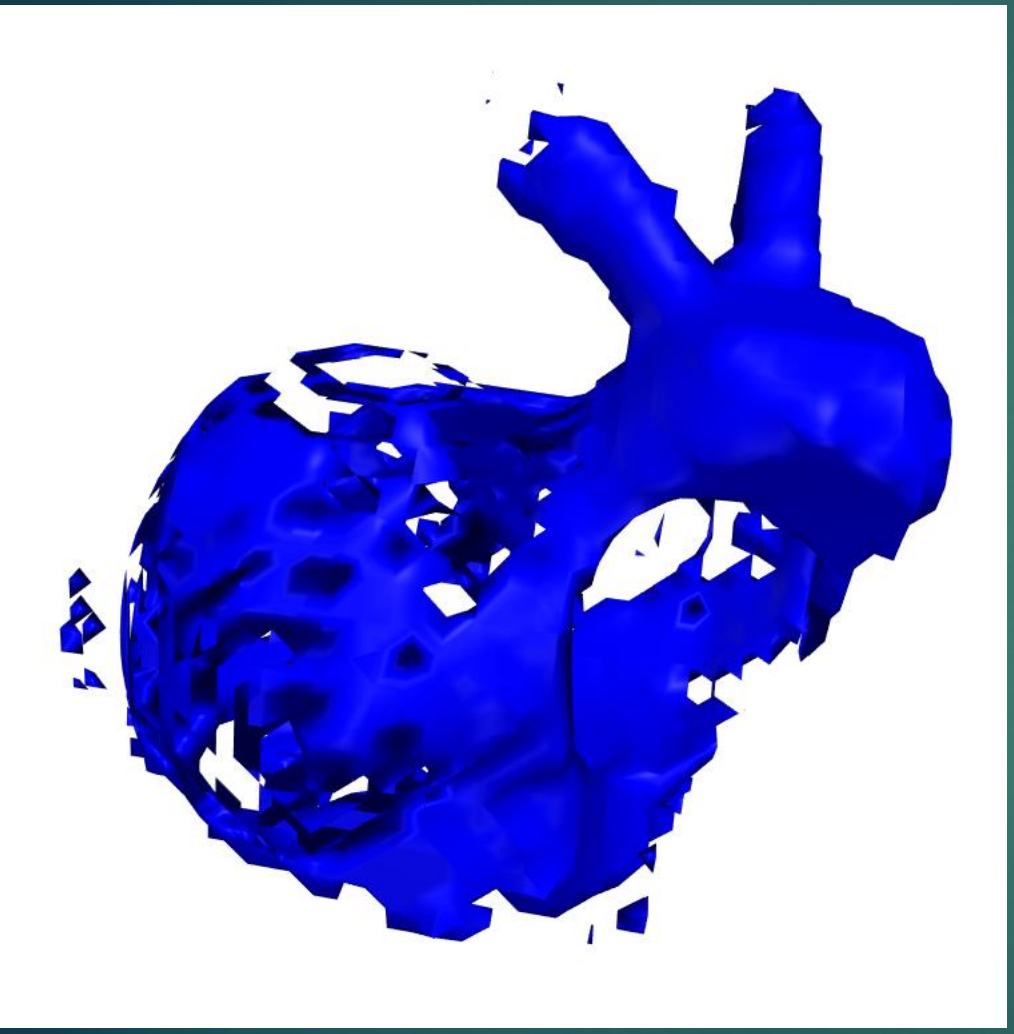




Eps = 0.0001, sigma =1.3

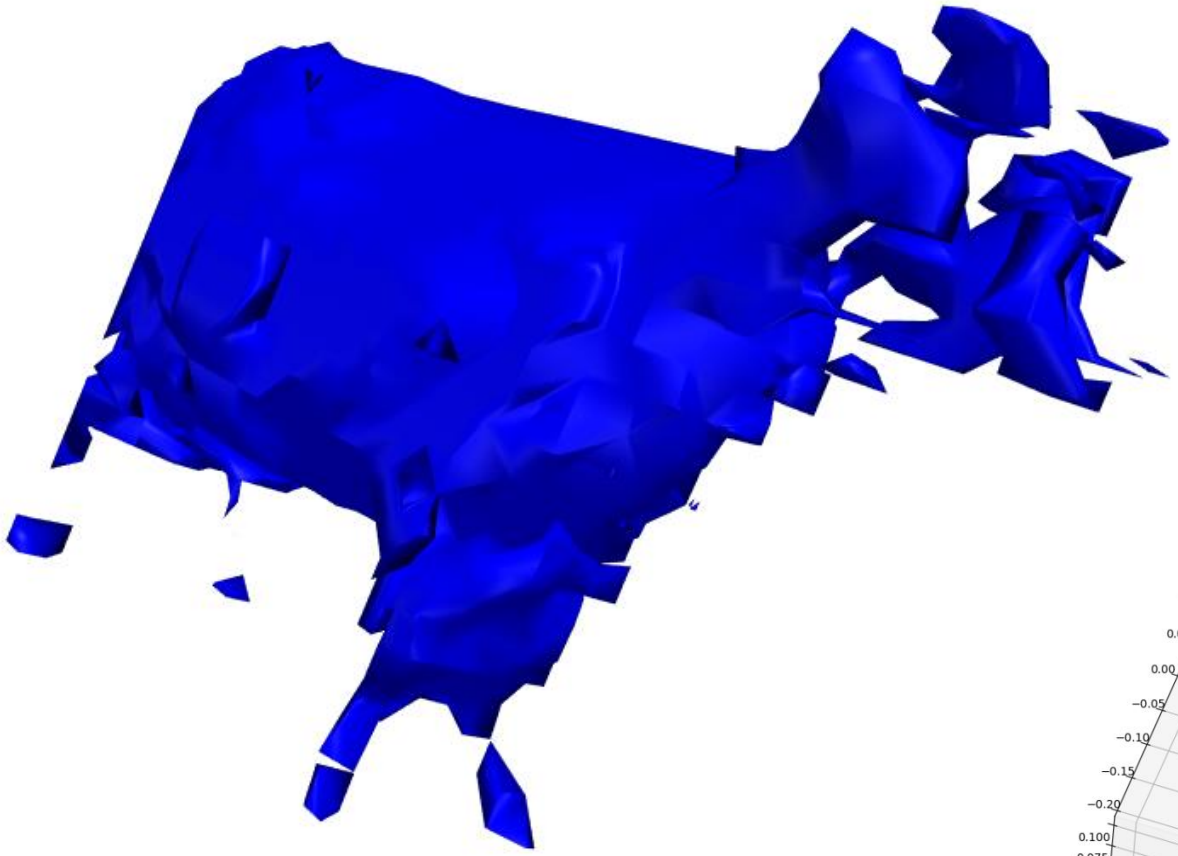


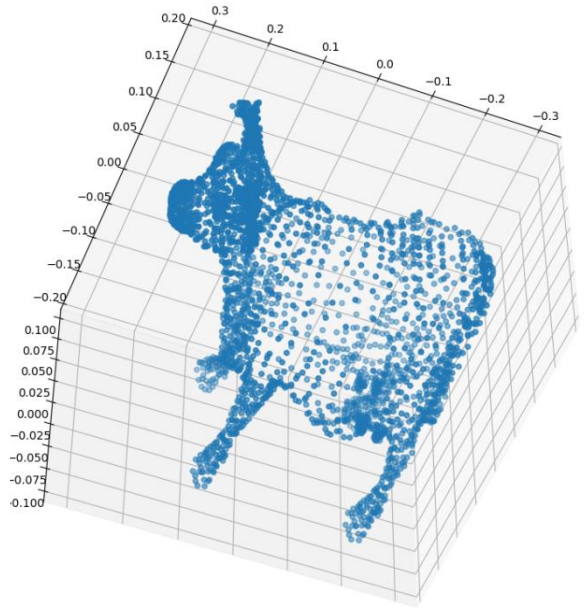
Eps = 0,0001, sigma =1,7

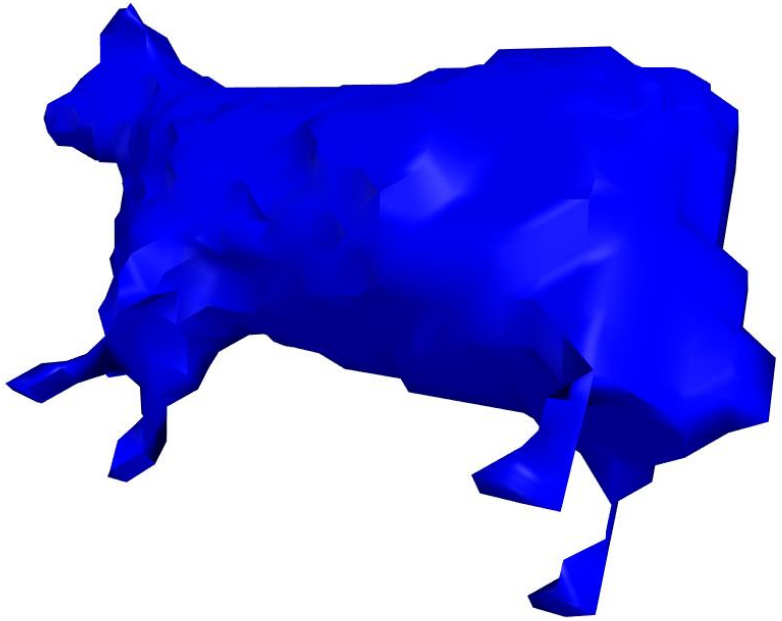


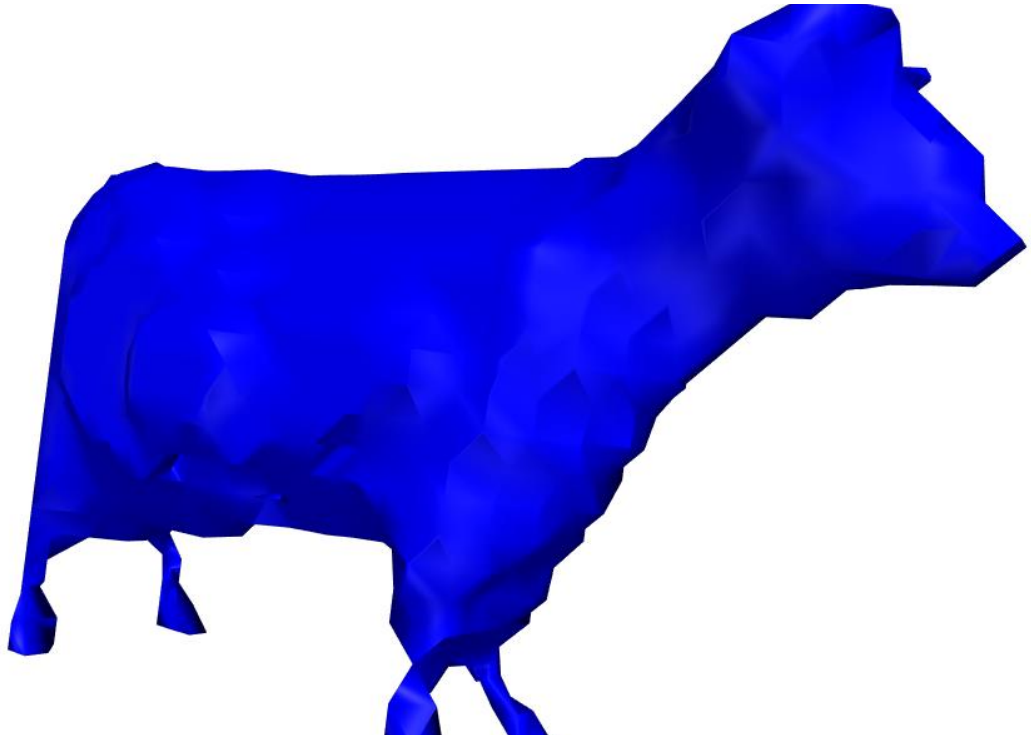
C=0

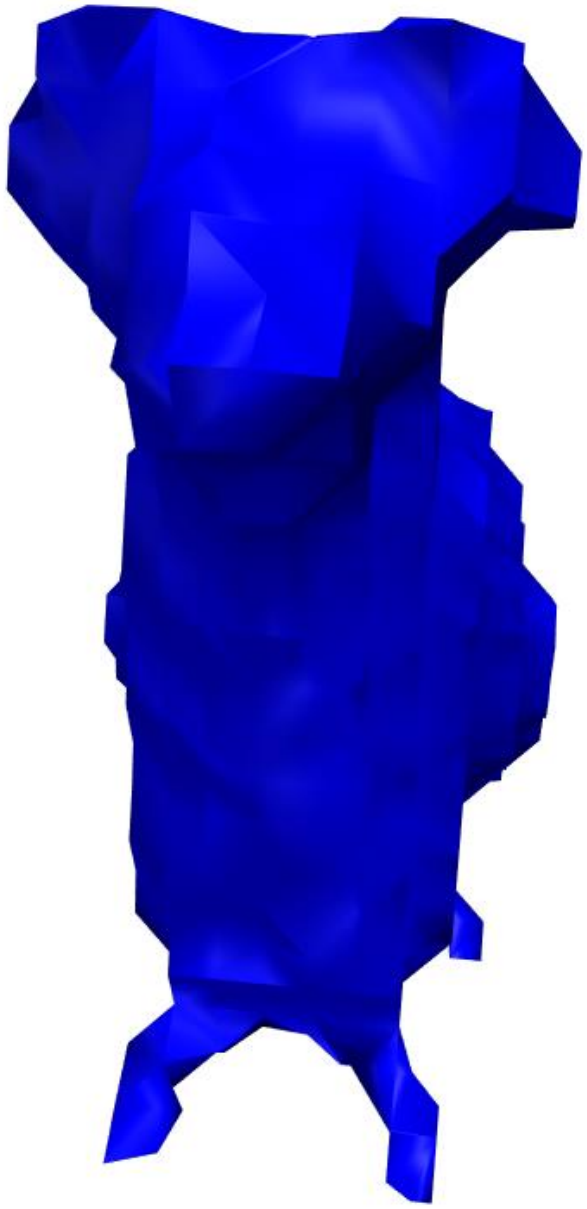
C=0,




0.20
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.15
−0.20
0.100
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3

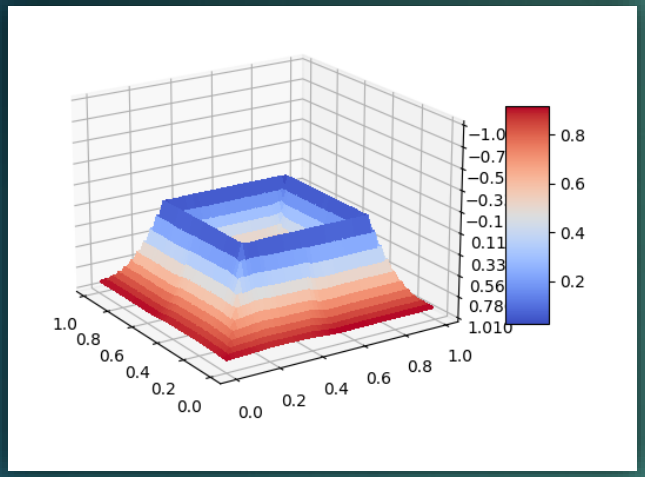




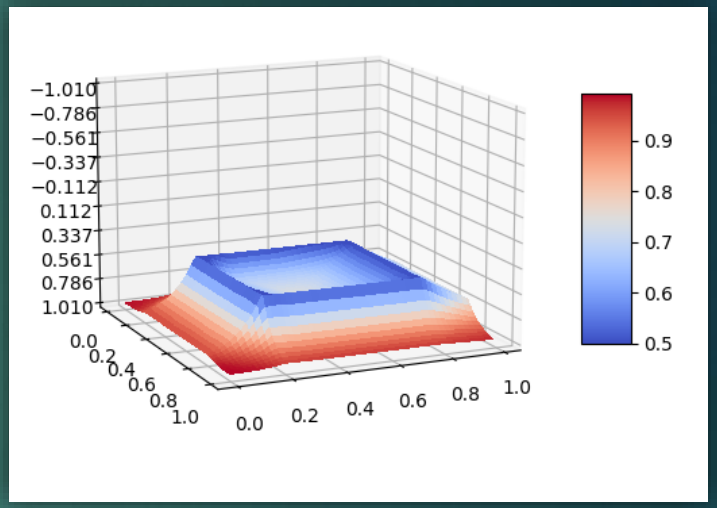


Ambrosio Tortorelli

Zero reconstruction loss

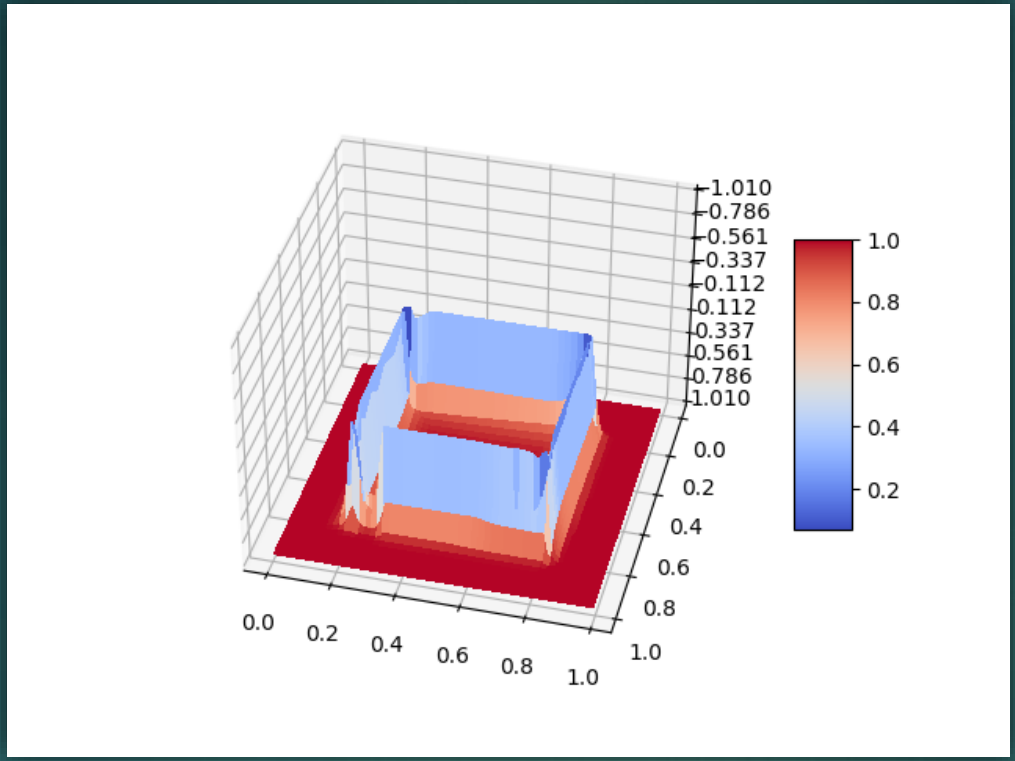


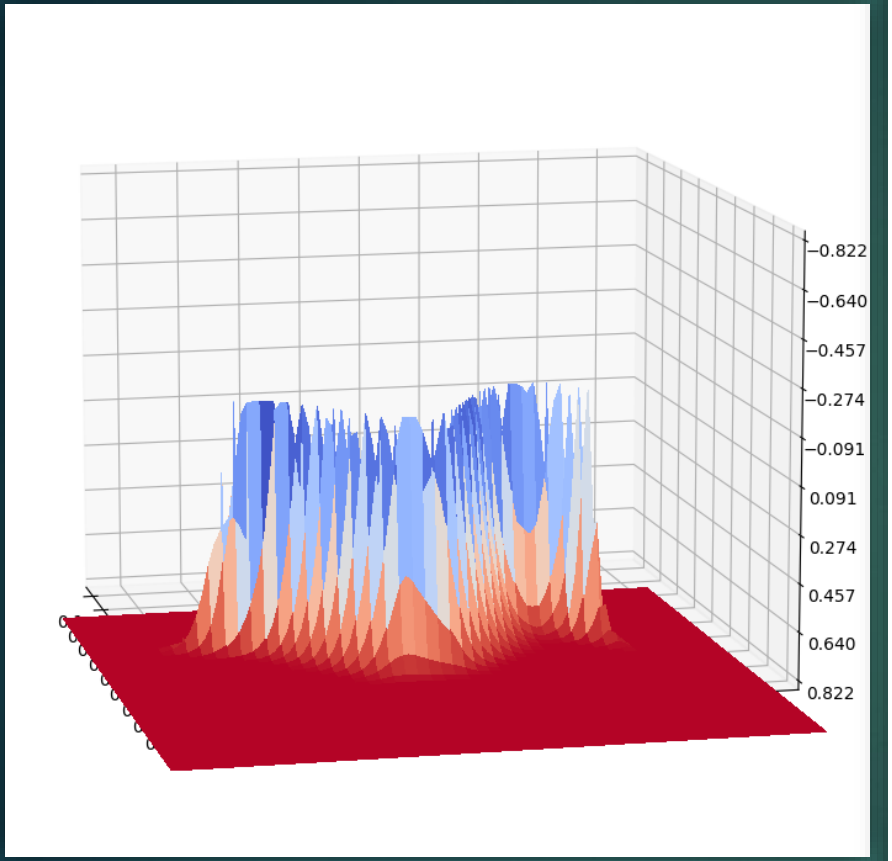
$$\frac{1}{|\mathcal{P}|}\sum_{p\in P}|u(p)|$$

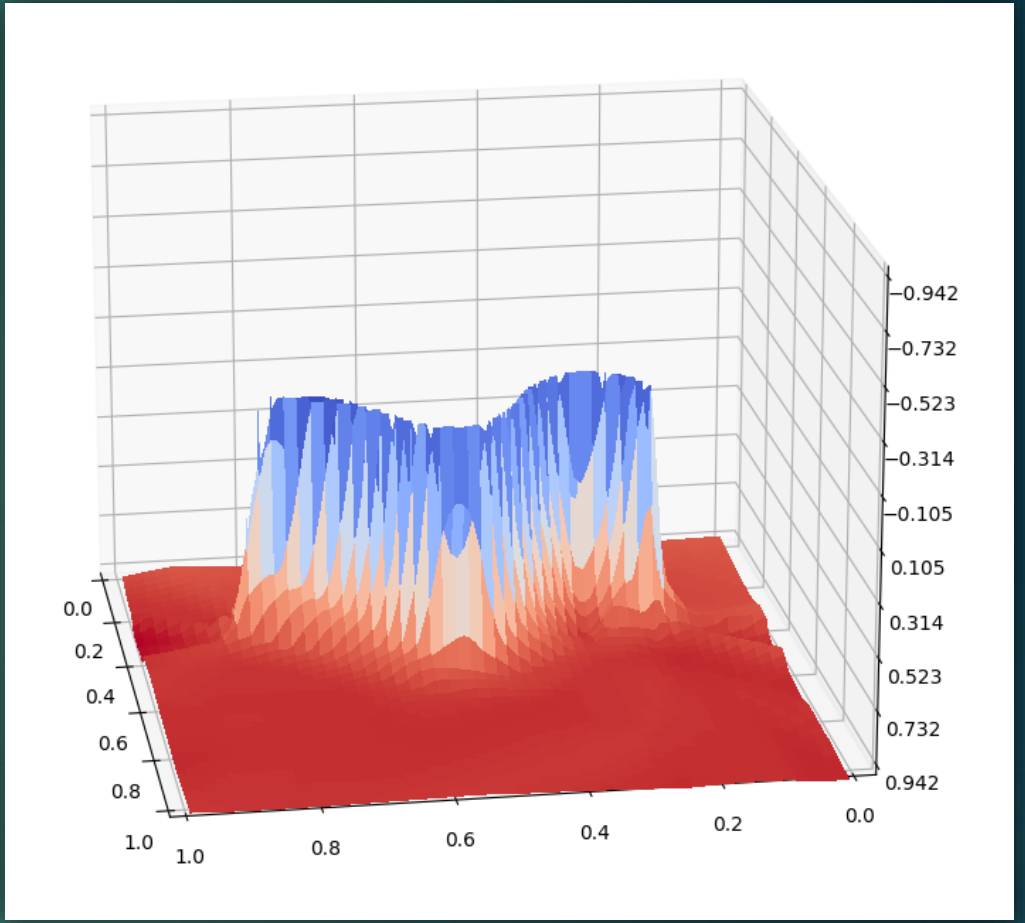


$$\sum_{p\in P}\left|\int_{B_\delta(p)} g(u(x))dx\right|$$

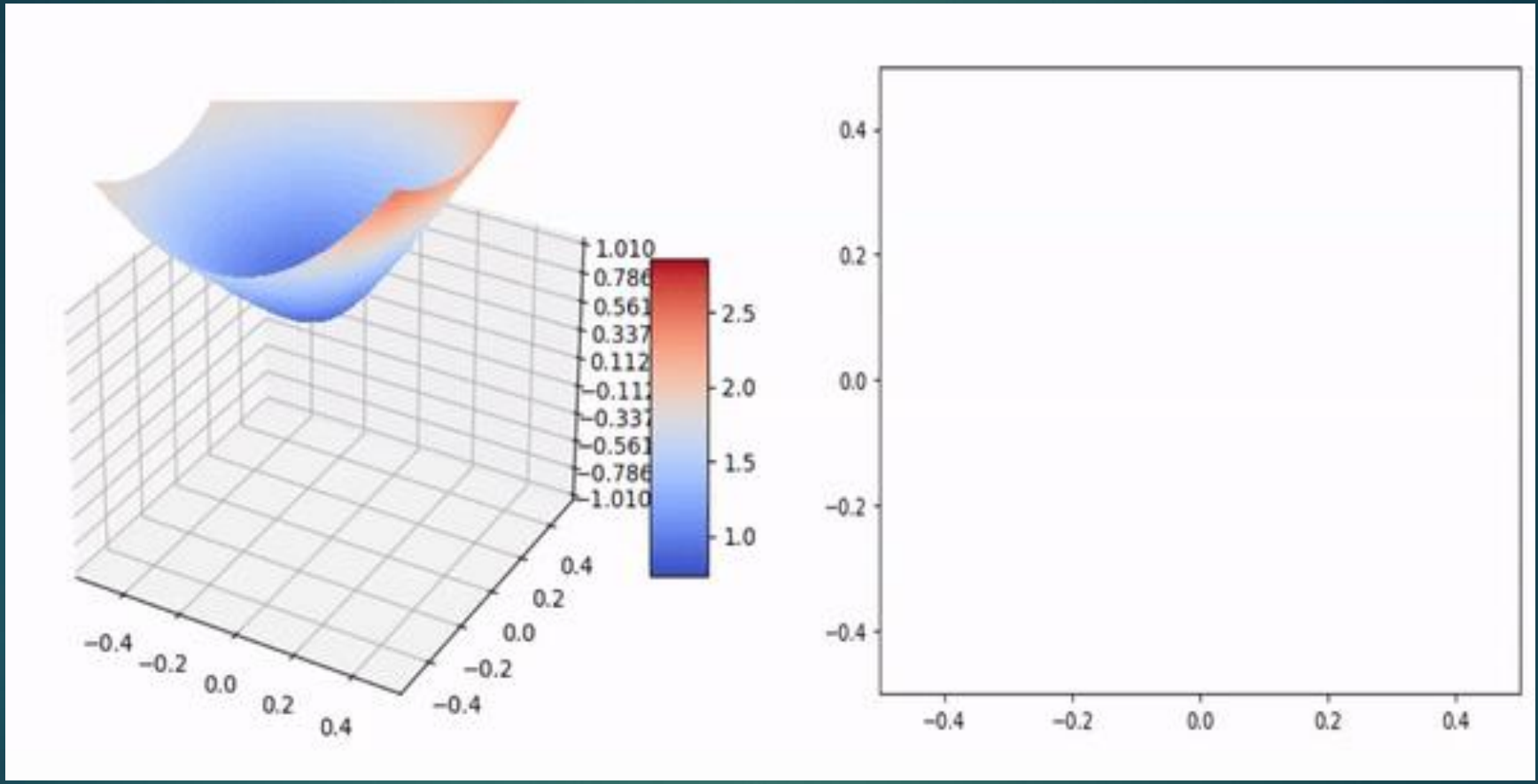
g(s)= 2s-1

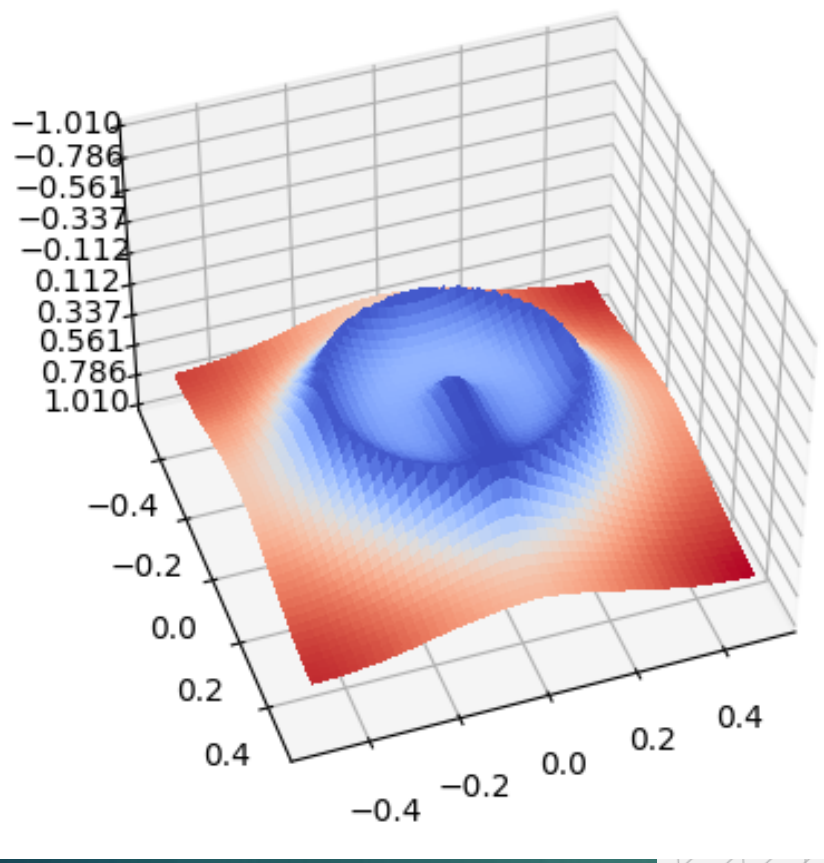

1.010
0.786
0.561
0.337
0.112
0.112
0.337
0.561
0.786
1.010
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0
0.2
0.4
0.6
0.8
1.0


−0.822
−0.640
−0.457
−0.274
−0.091
0.091
0.274
0.457
0.640
0.822

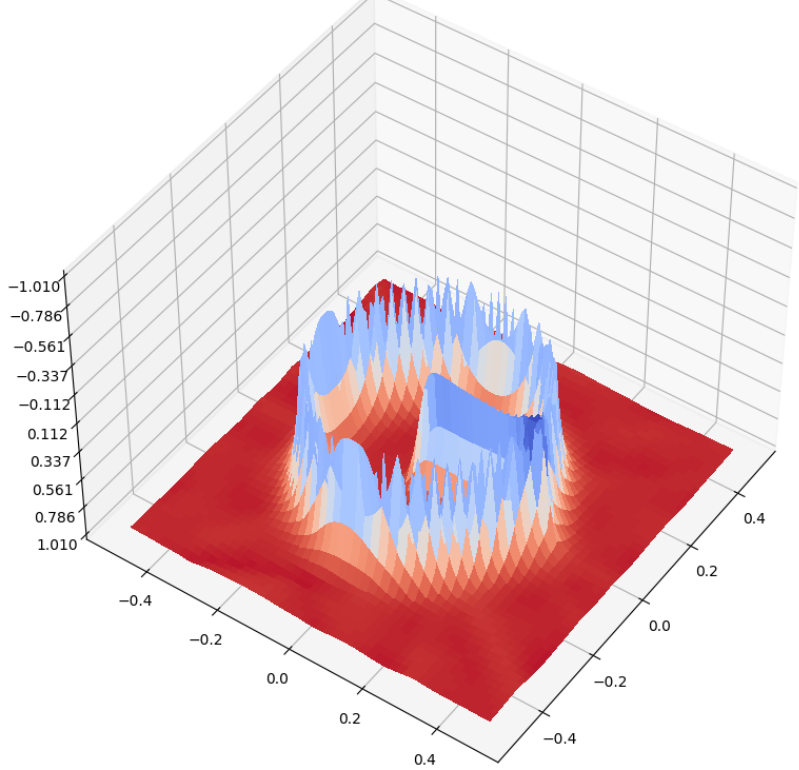

−0.942
−0.732
−0.523
−0.314
−0.105
0.105
0.314
0.523
0.732
0.942
0.0
0.2
0.4
0.6
0.8
1.0
1.0
0.8
0.6
0.4
0.2
0.0

FF

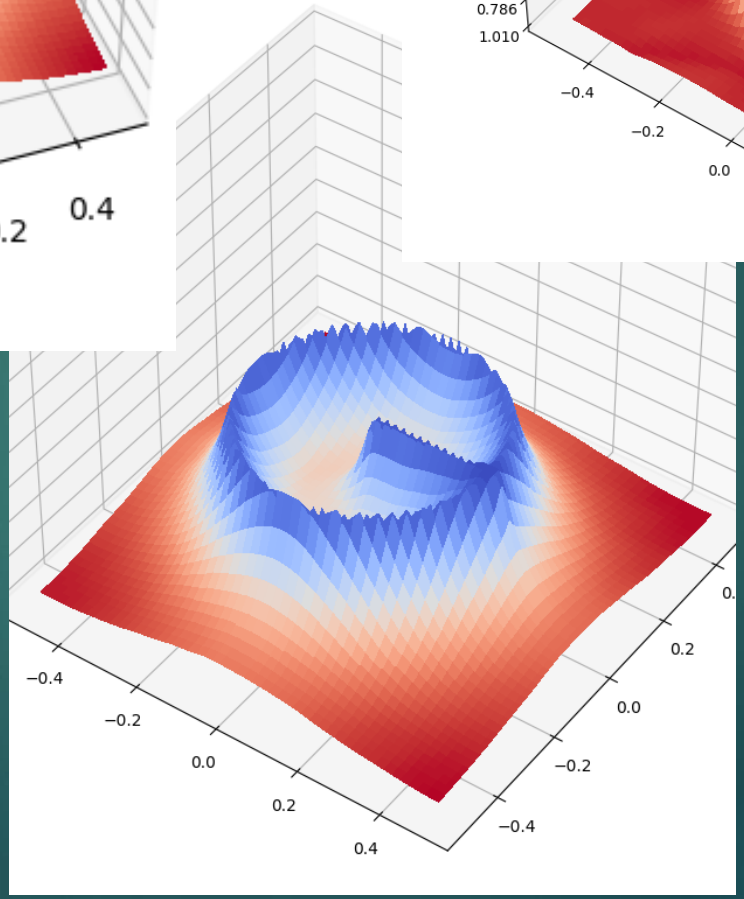

1.010
0.786
0.561
0.337
0.112
−0.112
−0.337
−0.561
−0.786
−1.010
2.5
2.0
1.5
1.0
−0.4
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4


−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
−0.4
−0.2
0.0
0.2
0.4
−0.4
−0.2
0.0
0.2
0.4

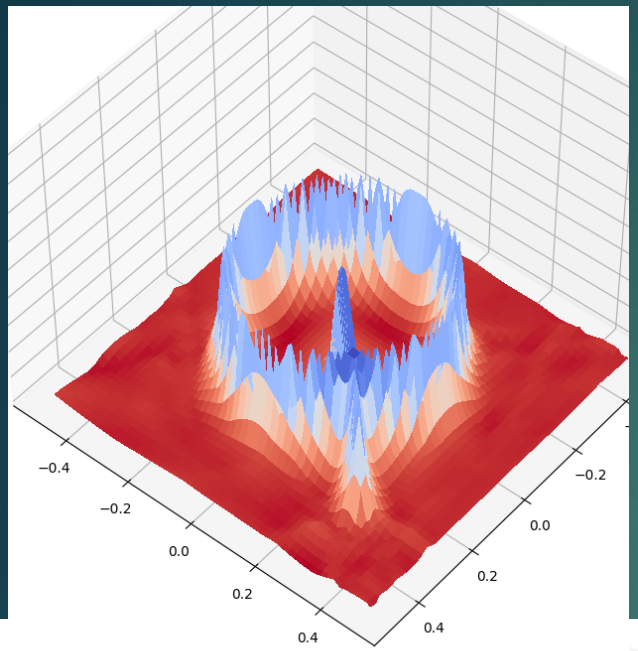
0,1

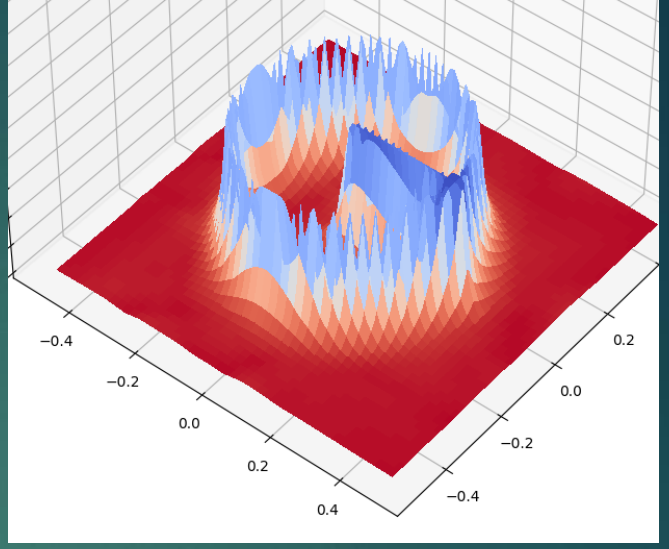

−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
−0.4
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4

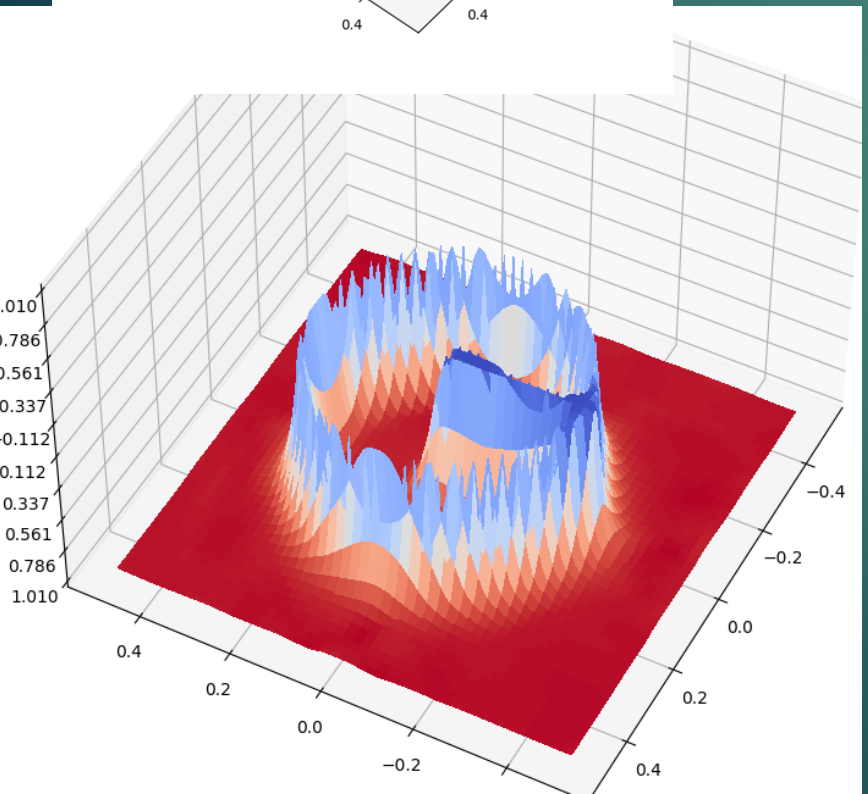
0,01

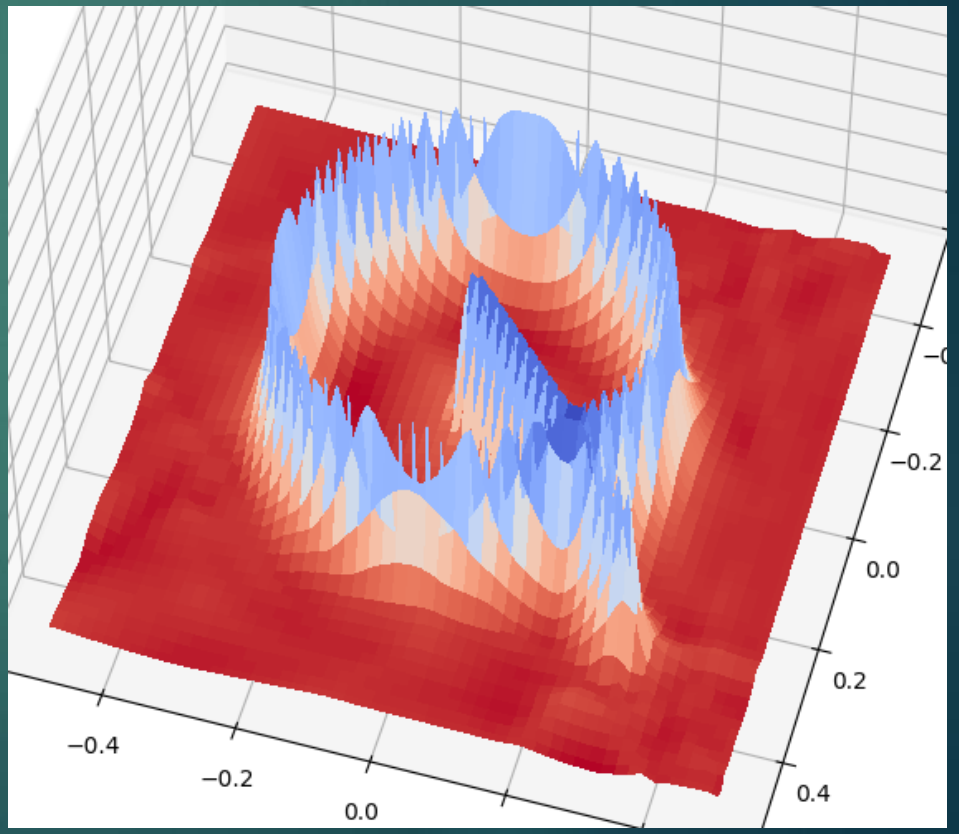

−0.4
−0.2
0.0
0.2
0.4
0.2
0.0
−0.2
−0.4

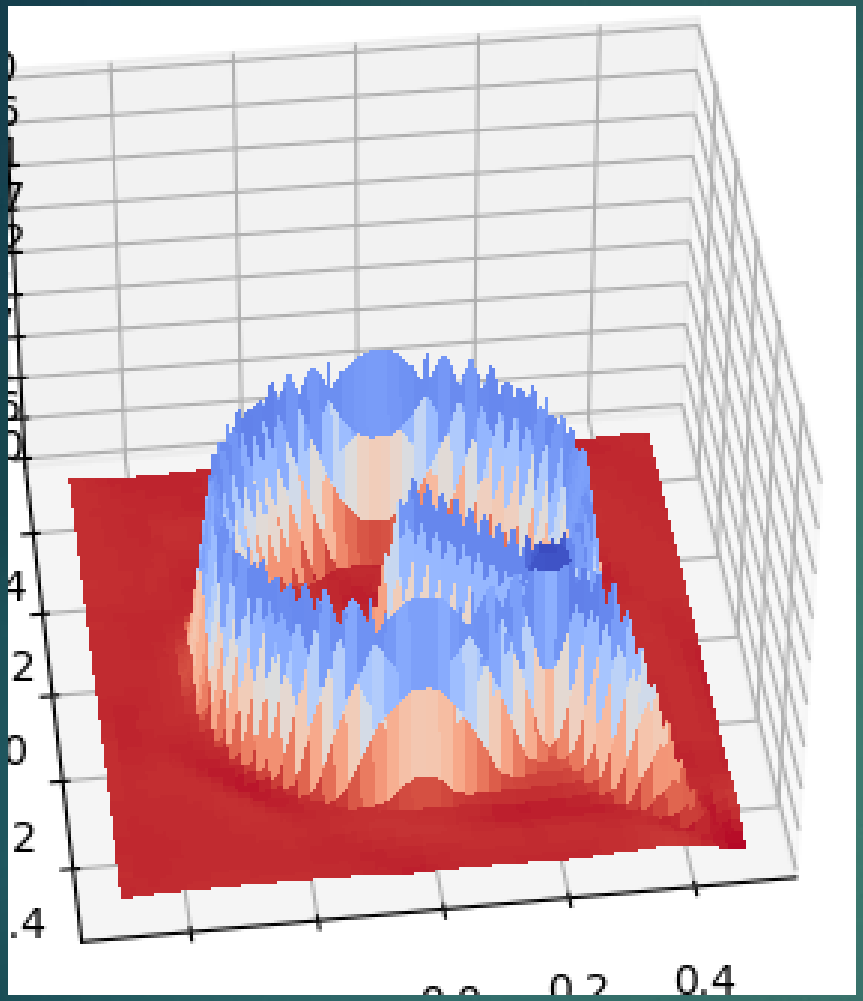
0,05

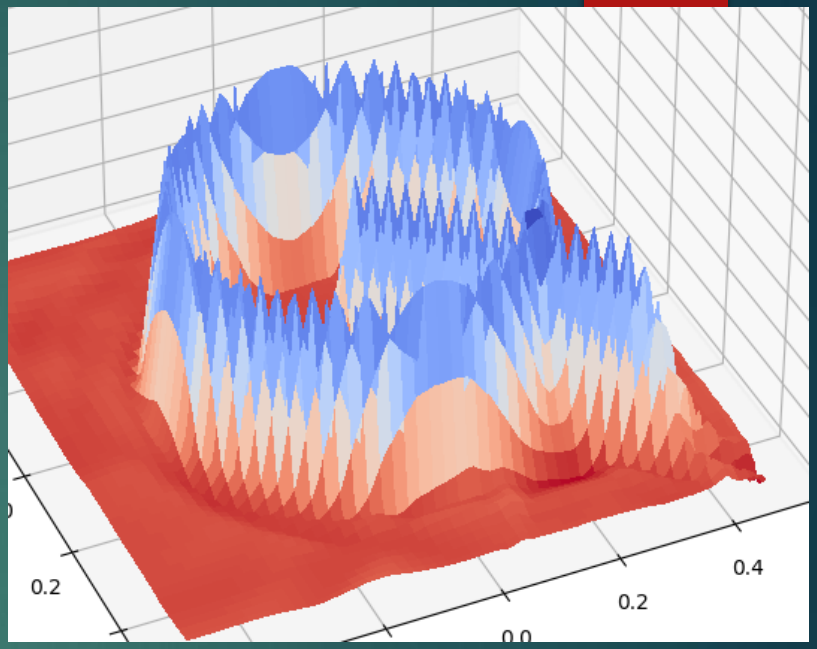

−0.4
−0.2
0.0
0.2
0.4
−0.2
0.0
0.2
0.4


−0.4
−0.2
0.0
0.2
0.4
0.2
0.0
−0.2
−0.4

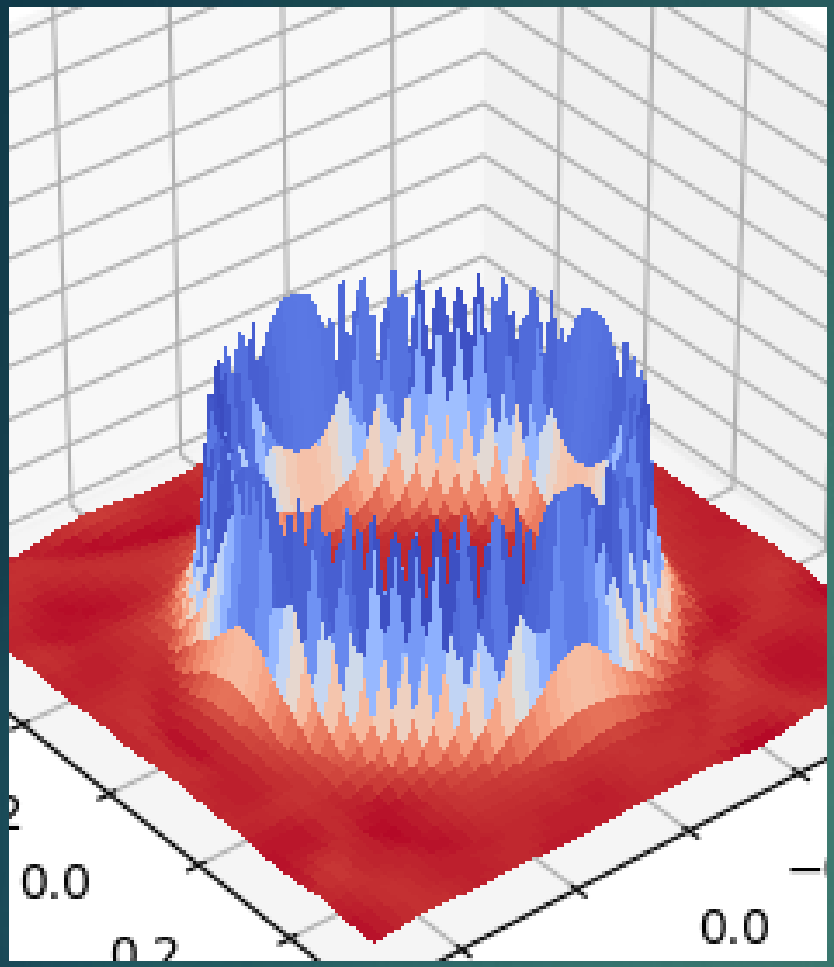

1.010
0.786
0.561
0.337
0.112
0.112
0.337
0.561
0.786
1.010
0.4
0.2
0.0
−0.2
−0.4
−0.2
0.0
0.2
0.4

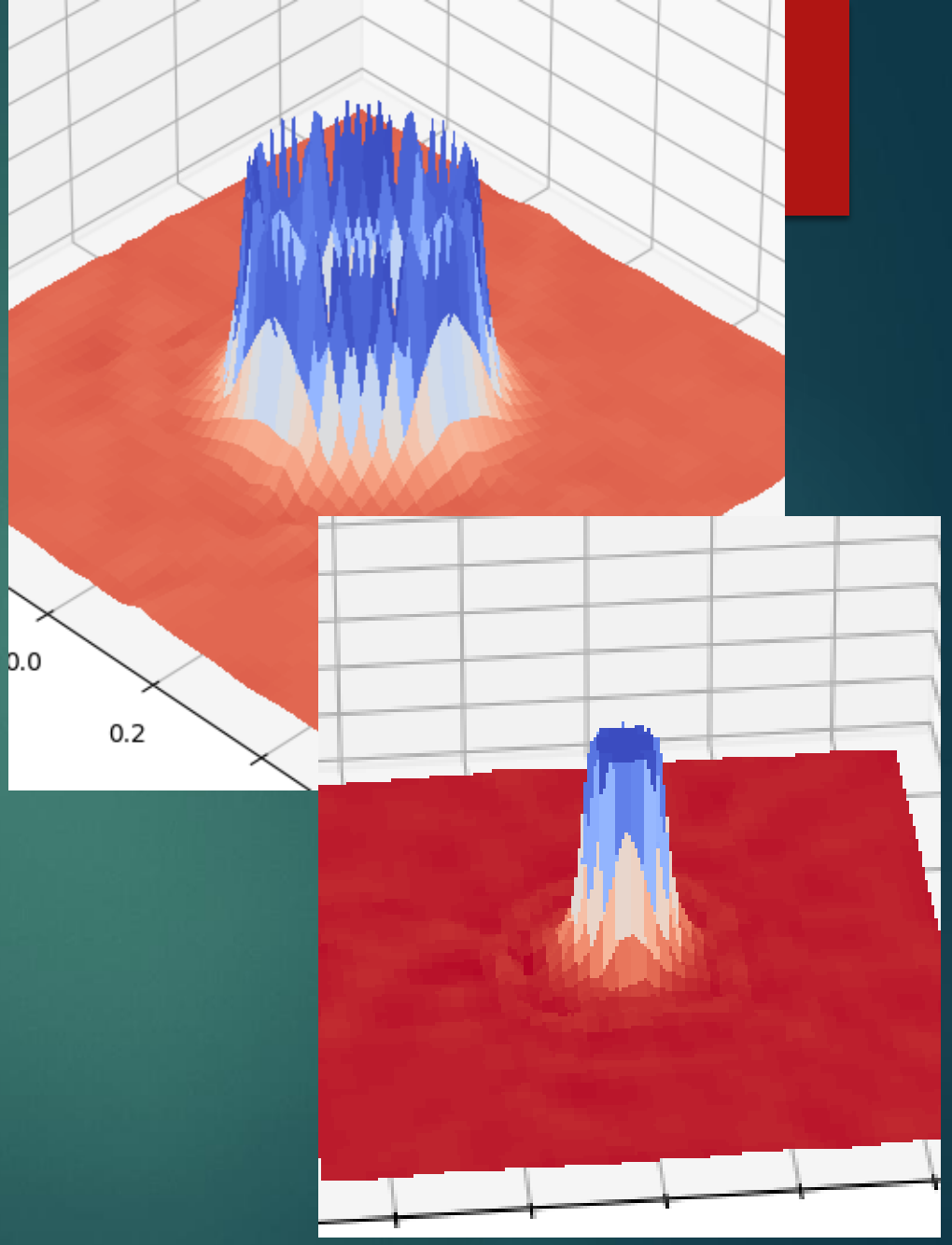

−0.4
−0.2
0.0
−0.2
0.0
0.2
0.4



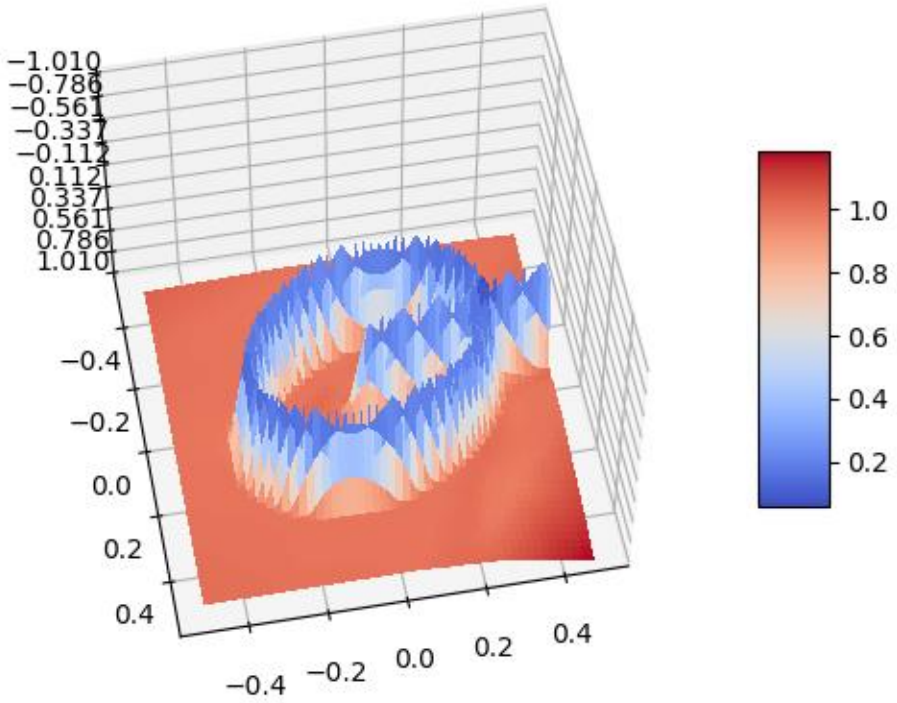


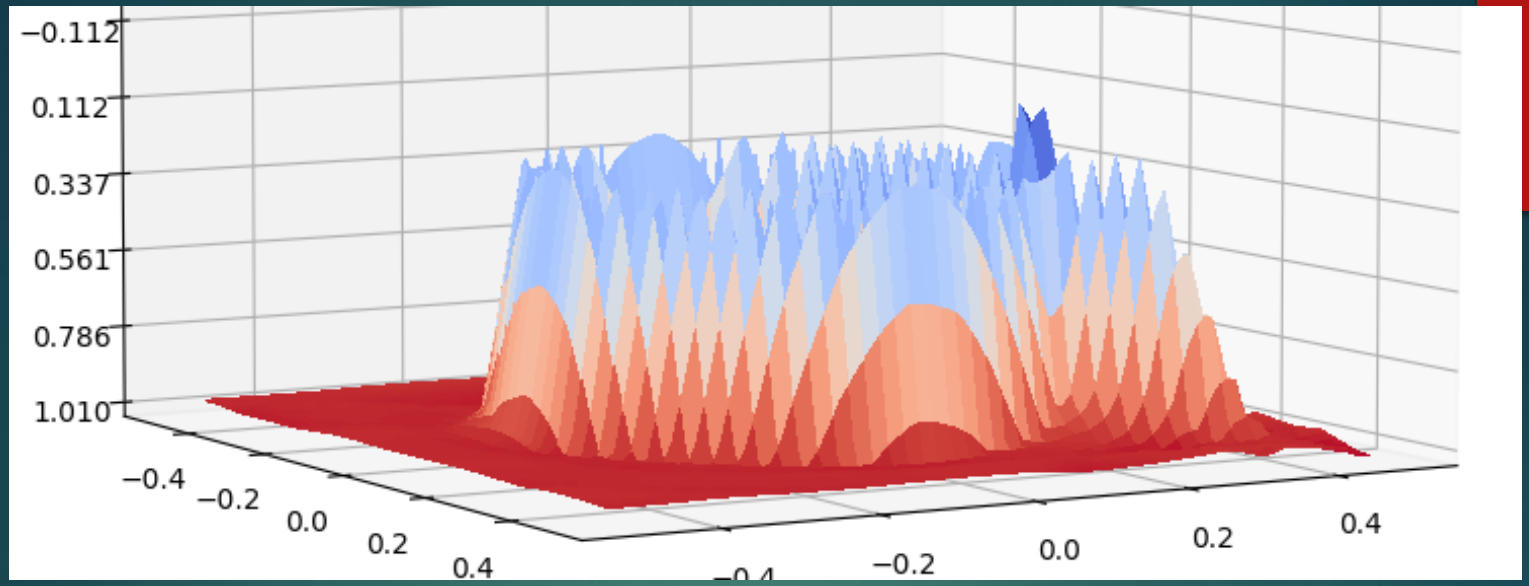
Beta = 0



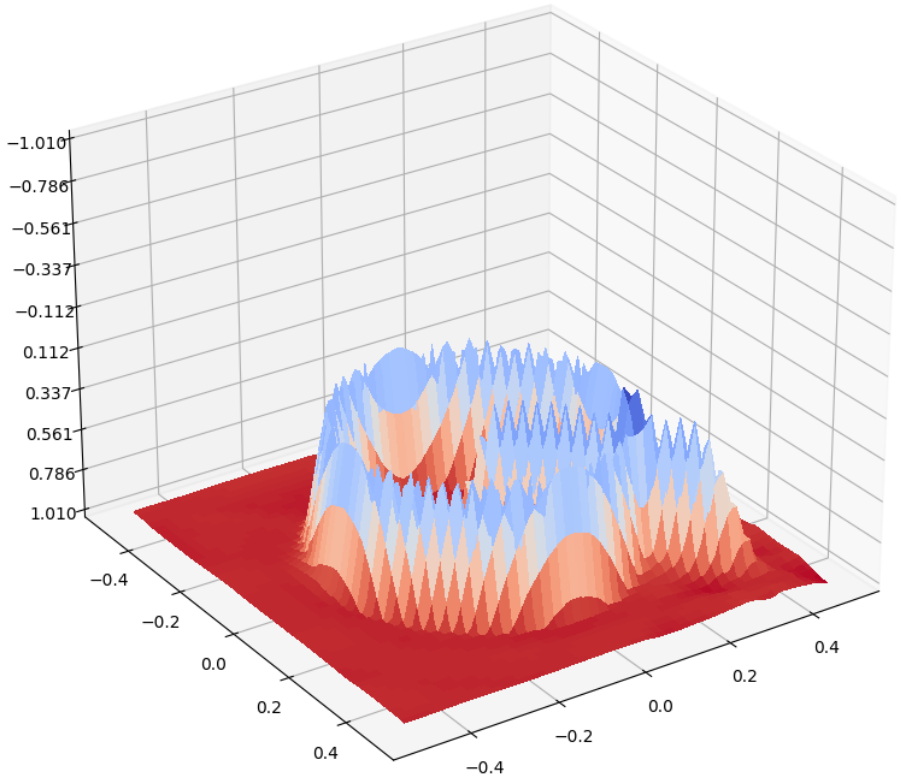


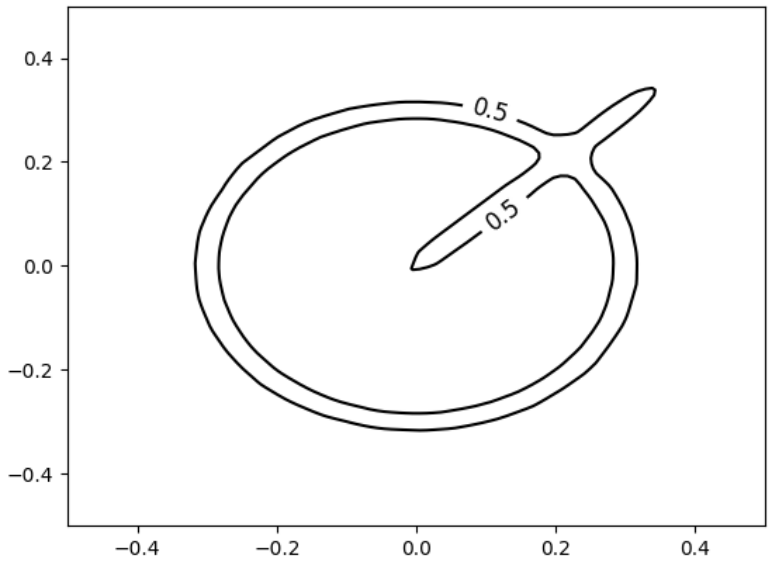
E=0,01

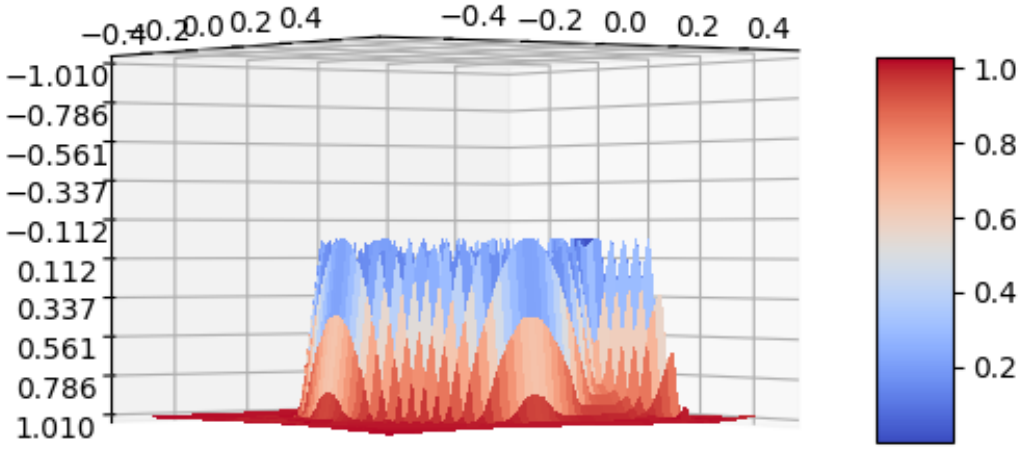

−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
−0.4
−0.2
0.0
0.2
0.4
−0.4
−0.2
0.0
0.2
0.4
1.0
0.8
0.6
0.4
0.2

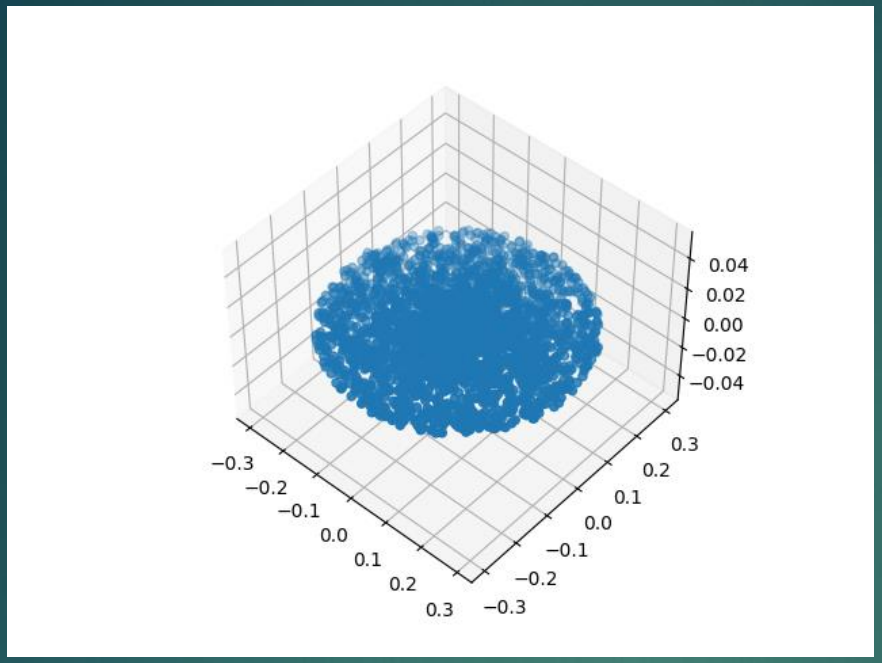

−0.112
0.112
0.337
0.561
0.786
1.010
−0.4
−0.2
0.0
0.2
0.4
−0.4
−0.2
0.0
0.2
0.4

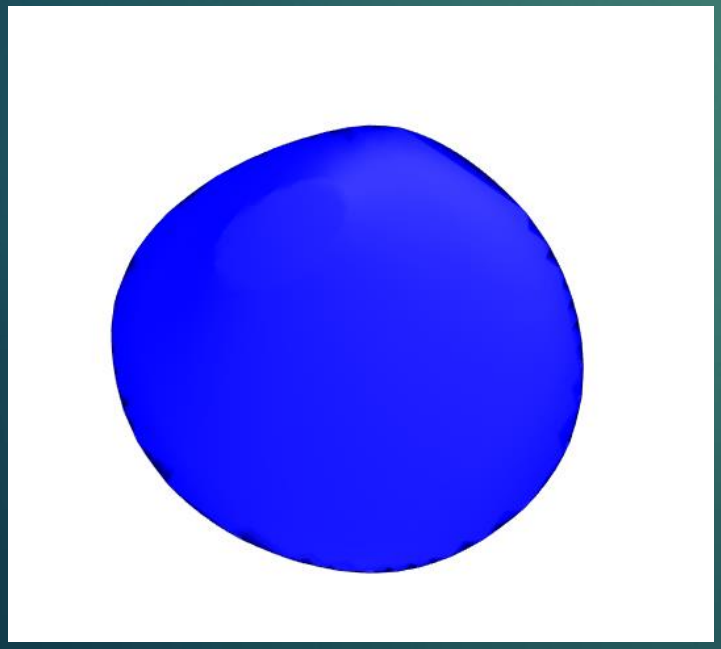
$$\sum \int v^2$$

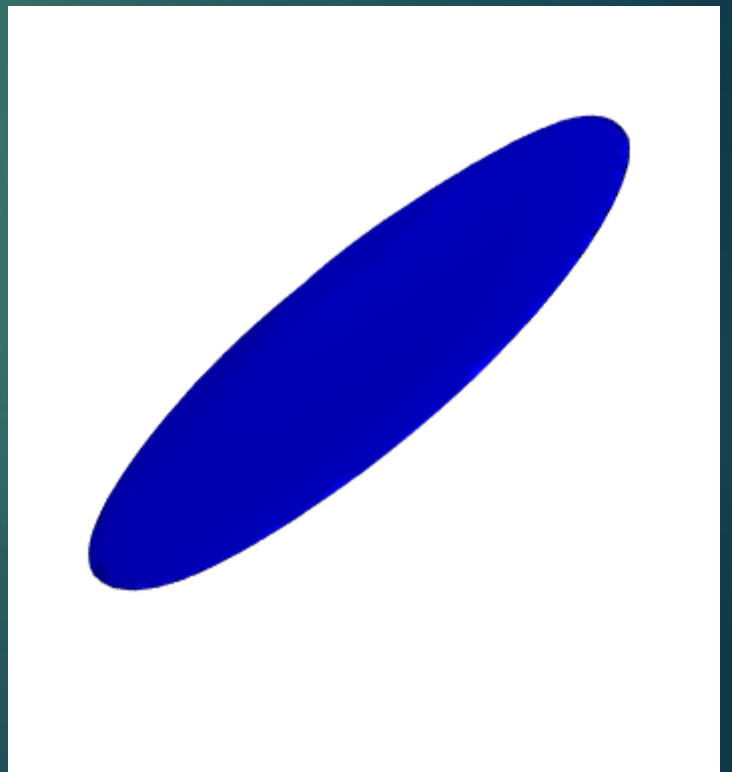

−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
−0.4
−0.2
0.0
0.2
0.4
−0.4
−0.2
0.0
0.2
0.4


0.4
0.2
0.0
−0.2
−0.4
0.5
0.5
−0.4
−0.2
0.0
0.2
0.4

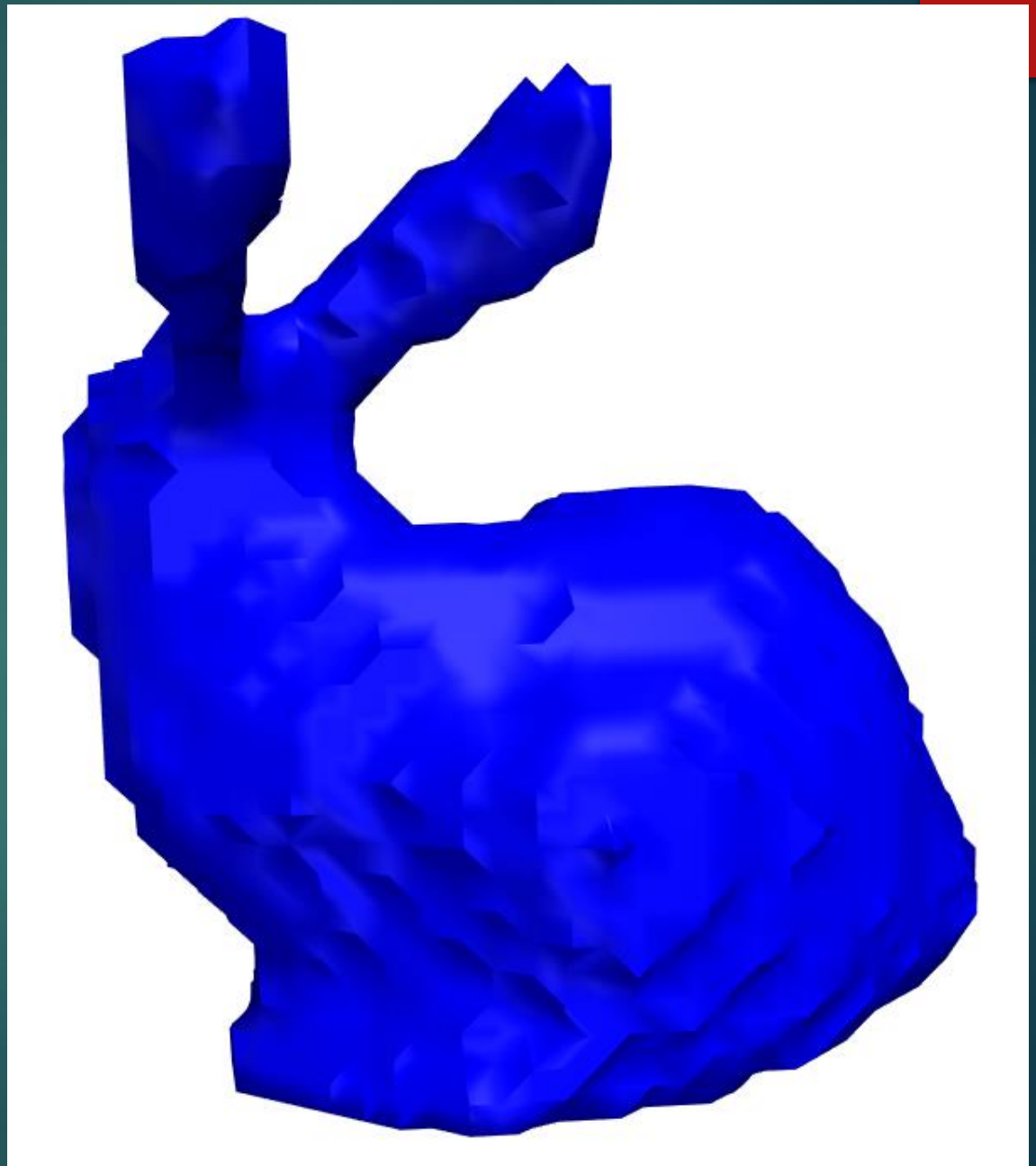

−0.4
−0.2
0.0
0.2
0.4
−0.4
−0.2
0.0
0.2
0.4
−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
1.0
0.8
0.6
0.4
0.2

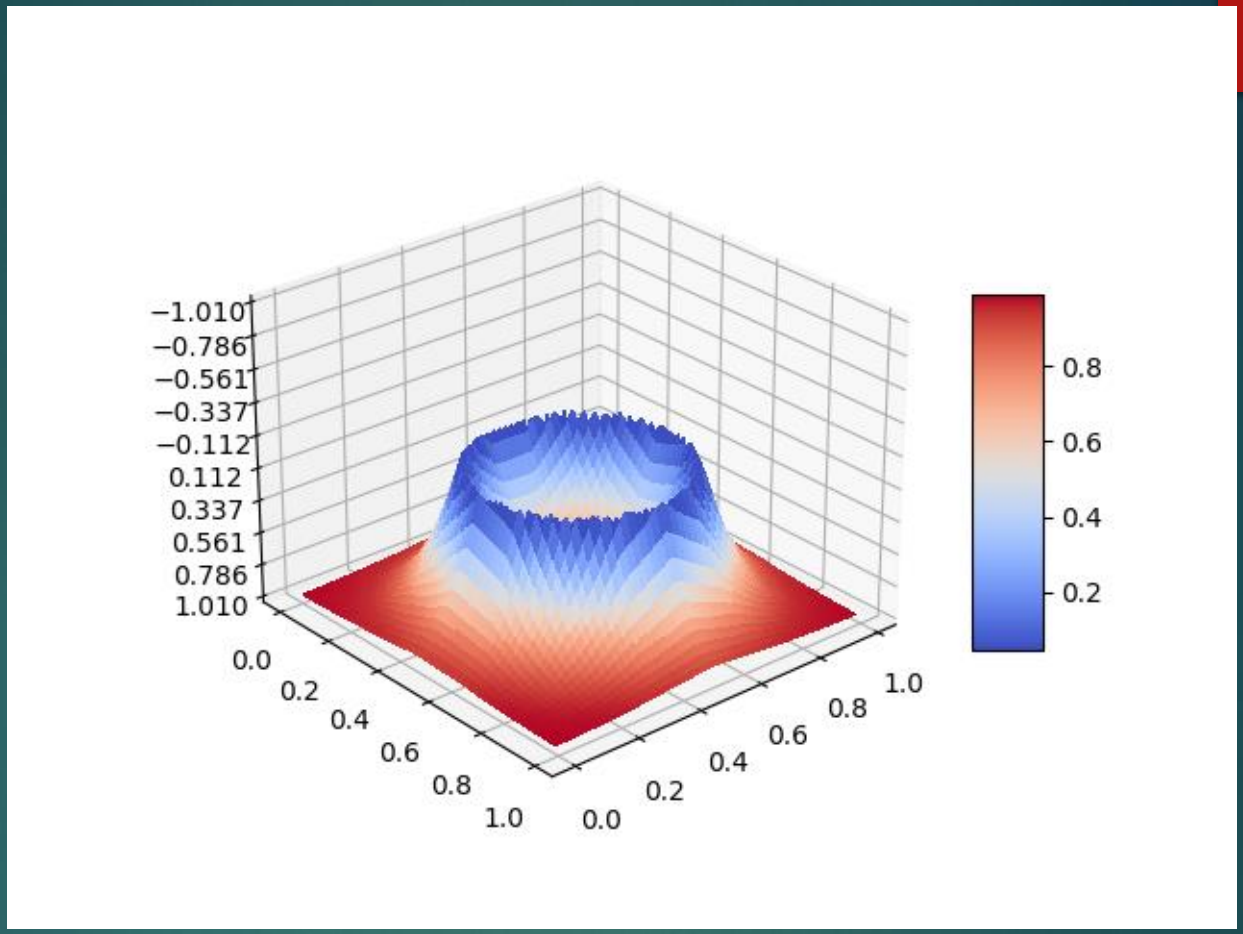

0.04
0.02
0.00
−0.02
−0.04
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3



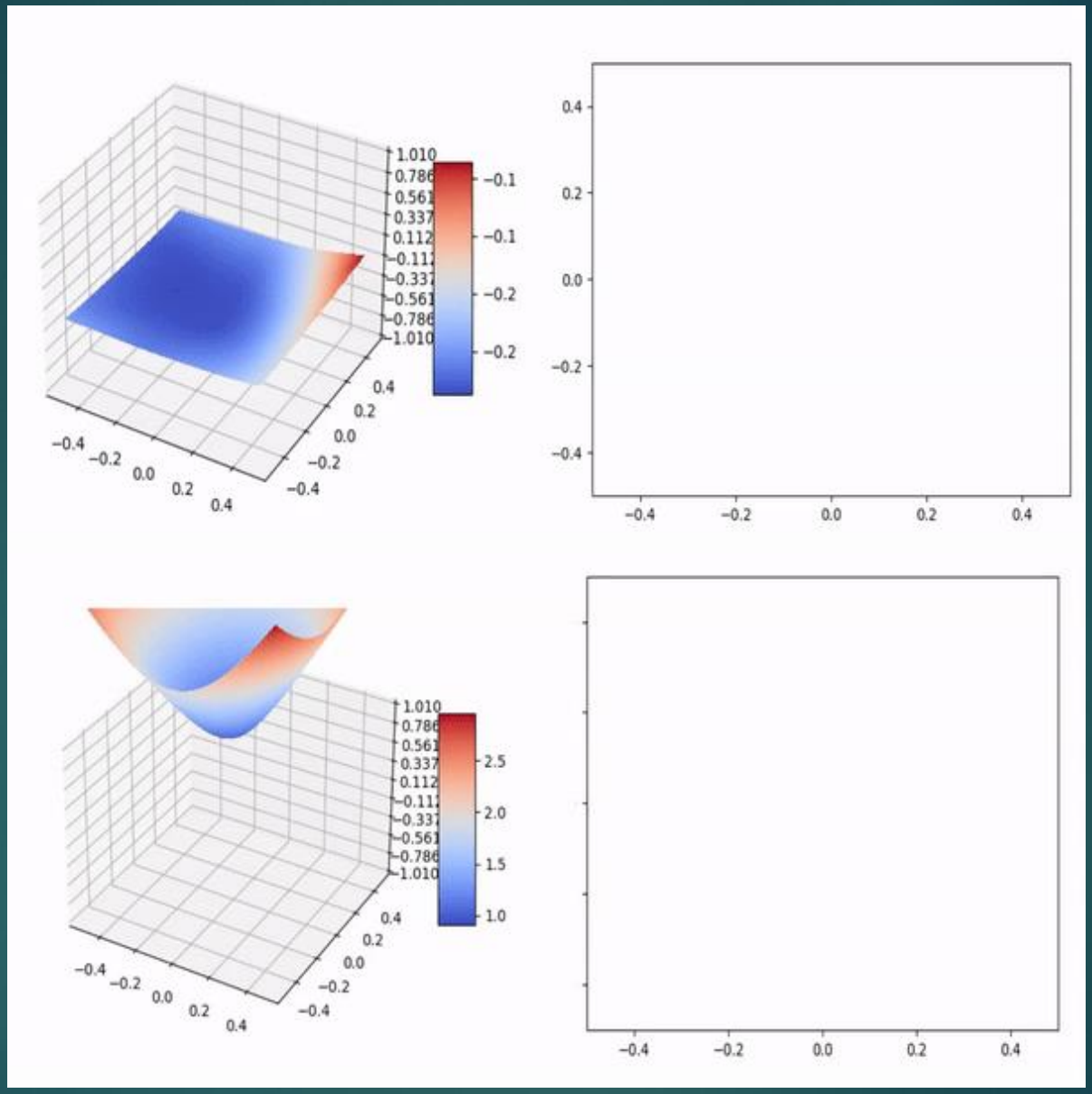


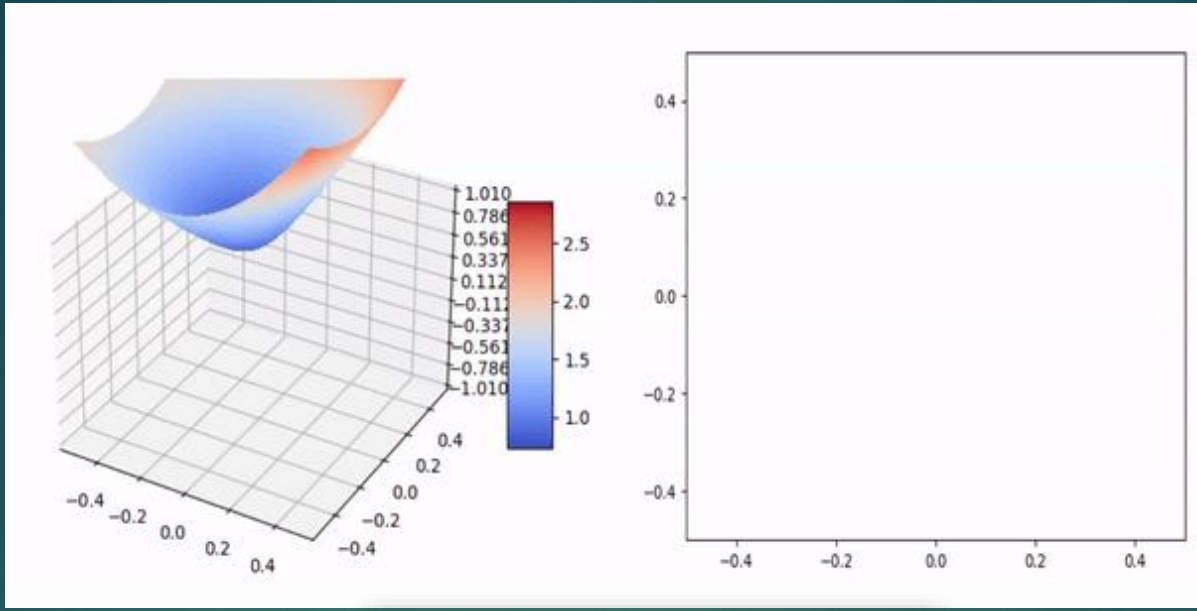
60^3

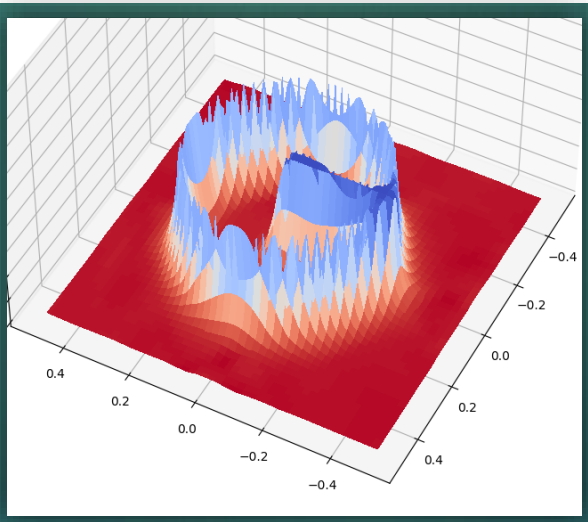


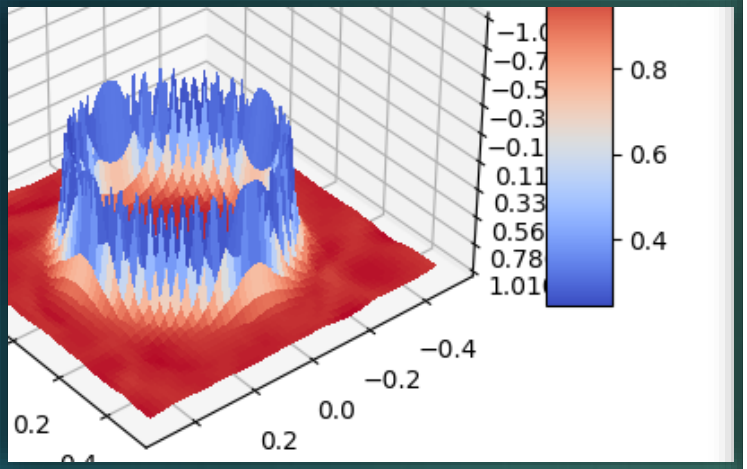

−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
0.8
0.6
0.4
0.2

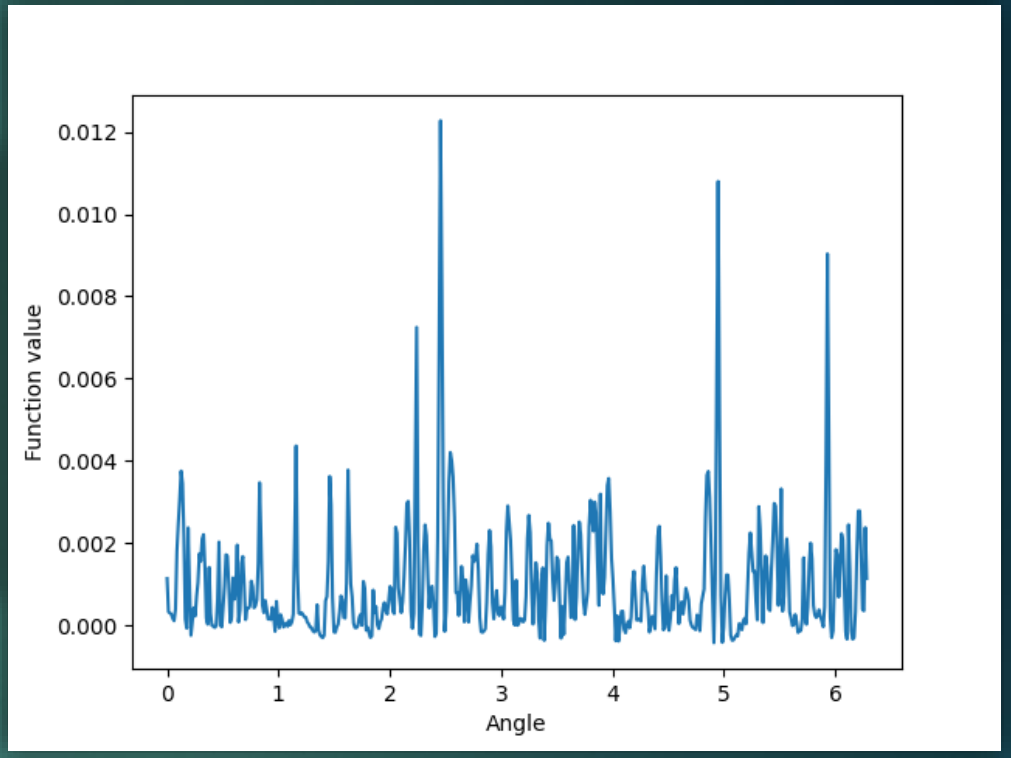
E=0,05

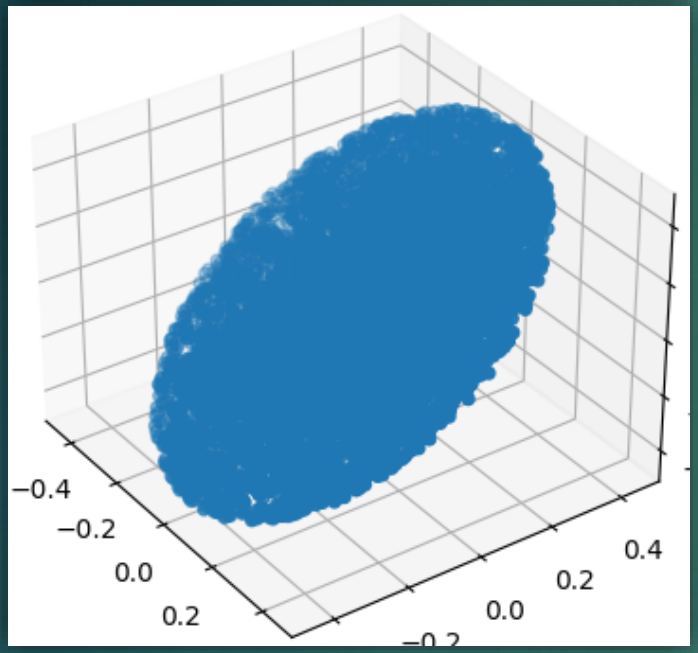

1.010
0.786
0.561
0.337
0.112
−0.111
−0.337
−0.561
−0.786
−1.010
−0.1
−0.1
−0.2
−0.2
−0.4
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
1.010
0.786
0.561
0.337
0.112
−0.111
−0.337
−0.561
−0.786
−1.010
2.5
2.0
1.5
1.0
−0.4
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4

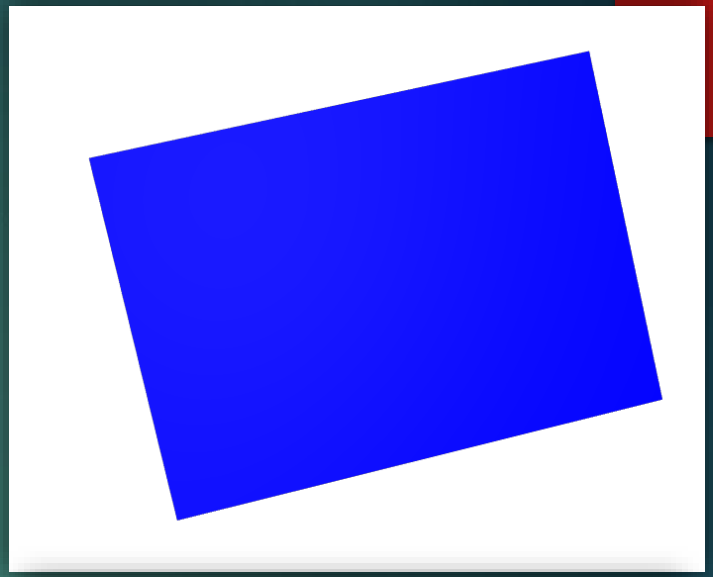

1.010
0.786
0.561
0.337
0.112
−0.112
−0.337
−0.561
−0.786
−1.010
2.5
2.0
1.5
1.0
0.4
0.2
0.0
−0.2
−0.4
0.4
0.2
0.0
−0.2
−0.4
0.4
0.2
0.0
−0.2
−0.4

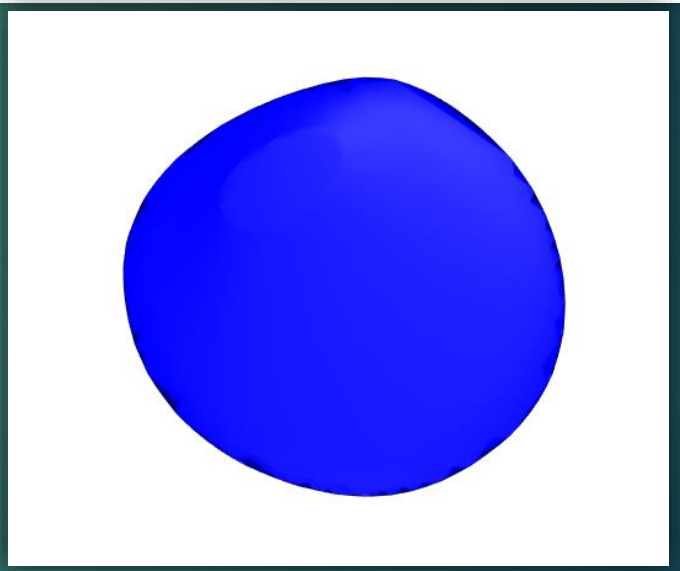

0.4
0.2
0.0
−0.2
−0.4
0.4
0.2
0.0
−0.2
−0.4

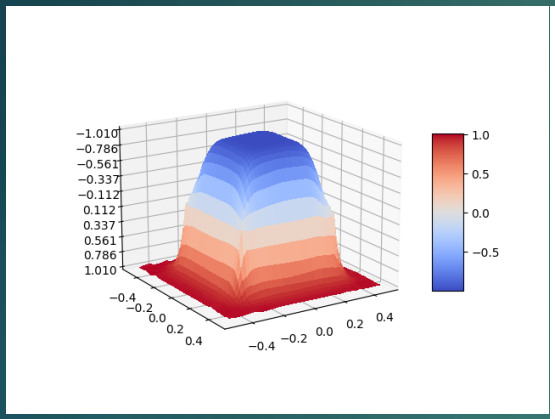

0.8
0.6
0.4
−0.4
−0.2
0.0
0.2
0.2

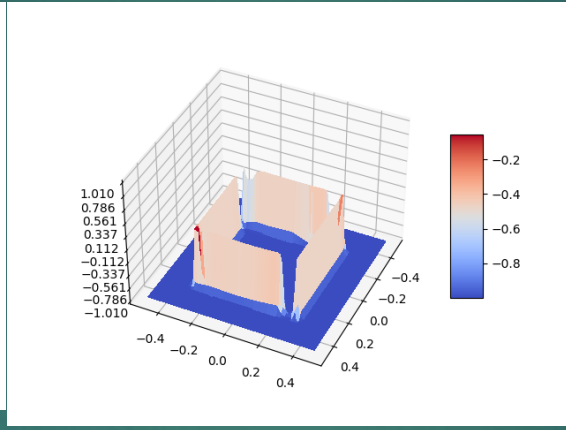

Function value
0.012
0.010
0.008
0.006
0.004
0.002
0.000
0
1
2
3
4
5
6
Angle

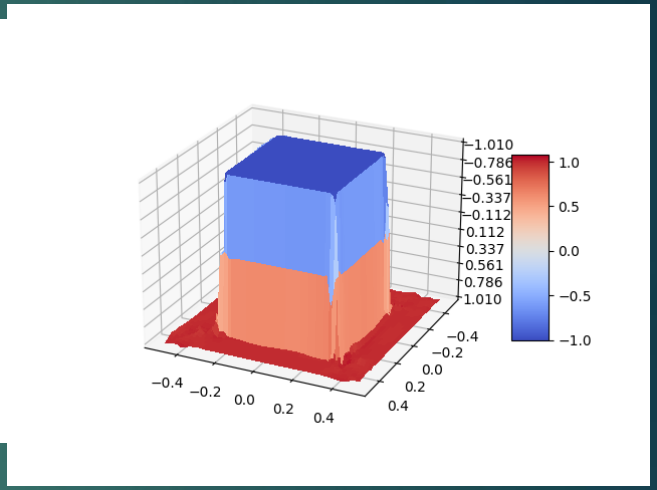

−0.4
−0.2
0.0
0.2
0.4
0.2
0.0

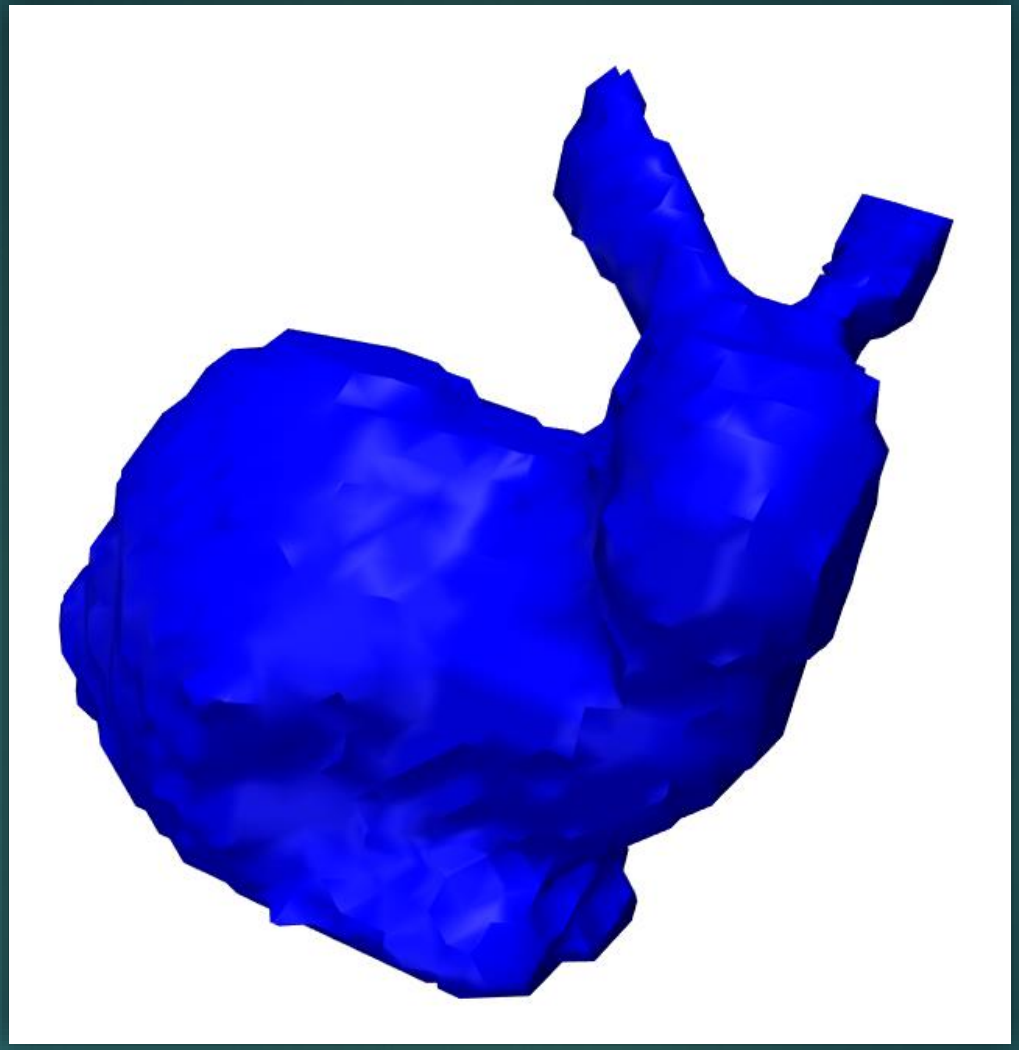





−1.010
−0.786
−0.561
−0.337
−0.112
0.112
0.337
0.561
0.786
1.010
−0.4
−0.2
0.0
0.2
0.4
1.0
0.5
0.0
−0.5


1.010
0.786
0.561
0.337
0.112
−0.112
−0.337
−0.561
−0.786
−1.010
−0.4
−0.2
0.0
0.2
0.4
−0.2
−0.4
−0.6
−0.8


1.010
0.786
0.561
0.337
0.112
−0.112
−0.337
−0.561
−0.786
−1.010
−0.4
−0.2
0.0
0.2
0.4
1.0
0.5
0.0
−0.5
−1.0



very different.

Consider the function $\gamma(t) = 1 - \exp(-t/2)$ for $t \geq 0$. We leave the following result to the reader:

Exercise. *Prove that* $v_\varepsilon(x) = \gamma(x/\varepsilon)$ *minimizes*

$$\int_0^\infty \varepsilon v'(x)^2 + \frac{(1-v(x))^2}{4\varepsilon}\,dx$$

on the set $\{v \in L^2_{loc}(0,+\infty), v' \in L^2(0,+\infty), v(0) = 0\}$, *and that the value of the minimum is* $1/2$.

Now, we set for every $\varepsilon > 0$ $v_\varepsilon(x) = 0$ if $|x| < a_\varepsilon$, and $v_\varepsilon(x) = \gamma\left(\frac{|x|-a_\varepsilon}{\varepsilon}\right)$ otherwise, where a_ε goes to zero with ε and will be fixed later on, and we set $u_\varepsilon(x) = u(-a_\varepsilon) + \frac{u(a_\varepsilon)-u(-a_\varepsilon)}{2a_\varepsilon}(x+a_\varepsilon)$ if $|x| < a_\varepsilon$, and $u_\varepsilon(x) = u(x)$ otherwise.

Then (using the result of the previous exercise),

$$\begin{aligned}
E_\varepsilon(u_\varepsilon, v_\varepsilon) &= \int_{|x|\geq a_\varepsilon} (k_\varepsilon + v_\varepsilon(x))u'(x)^2\,dx + 2a_\varepsilon k_\varepsilon \left(\frac{u(a_\varepsilon) - u(-a_\varepsilon)}{2a_\varepsilon}\right)^2 \\
&\quad + \int_{|x|\geq a_\varepsilon} \frac{1}{\varepsilon}\gamma'\left(\frac{|x|-a_\varepsilon}{\varepsilon}\right)^2 + \frac{1}{4\varepsilon}\left(1 - \gamma\left(\frac{|x|-a_\varepsilon}{\varepsilon}\right)\right)^2 dx \\
&\quad + \frac{2a_\varepsilon}{4\varepsilon} + \int_{|x|\geq a_\varepsilon} (u(x) - g(x))^2\,dx + \int_{-a_\varepsilon}^{a_\varepsilon} (u_\varepsilon(x) - g(x))^2\,dx \\
&\leq (1+k_\varepsilon)\int_{-1}^{1} u'(x)^2\,dx + 2\|u\|_\infty^2 \frac{k_\varepsilon}{a_\varepsilon} \\
&\quad + 2 \times \frac{1}{2} + \frac{2a_\varepsilon}{4\varepsilon} \\
&\quad + \int_{-1}^{1} (u(x) - g(x))^2\,dx + 2a_\varepsilon(\|u\|_\infty + \|g\|_\infty)^2.
\end{aligned}$$

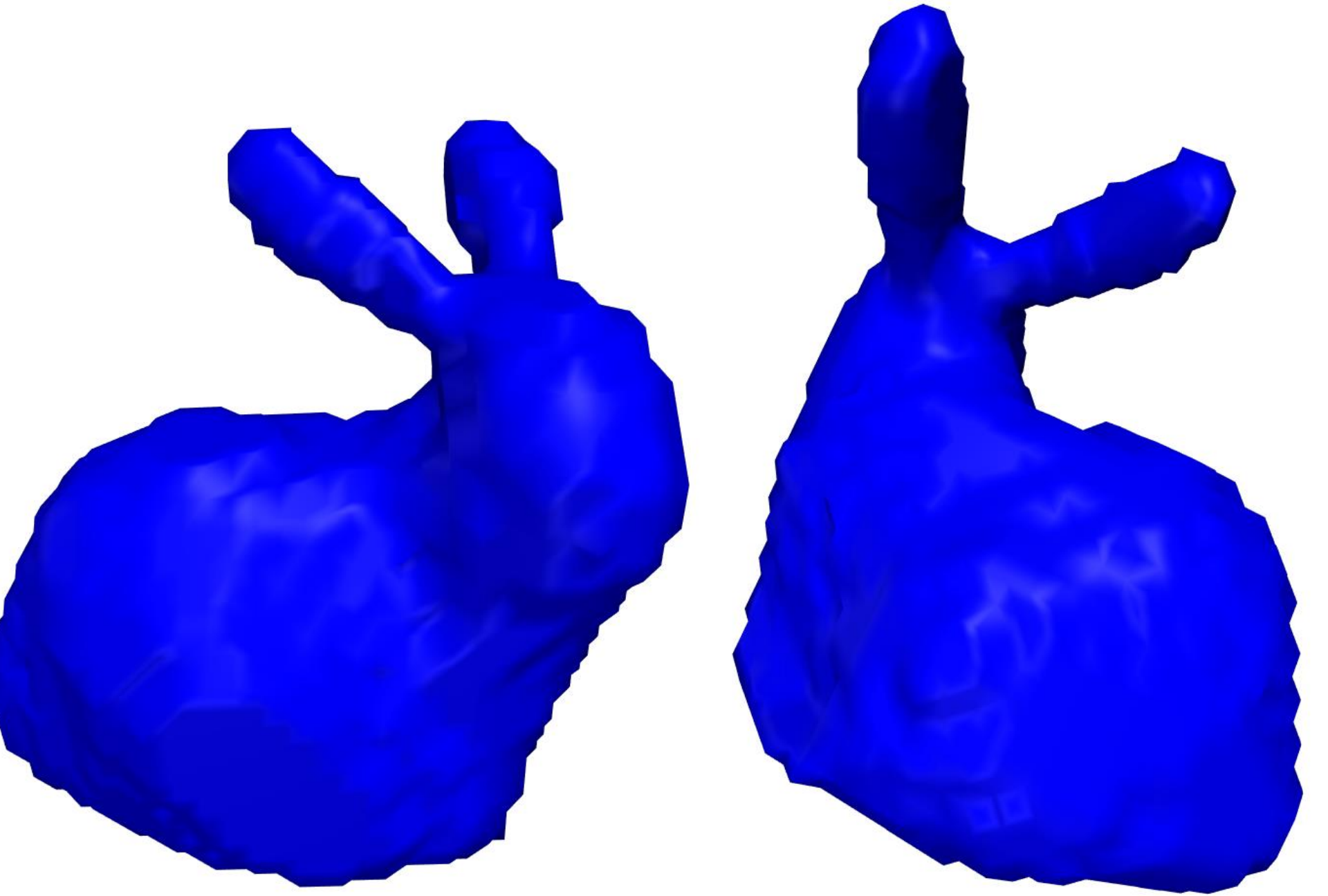
final loss this yields

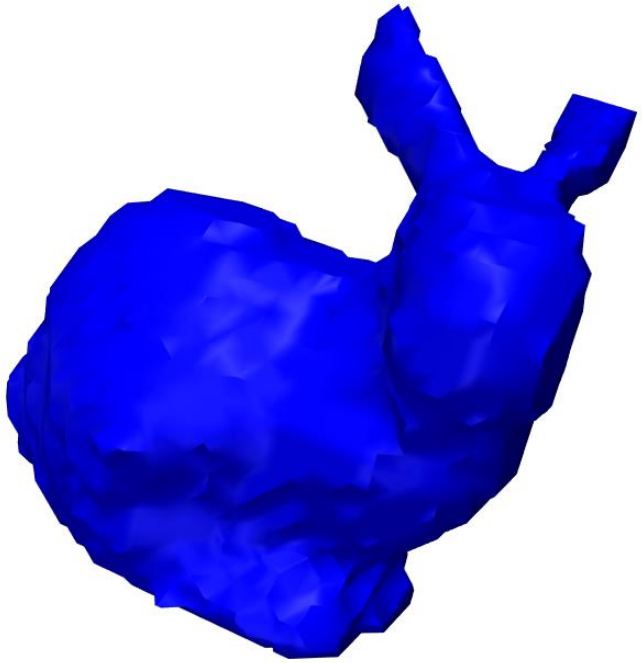
AT-phase fields

$$\min_v \int_\Omega \varepsilon |\nabla v(x)|^2 + \frac{(1-v(x))^2}{4\varepsilon} \, \mathrm{d}x + \frac{C}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} |v(p)|$$

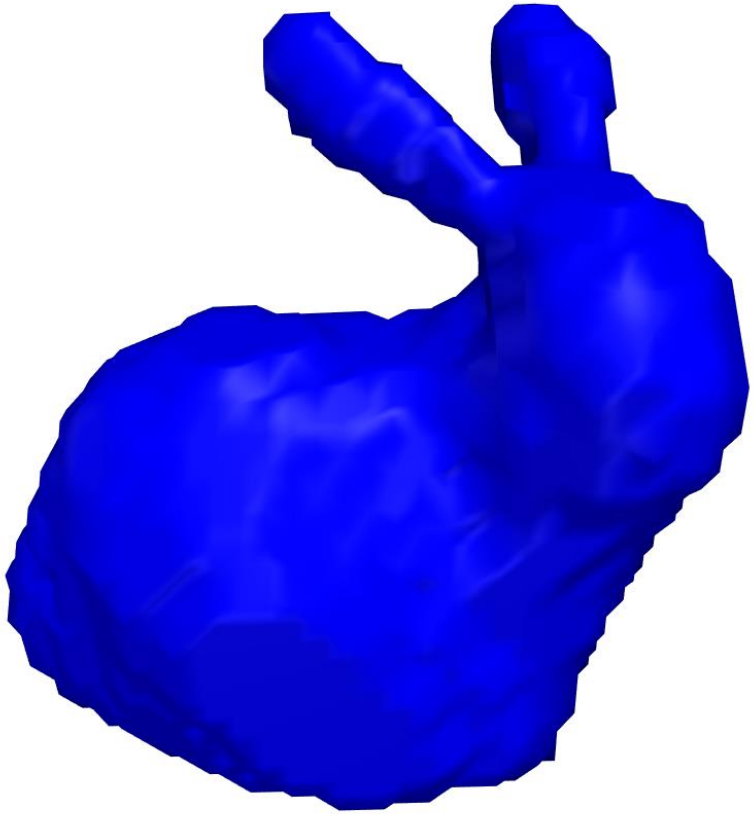
mpare this to the Modica-Mortola based approach, this looks quiet similar. The

2id

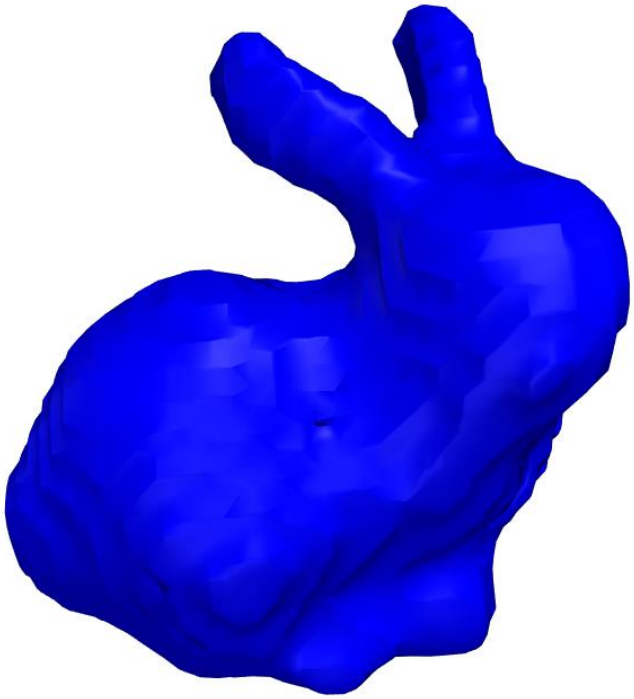


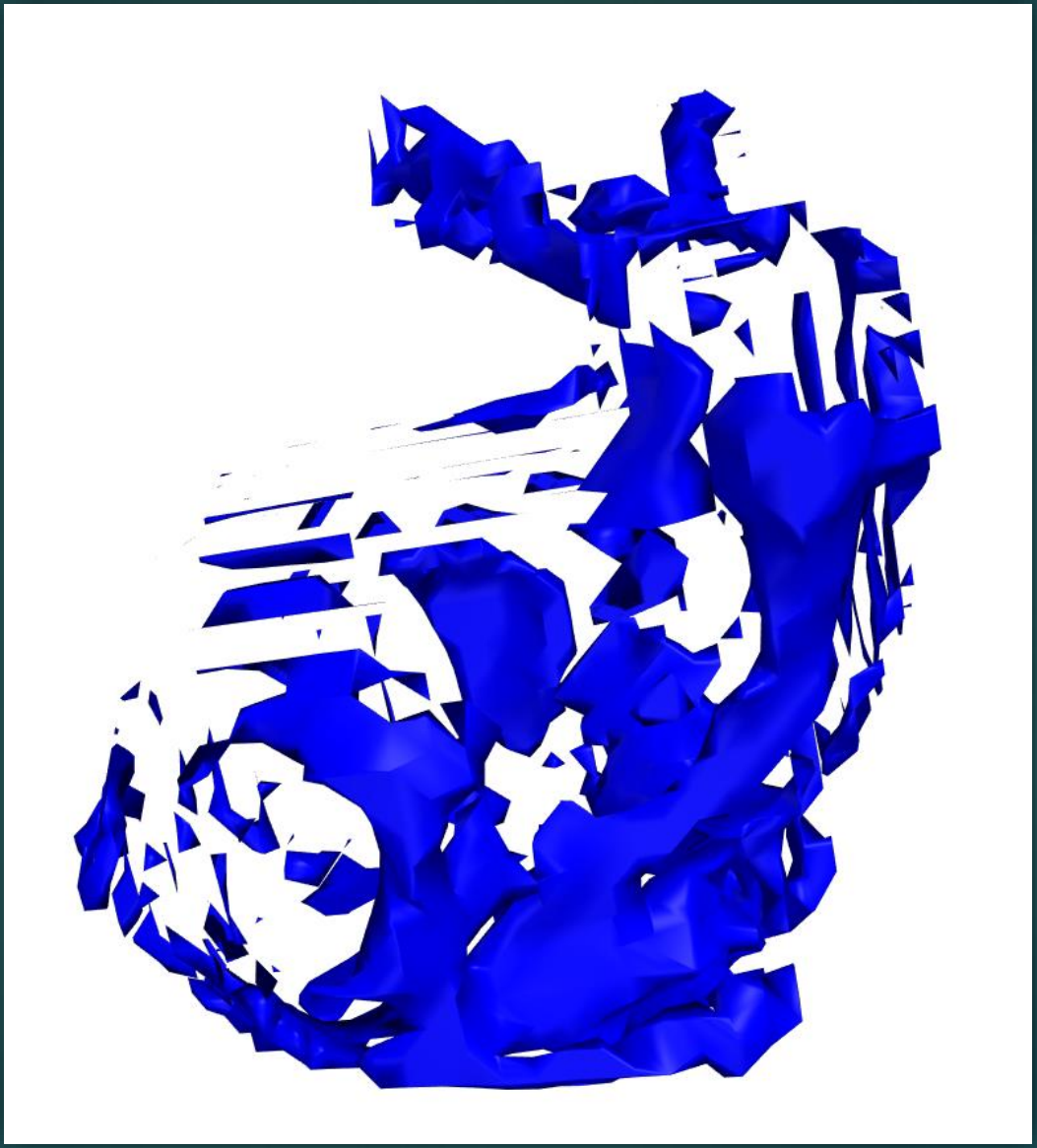


MM

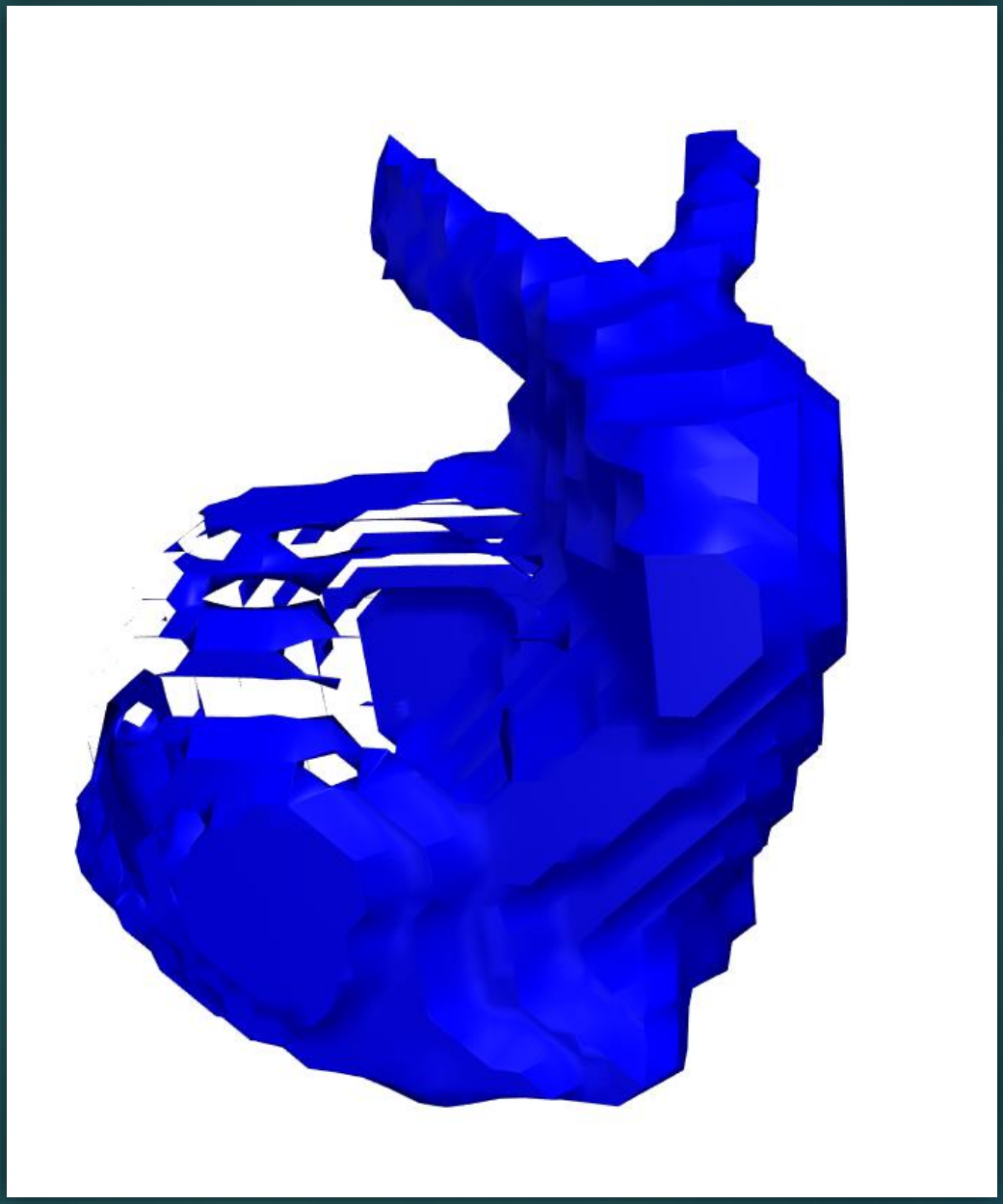


Mix

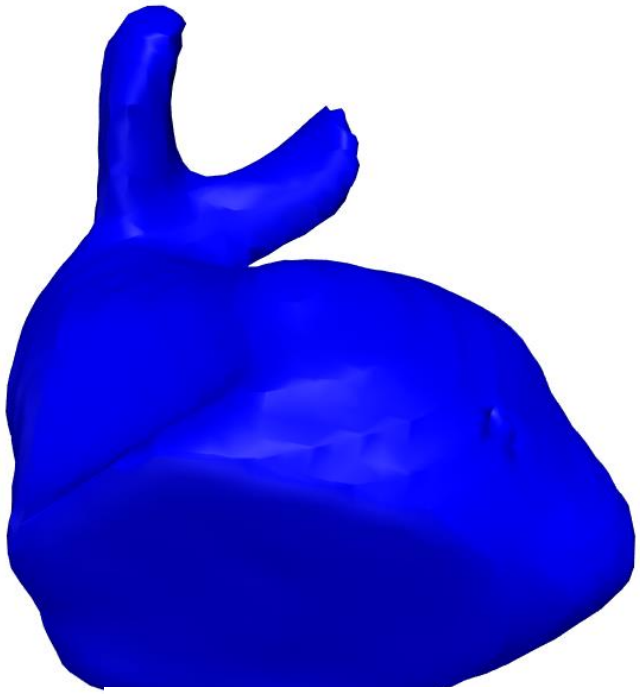
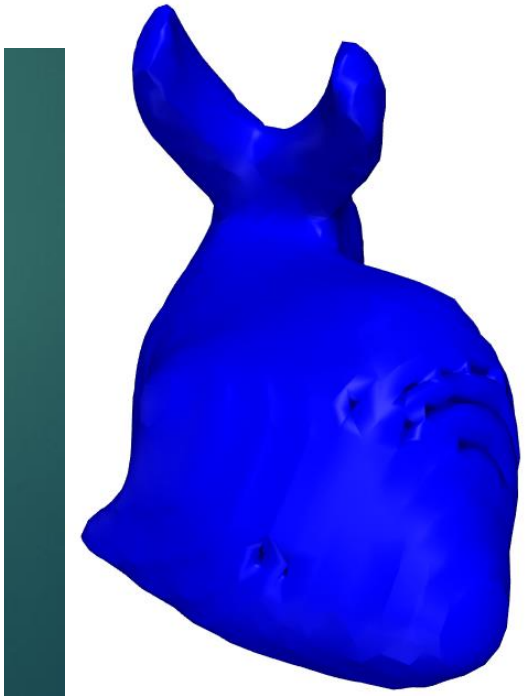




IC=0

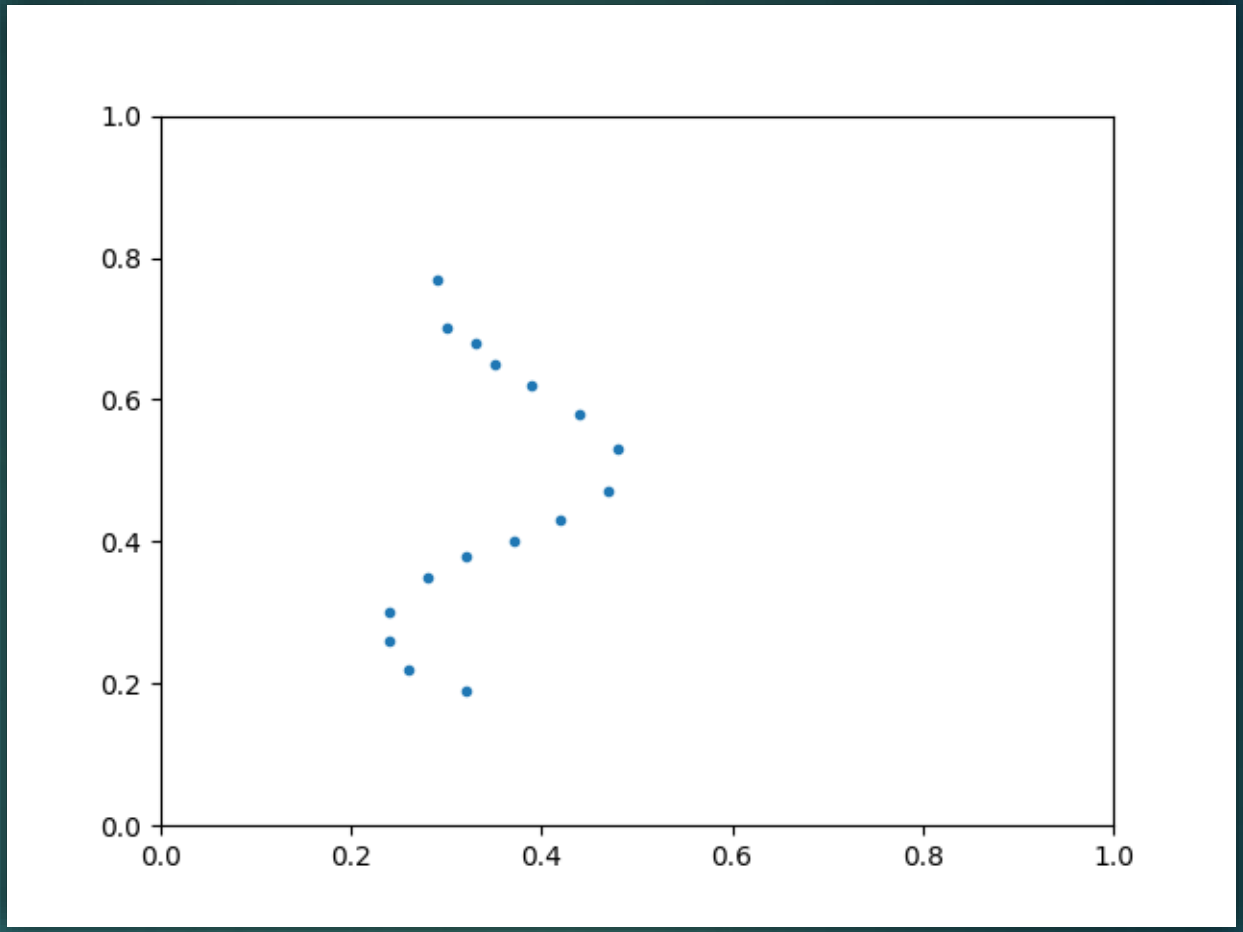


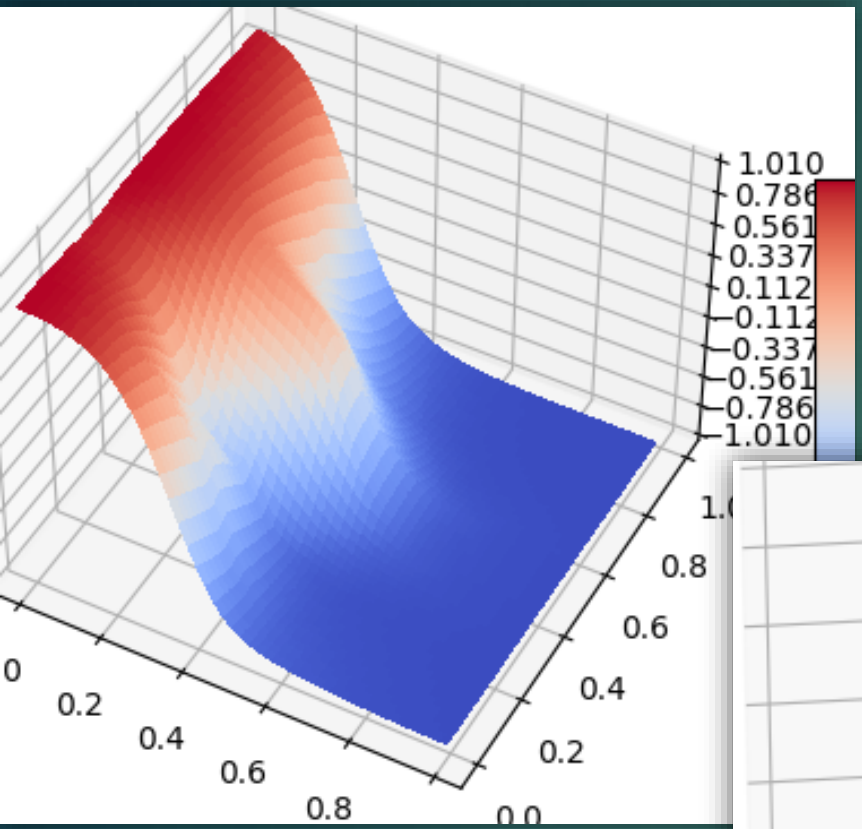
IC=0,01

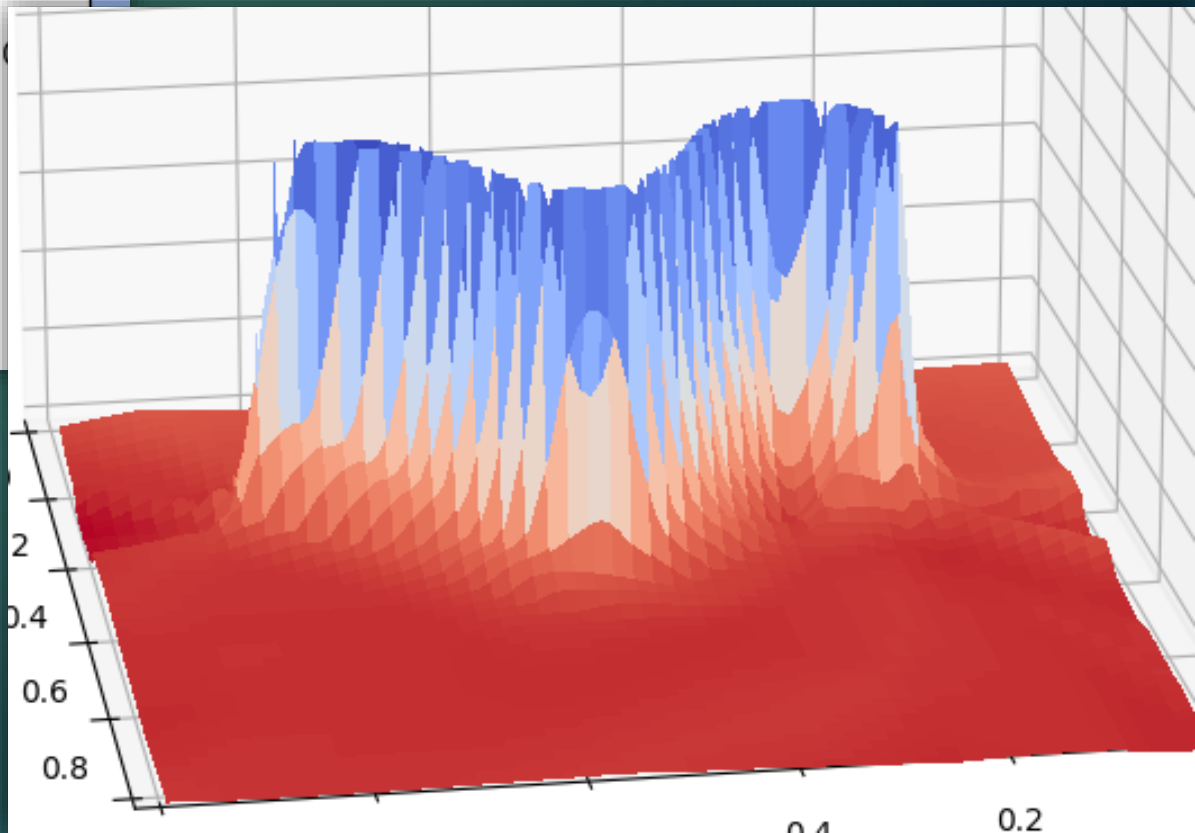




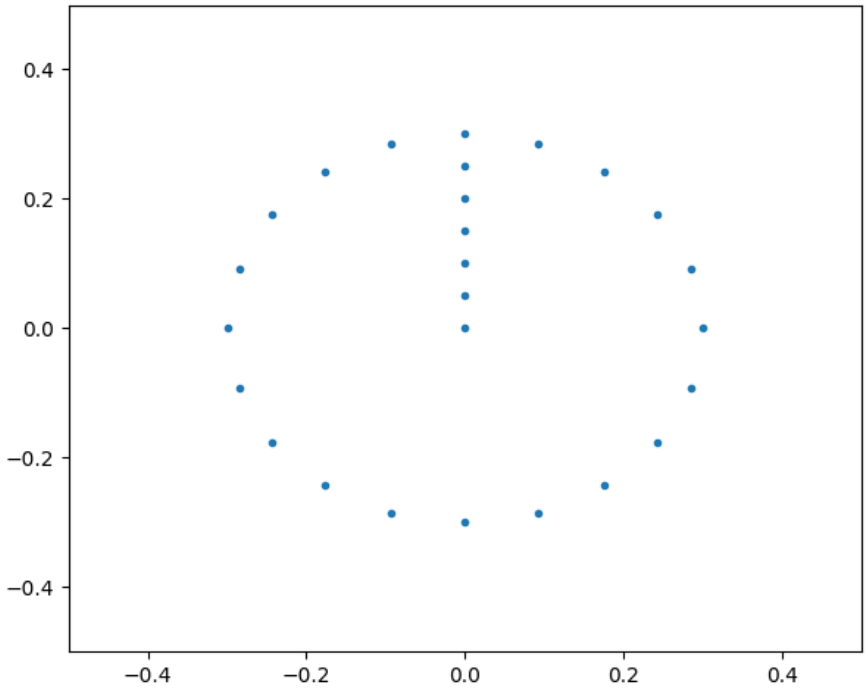
MODICA MORTOLA VS. AMBROSIO

- GSBV

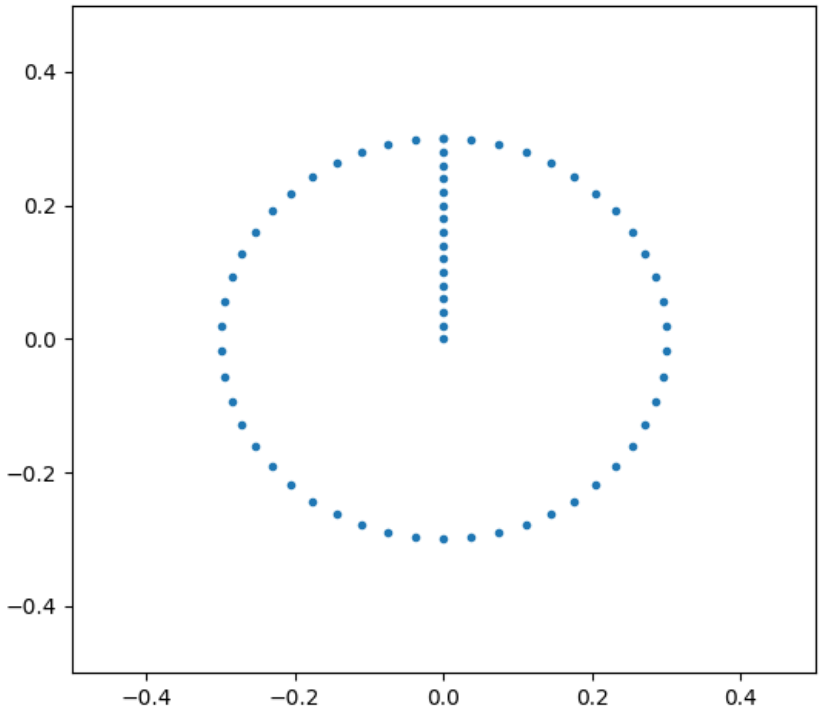

1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0


1.010
0.786
0.561
0.337
0.112
0.112
0.337
0.561
0.786
1.010
0.2
0.4
0.6
0.8
0.2
0.4
0.6
0.8


0.4
0.6
0.8
0.2

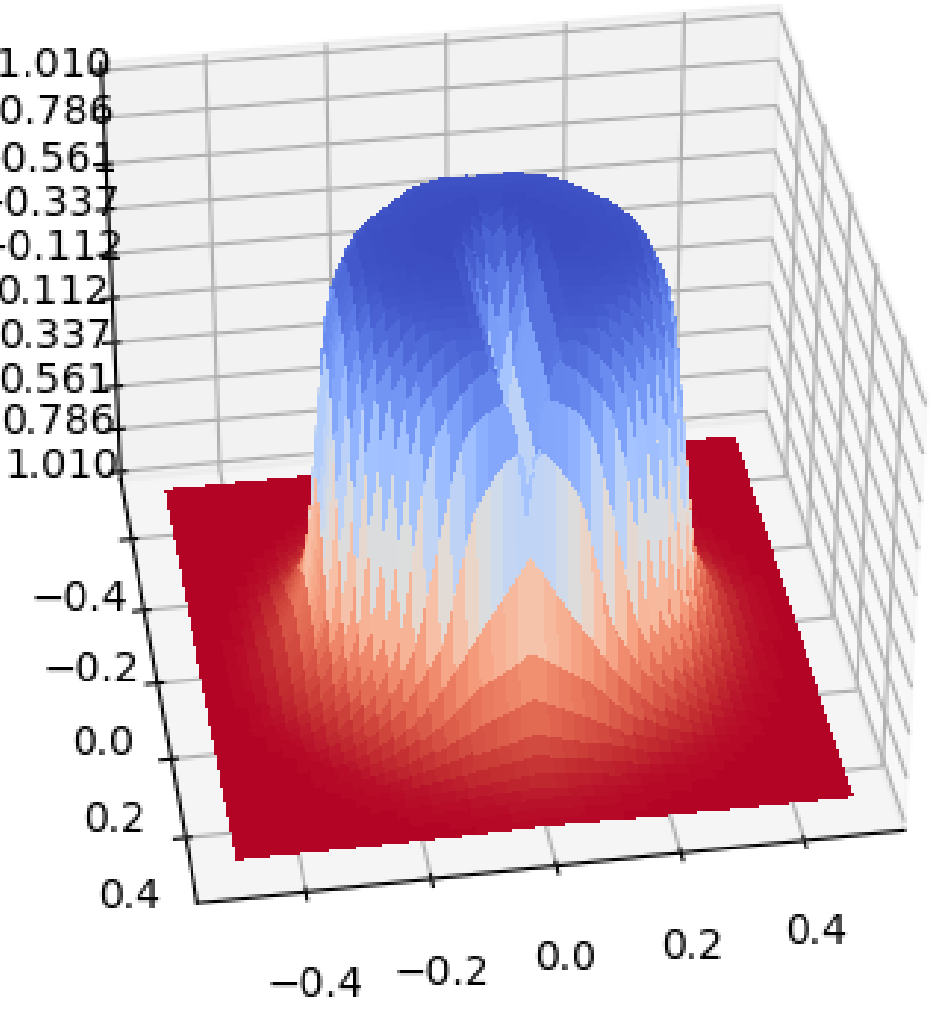


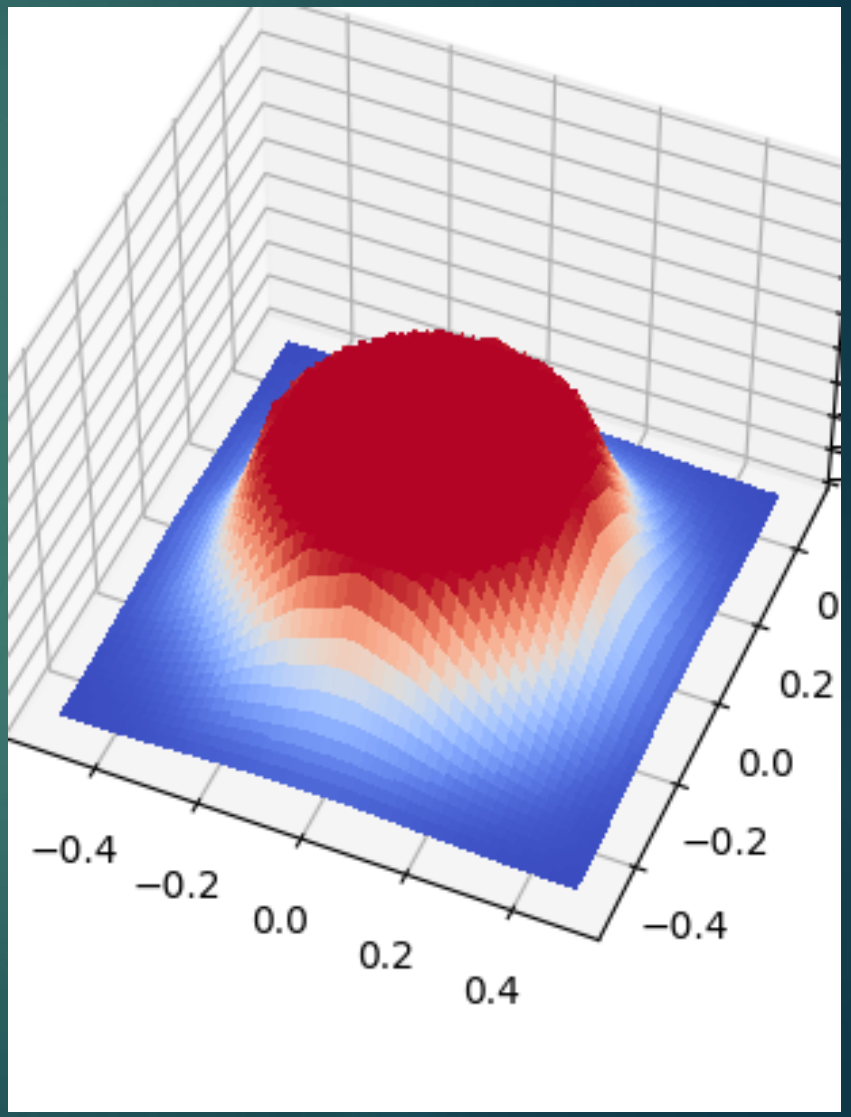
20



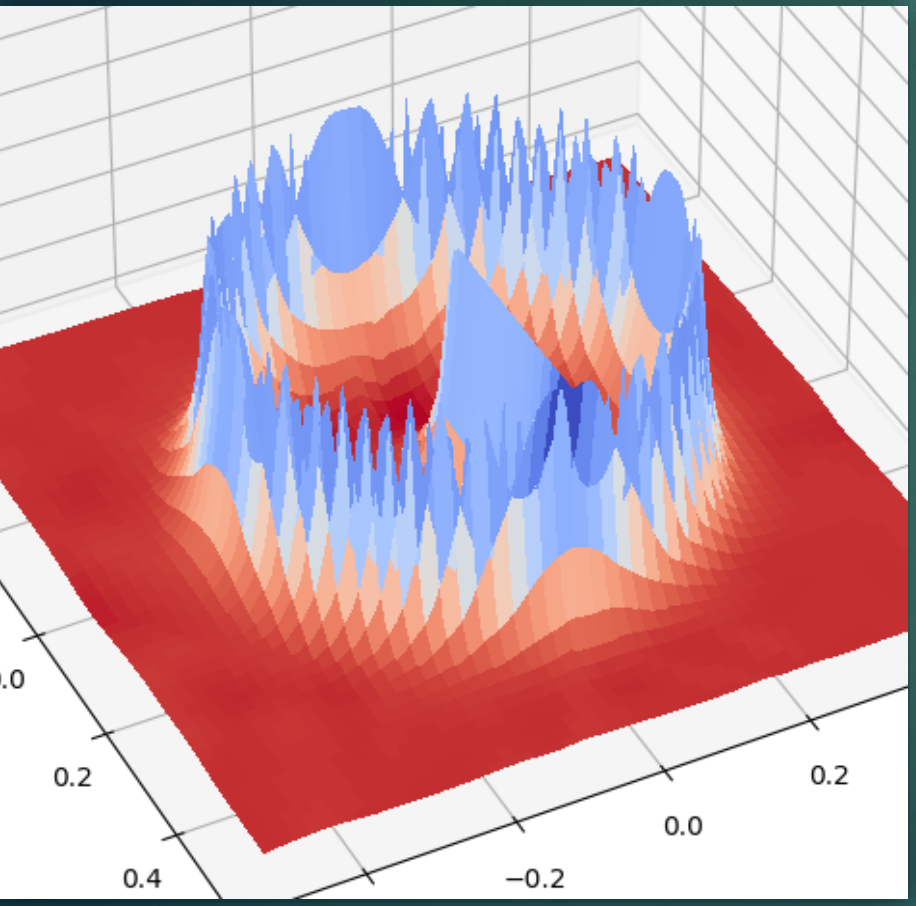
50

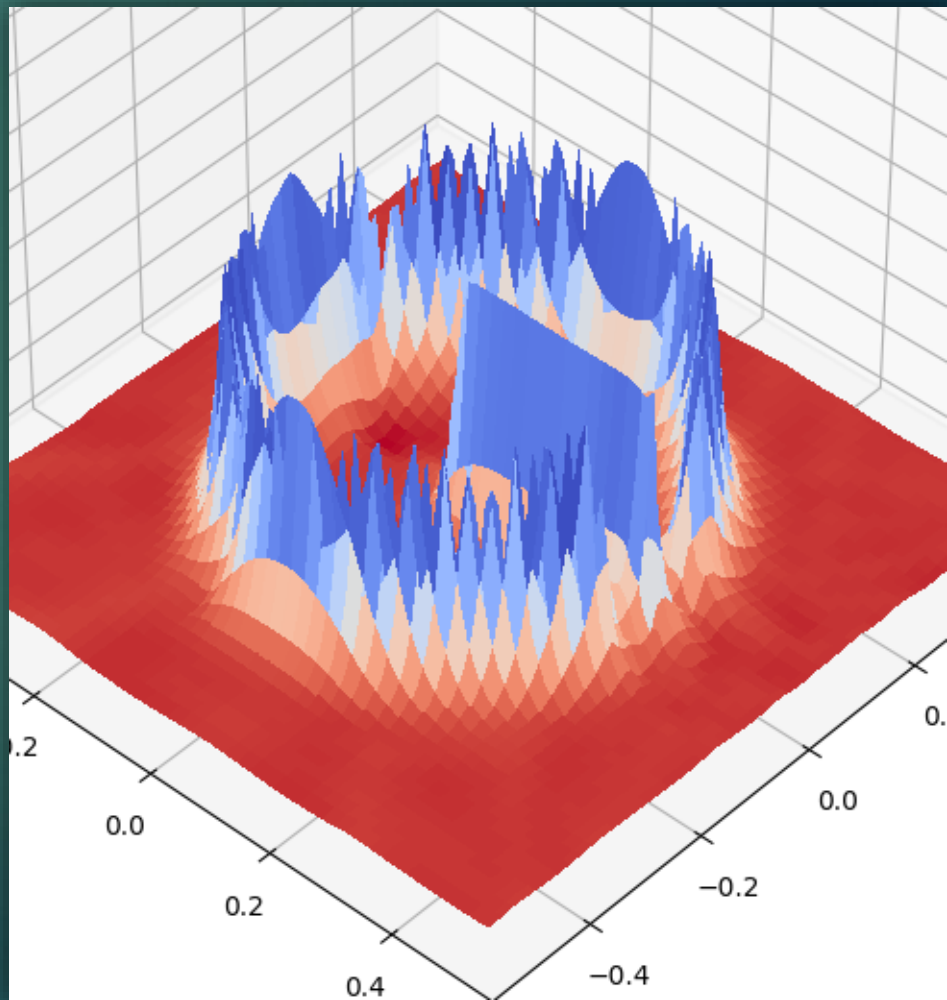
Modica-Mortola

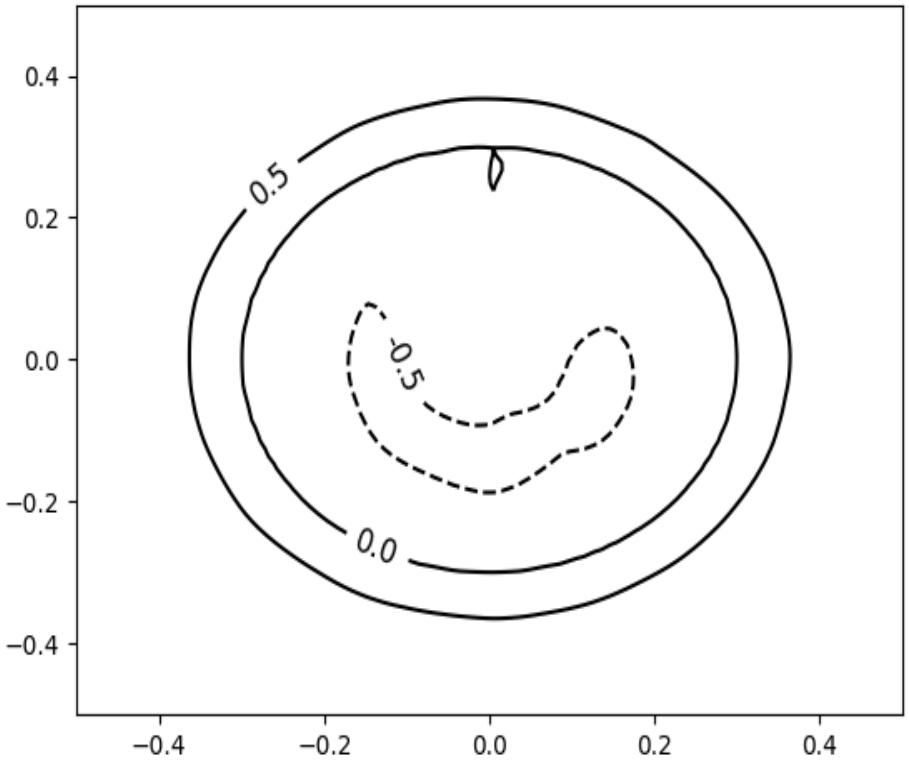




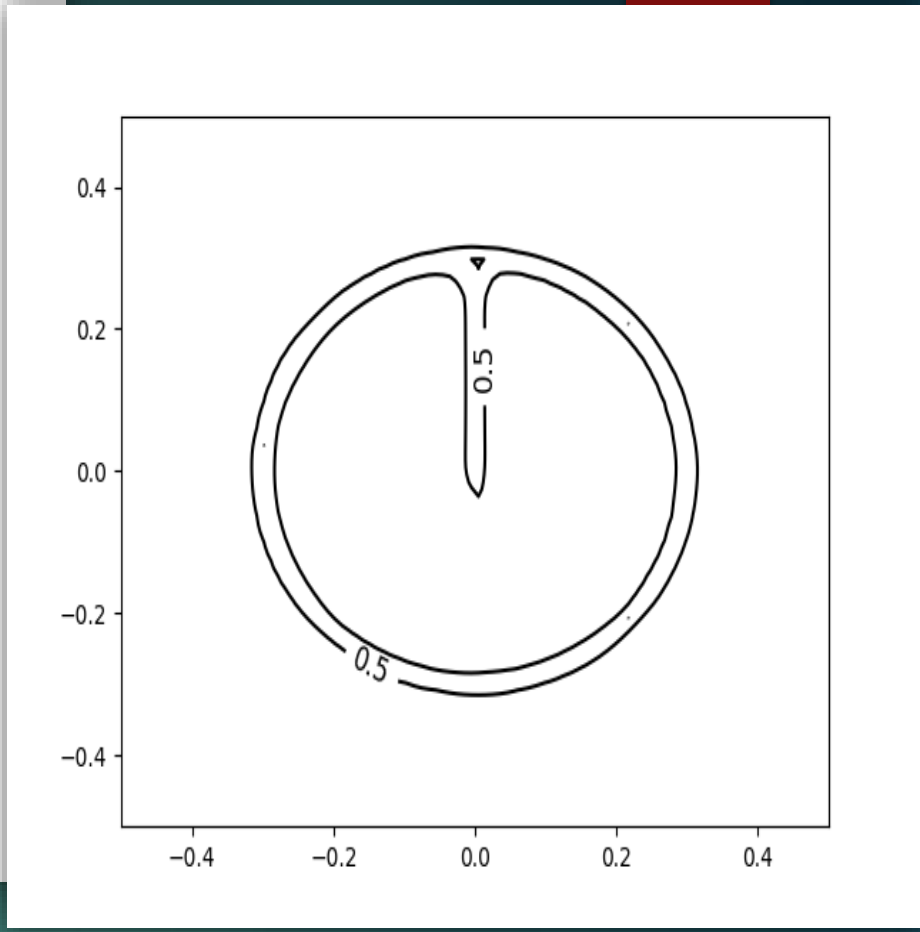
Ambrosio-Tortorelli



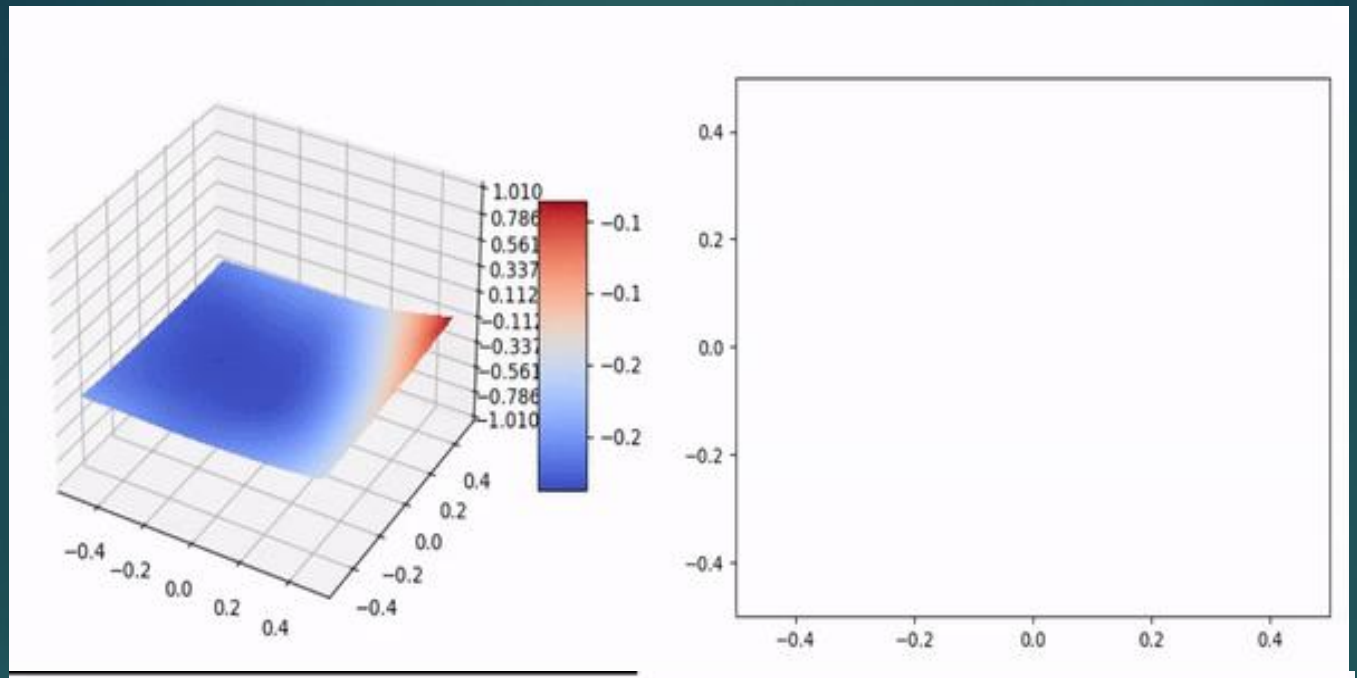


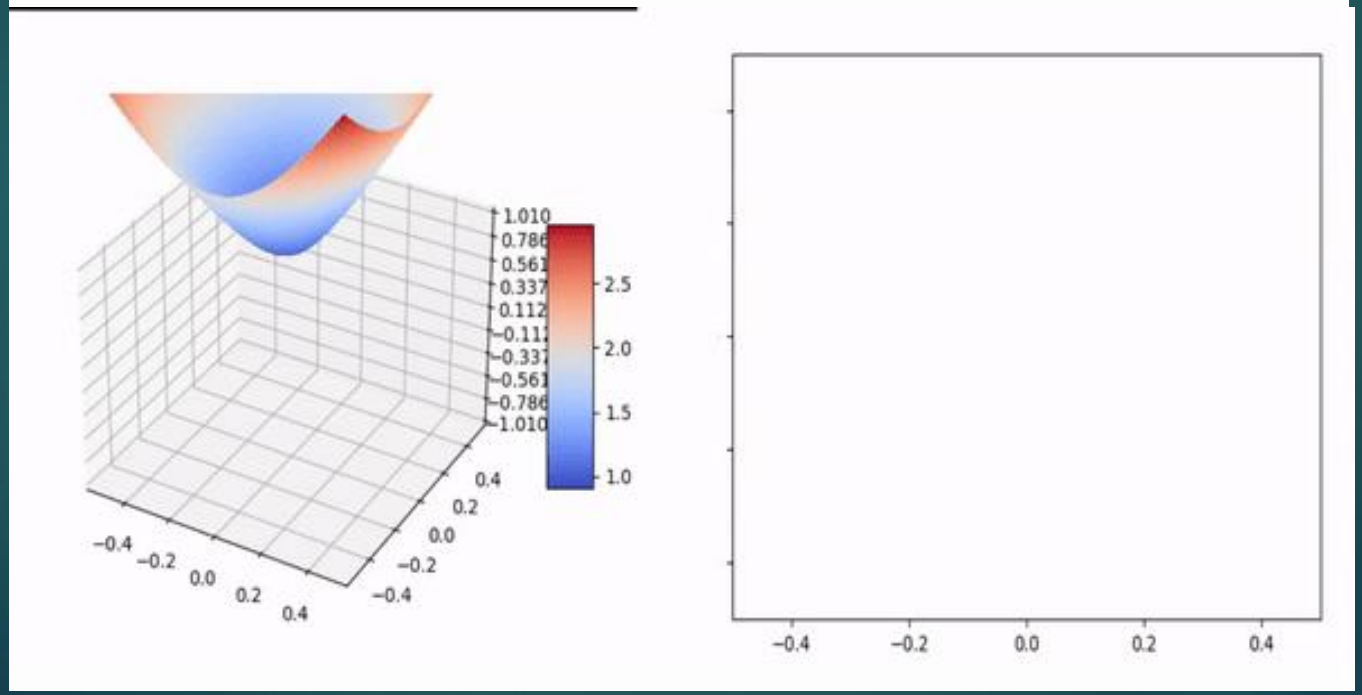

0.4
0.2
0.0
−0.2
−0.4
0.5
-0.5
0.0
−0.4
−0.2
0.0
0.2
0.4

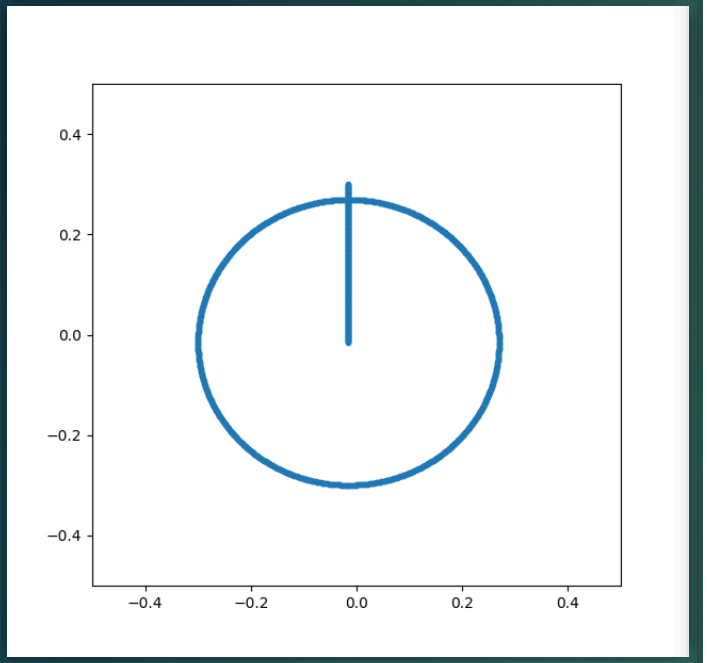
MM

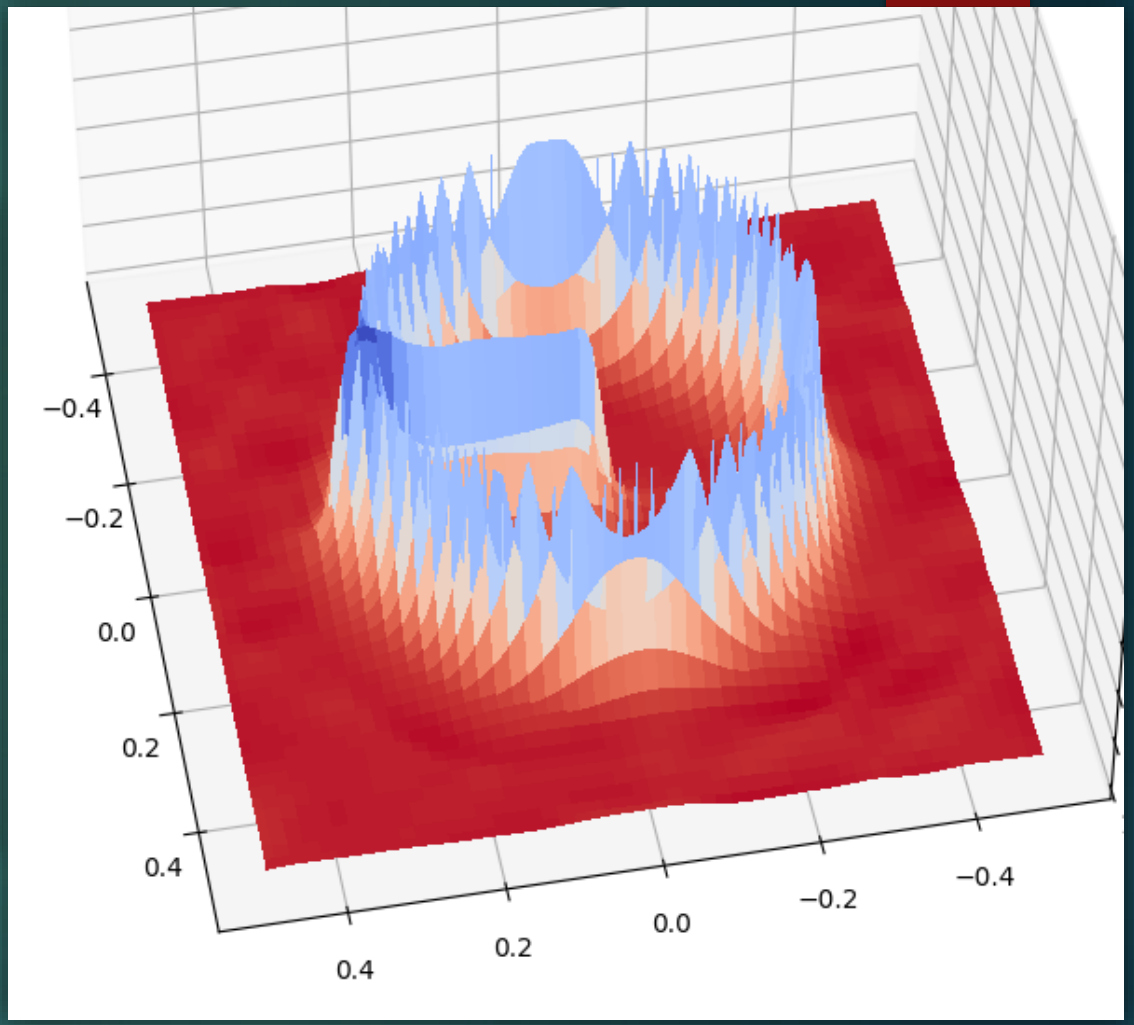

0.4
0.2
0.0
−0.2
−0.4
0.5
0.5
−0.4
−0.2
0.0
0.2
0.4

AT


1.010
0.786
0.561
0.337
0.112
−0.111
−0.337
−0.561
−0.786
−1.010
−0.1
−0.1
−0.2
−0.2
−0.4
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4


1.010
0.786
0.561
0.337
0.112
−0.111
−0.337
−0.561
−0.786
−1.010
2.5
2.0
1.5
1.0
−0.4
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4





1000 P.