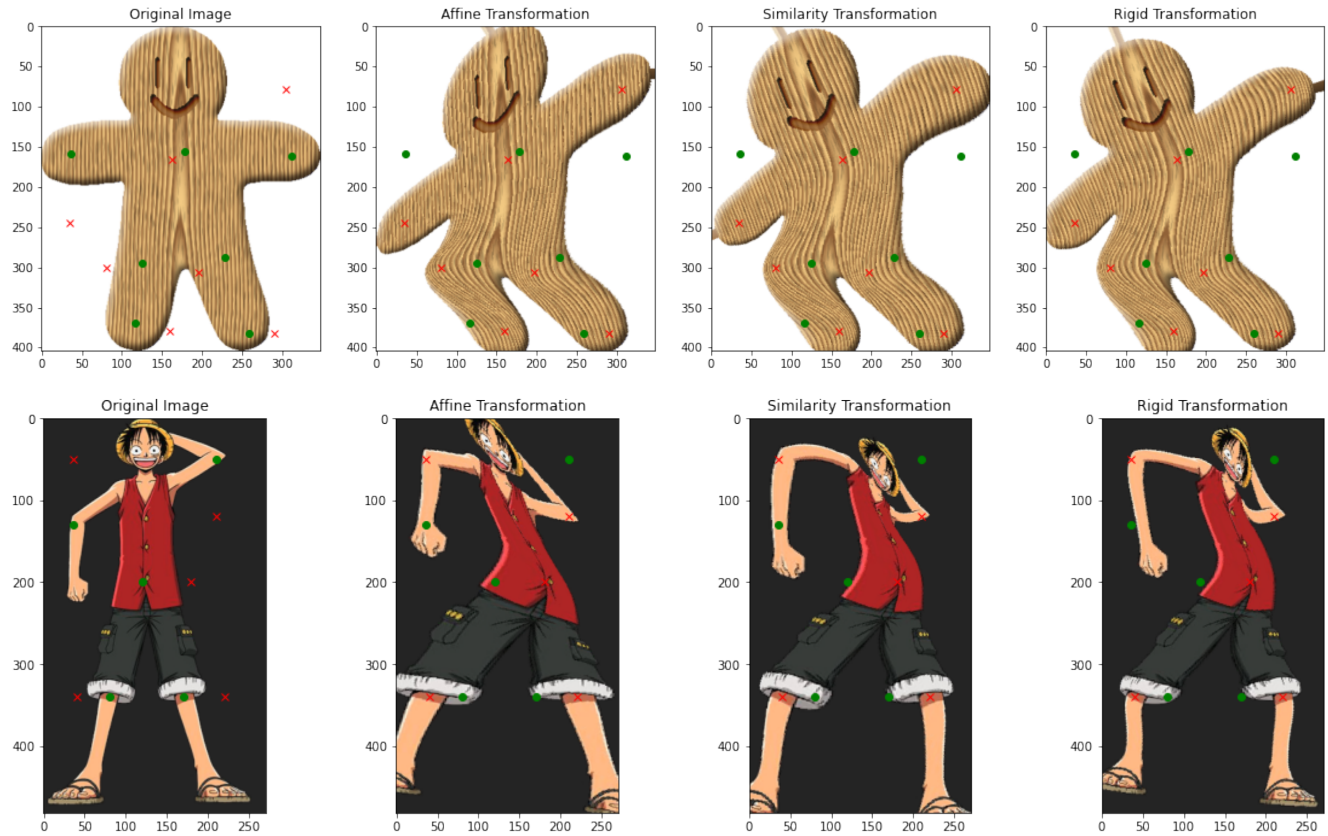
EXERCISE 3 - MLS FOR CURVES, MESHES AND IMAGES

Please add the image in the zip folder to the `data` folder and run the `ipynb` file.

Task 1: Image deformation using MLS

- The code is in the `ipynb` file
- The result of transformation is shown below:



Task 2.1/2.2: Curve and Surface Reconstruction Using MLS

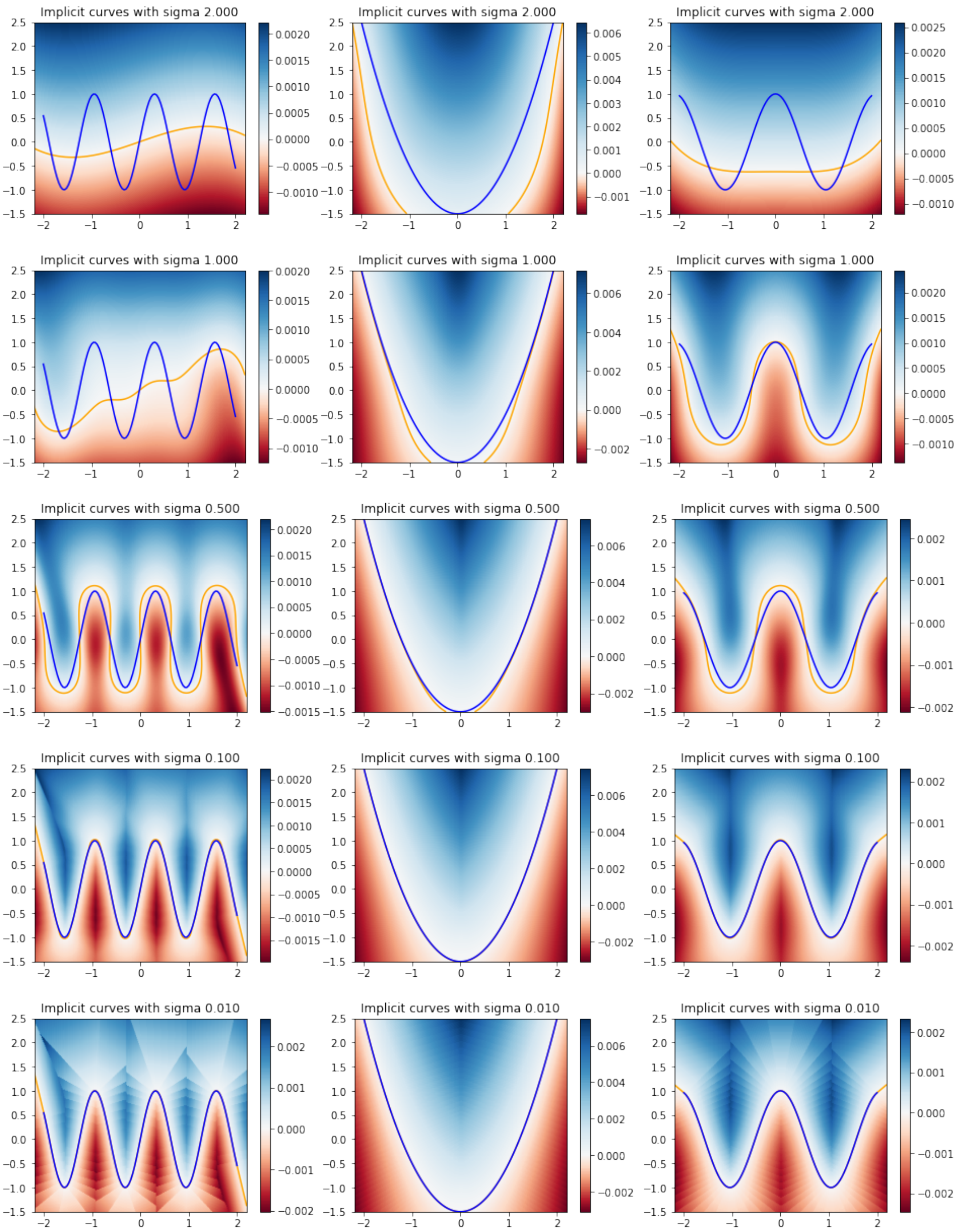
- Implicit function derivation
  - The objective function can be expressed in the following form:

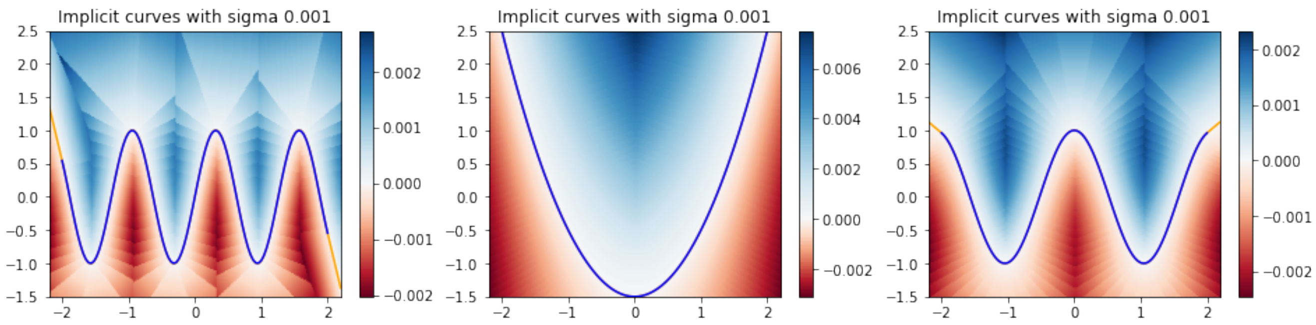
$$\underset{c_0}{\operatorname{argmin}}\, f(c_0) = \underset{c_0}{\operatorname{argmin}} \sum_i \phi_i(\mathbf{x})(c_0 - \mathbf{n_i}^T(\mathbf{x} - \mathbf{x_i}))^2 \tag{1}$$

  - If we take the derivative of the objective function f w.r.t c_0:

$$\begin{aligned} \frac{\delta f}{\delta c_0} &= \sum_i 2\phi_i(\mathbf{x})(c_0 - \mathbf{n_i}^T(\mathbf{x} - \mathbf{x_i})) = 0 \\ 2c_0 \sum_i \phi_i(\mathbf{x}) &= 2\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x} - \mathbf{x_i}) \\ f(\mathbf{x}) = c_0 = \frac{\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x} - \mathbf{x_i})}{\sum_i \phi_i(\mathbf{x})} &= \sum_i \frac{\phi_i(\mathbf{x})}{\sum_i \phi_i(\mathbf{x})}\mathbf{n_i}^T(\mathbf{x} - \mathbf{x_i}) \end{aligned} \tag{2}$$

- The visualization of the grid values estimated by MLS with different sigma is shown below:


Implicit curves with sigma 2.000
Implicit curves with sigma 2.000
Implicit curves with sigma 2.000
Implicit curves with sigma 1.000
Implicit curves with sigma 1.000
Implicit curves with sigma 1.000
Implicit curves with sigma 0.500
Implicit curves with sigma 0.500
Implicit curves with sigma 0.500
Implicit curves with sigma 0.100
Implicit curves with sigma 0.100
Implicit curves with sigma 0.100
Implicit curves with sigma 0.010
Implicit curves with sigma 0.010
Implicit curves with sigma 0.010



- Bonus: The bonus is achieved by replacing the orignal ϕ function by a numerical stable softmax function. The softmax function is implemented as following:

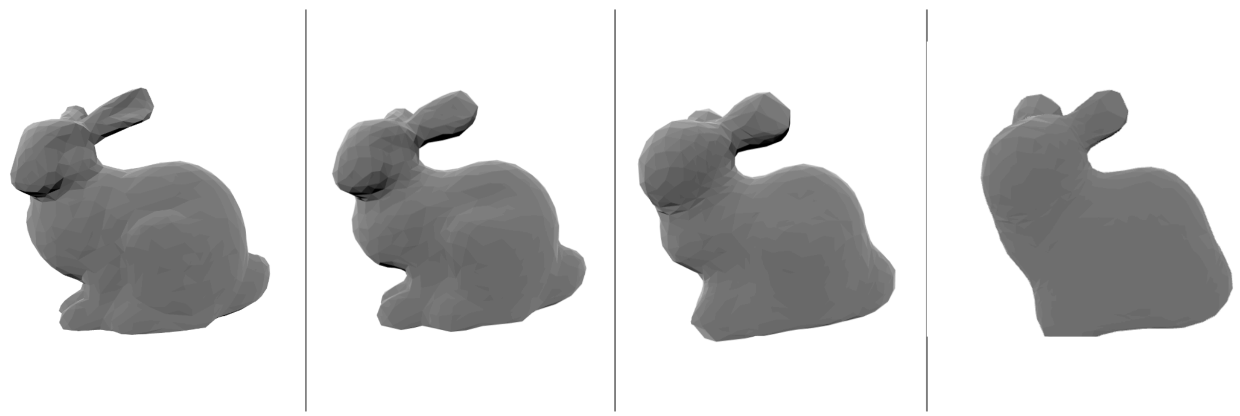
```
phi = np.exp(x - np.max(x))
softmax = phi / np.sum(phi)
```

Task 3: Curve and Surface Reconstruction Using MLS

- Derivation of gradient of function f

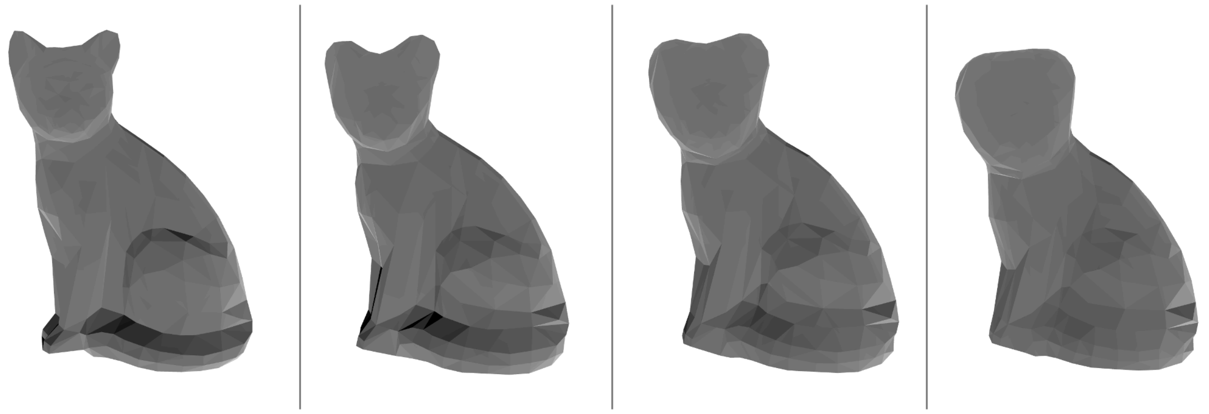
$$
\begin{aligned}
f(\mathbf{x}) = c_0 = \frac{\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i})}{\sum_i \phi_i(\mathbf{x})} &= \sum_i \frac{\phi_i(\mathbf{x})}{\sum_i \phi_i(\mathbf{x})}\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}) \\
\nabla f(\mathbf{x}) &= \nabla \frac{\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i})}{\sum_i \phi_i(\mathbf{x})} \\
&= \frac{\nabla(\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}))(\sum_i \phi_i(\mathbf{x})) - (\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}))\nabla(\sum_i \phi_i(\mathbf{x}))}{(\sum_i \phi_i(\mathbf{x}))^2} \\
&= \frac{(\sum_i \nabla\phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}) + \sum_i \phi_i(\mathbf{x})\mathbf{n_i})(\sum_i \phi_i(\mathbf{x})) - (\sum_i \phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}))\nabla(\sum_i \phi_i(\mathbf{x}))}{(\sum_i \phi_i(\mathbf{x}))^2} \\
&= \frac{(\sum_i \nabla\phi_i(\mathbf{x})\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}) + \sum_i \phi_i(\mathbf{x})\mathbf{n_i}) - f(\mathbf{x})\nabla(\sum_i \phi_i(\mathbf{x}))}{\sum_i \phi_i(\mathbf{x})} \\
&= \frac{\sum_i(\nabla\phi_i(\mathbf{x})(\mathbf{n_i}^T(\mathbf{x}-\mathbf{x_i}) - f(\mathbf{x})) + \phi_i(\mathbf{x})\mathbf{n_i})}{\sum_i \phi_i(\mathbf{x})}
\end{aligned} \tag{3}
$$

- The result is shown below
  - `bunny.off`



original | $\sigma = 0.01$ | $\sigma = 0.02$ | $\sigma = 0.03$

  - `cat.off`



original | $\sigma = 70$ | $\sigma = 100$ | $\sigma = 130$