**Report questions**

**Q1** - The polynomial part is not needed because the Gaussian radial basis function (RBF) is positive definite, which means that Gaussian kernel matrix is by definition non-singular, which would be otherwise guaranteed with the inclusion of a polynomial part for other kinds of RBFs.

**Q2 –** Linear algebra used to fit spline, using *n* number of point-landmarks (control points) and in *k* dimensions: $(\boldsymbol{K} + \lambda \boldsymbol{W^{-1}}) * \boldsymbol{\alpha} = \boldsymbol{q}_k$ **,** where $K_{ij} = R(||p_i - p_j||)$ i.e., a *n* x *n* submatrix representing the Gaussian Kernel containing the RBF interpolation from the given control points; $\boldsymbol{W} = diag\left\{\frac{1}{\sigma_1^2}, \dots, \frac{1}{\sigma_n^2}\right\}$ with $\sigma_i$ represents the control points' localization error although in our case $\boldsymbol{W}$ will be identity matrix; λ is a general weighting parameter; **α** is a vector containing the spline coefficients; $\boldsymbol{q}_k$ is a vector containing the kth coordinate of the target landmarks (displaced control points).

To solve the above equation i.e. linear system: $\boldsymbol{\alpha} = (\boldsymbol{K} + \lambda \boldsymbol{W})^{-1} * \boldsymbol{q}_k$ **.**

**Q3 -** In python, one can use the **solve()** numpy function to find the coefficients, as it is efficient and simple to use.

**Q4 –** The control points are the point-landmarks in the source image i.e. non-displaced control points. These points need to be the points we used to fit the spline to ensure we are evaluating the right interpolation function.

**Q5 –** We do not need the weighting parameter lambda as it is not included in the equation used for evaluation: $R_s(x) = \sum_{i=1}^{n} \alpha_i * R(||\boldsymbol{x} - \boldsymbol{p}_i||)$

**Q6** – Python has good broadcasting computation and therefore we can take advantage of this to perform multi-dimensional matrix multiplication. I have not needed to use this however, as I used very small point sets to start with.

**Q7 –** Sigma is important because it provides information about the strength of the displacement occurring as it represents the amount of variation allowed around the mean of the Gaussian.

**Q8 –** To represent a biophysically plausible deformation, the control points should move minimally. Perhaps a reasonable distance for the points to move would be between 0 and 50mm (relative to a total size of 167x167x69 mm roughly) which we can then transform into voxel size using the voxel dimensions. A constraint could be that the displacement distance has to be a factor of the voxel size in that dimension to make the transformation easier. Also, a control point at the border of the image should be fixed, hence experiencing no displacement.

**Q9 –** I would say yes to some extent as the interpolation method does not perfectly simulate what kind of deformations might occur in real patients.

**Q10** – Steps to compute a warped 3D image:

1. Set a number and position of control points (called landmarks) in the source image.
2. Randomly displace these control points in a biologically plausible manner.
3. Fit a spline function representing the random displacement of the control points and hence obtain the spline coefficients for each x,y,z dimension.
4. Set a query pointset representing the points wished to be displaced.
5. Get the coordinates of the transformed query pointset from the target image using the radial basis interpolation function where coefficients were previously estimated in step 3.

6. Implement the transformed query point grid coordinates to the source image to obtain the target warped image.

**Q11 –** Despite my best efforts I was not able to get to the end of this part and therefore could not visualise. However, with a bit of imagination I believe it is likely I would have seen uneven borders making certain organs look unrealistic. This supports my idea of the limitations of the method in producing completely biologically plausible deformations.

**Q12** – Again using my intuition:

- Changing the random generator strength parameter will I think produce less biologically plausible images with much higher randomness and vis versa.
- Changing the number of control points will I think improve the end result if the number is increased as the spline will be better fitted and vis versa.
- Changing the Gaussian kernel parameter will affect the precision of the Gaussian kernel but I don't expect it to make much difference.