AN INITIAL-MORPHOGENESIS TECHNIQUE OF FREE-FORM SHELL ROOFING BASED ON A FOURIER TRANSFORM

Kiichiro SAWADA

Prof., Dept. of Architectural Design, Shimane Univ., Matsue, Shimane, JAPAN, kich@riko.shimane-u.ac.jp

Editor's Note: Manuscript submitted 16 April 2022; revisions received 25 September 2022 and 08 April 2023; accepted 19 June 2023. This paper is open for written discussion, which should be submitted to the IASS Secretariat no later than March 2024.

DOI: https://doi.org/10.20898/j.iass.2023.015

ABSTRACT

NURBS can create free-form shells by specifying a number of control points and their weights. However, it is challenging to create a form that strictly passes through all the specified control points even with increased weights. This paper proposes an initial-morphogenesis technique of free-form shell roofing using a Fourier Transform. The technique can create forms that strictly pass through all the specified control points. The comparison between the proposed technique and NURBS is discussed and three examples are demonstrated.

Keywords: *Initial-morphogenesis, Free-form shell roofing, Discrete Fourier Transform*

1. INTRODUCTION

Starting with The Guggenheim Museum of Frank O. Gehry in Bilbao, various free-form shells have been designed and constructed worldwide in the last 30 years [1-3]. Many of them seem to start from designer's preferred initial shape and subsequent engineering process. Interpolation by NURBS [4,5] is one of the effective digital tools for representing designer's shapes. NURBS can create free-form shells by specifying a number of control points and their weights. Moreover, various optimization algorithms are applied to a structural morphogenesis for free-form shells using NURBS [6,7]. However, it is challenging to create a form that strictly passes through all the specified control points even with increased weights, using NURBS. In this work, we propose an initial-morphogenesis technique for free-form shell roofing using a discrete Fourier Transform (DFT) [8,9]. The technique can create forms that strictly pass through all the specified control points. Previous studies demonstrated an application to cyclic structures using Fourier series [10] and topology optimization by inverse Fourier transform [11-13]. The DFT-based technique proposed in this work is different from previous studies in that the initial preferred form can be generated with a relatively low-order DFT and inverse transform. The outline of the article is as follows: Section 2 gives the procedure of the DFT-based technique; Section 3 discusses the comparisons between the technique and NURBS; Section4 provides three examples; Section 5 gives the concluding remarks.

2. PROCEDURE OF THE DFT-BASED TECHNIQUE FOR MORPHOGENESIS

Figure1 shows the procedure of a morphogenesis technique using DFT. After specifying the control points' coordinate values or control magnification, DFT is performed on the sequence or matrix of the control point information, and zero values are added as higher-order components to the generated DFT

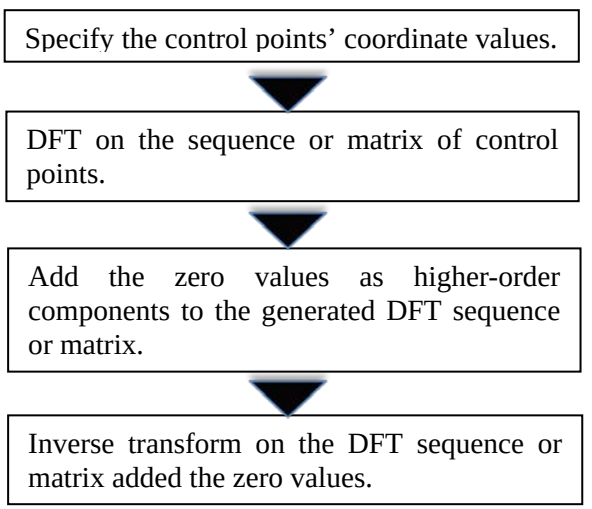


Figure 1: *Procedure of a morphogenesis technique*

sequence or matrix. Finally, by inverse transform, morphological creation can be synthesized with relatively low-order sine waves.

3. COMPARISON BETWEEN DFT BASED TECHNIQUE AND NURBS

3.1. Comparison of mathematical formulas and one-dimensional numerical sequences

It is obvious from the definition [9] that Fourier transform can generate the data sequence passing through the specified control points. The essential meaning of the Fourier transform is to find the amplitudes of sine waves of various frequencies passing through all given points (here, control points). In addition, a data string passing through designated control points is generated by summing sine waves (inverse Fourier transform) of various frequencies having the obtained amplitudes. For example, z=[28.001+0.0j, -1.652-1.460j, -1.652 +1.460j] can be obtained by substituting the one-dimensional numerical sequence of height coordinates of three control points (n_{fx}=3), that is, Z=[8.232, 10.728, 9.041] into the following DFT formula [9].

$$z_I = \sum_{K=0}^{n_{fx}-1} Z_K e^{-2\pi j(I\cdot K/n_{fx})}$$

(I=0,…n_{fx}-1) (1)

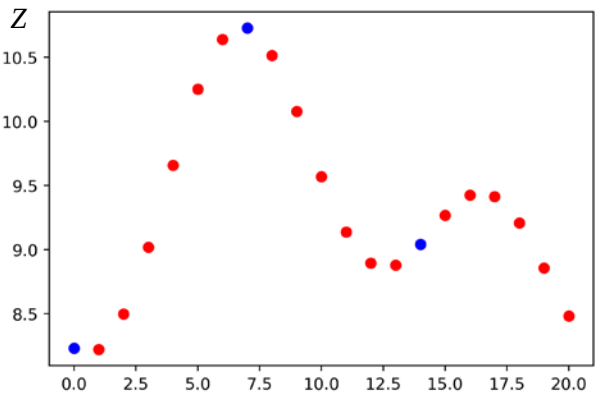
Here, the number of data strings to be generated, n_x, is set to 21. 21 data strings passing through all three control points, Z=[**8.232**, 8.222, 8.499, 9.019, 9.658, 10.250, 10.639, **10.728**, 10.513, 10.079, 9.568, 9.136, 8.895, 8.880, **9.041**, 9.267, 9.424, 9.414, 9.207, 8.856, 8.483] can be obtained as shown in fig.2 by substituting the resulting z into the inverse transform formula below.

$$Z_K = \frac{1}{n_{fx}} \sum_{I=0}^{n_x-1} z_I e^{2\pi j(I\cdot K/n_x)}$$

(K=0,…n_x-1) (2)

However, the x-coordinate of the last control point (relative position in the 21 data strings) is not the final end point, but the position obtained by multiplying the difference between the minimum and maximum values of the x-coordinate by r0x (=0.7) according to the following equation.

$$r_{0x} = \frac{n_{fx}-1}{n_{fx}} \cdot \frac{n_x}{n_x-1} \quad (3)$$



Coordinates corresponding to index number of Z

Figure 2: *Obtained 21 data strings (red circles) and control points (blue circles) by Eqs.(1) and (2)*

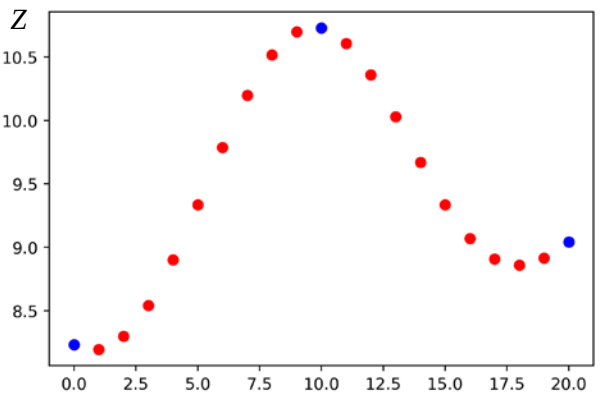
If it is desired to set the control point to the endpoints at both ends and the midpoint equally spaced, it is necessary to expand the wavelength by 1/r0x times for inverse transform as the following equation.

$$Z_K = \frac{1}{n_{fx}} \sum_{I=0}^{n_x-1} z_I e^{2\pi j(r_{0x}\cdot I\cdot K/n_x)}$$

(K=0,…n_x-1) (4)

Using Eq.(3) and Eq.(4) for the above problem, the following data sequence can be obtained as shown in fig.3.

Z = [**8.232**, 8.195, 8.298, 8.541, 8.900, 9.334, 9.786, 10.198, 10.516, 10.699, **10.728**, 10.606, 10.358, 10.028, 9.669, 9.334, 9.070, 8.909, 8.860, 8.914, **9.041**]



Coordinates corresponding to index number of Z

Figure 3: *Obtained 21 data strings (red circles) and control points (blue circles) by Eqs.(1) and (4)*

Appendix A shows the Python program for the above computation.

According to [5], A NURBS curve C(u) is a vector valued piecewise rational polynomial function of the form

$$C(u) = \sum_{i=0}^{n} R_{i,p}(u) P_i \qquad (5)$$

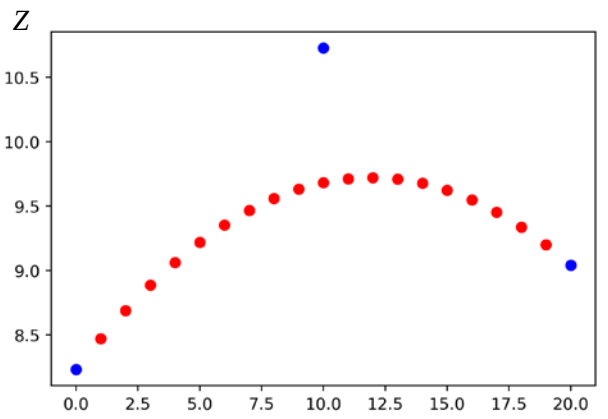
where

$$R_{i,p}(u) = \frac{w_i N_{i,p}(u)}{\sum_{j=0}^{n} N_{j,p}(u) w_j} \qquad (6)$$

where P_i are the control points forming a control polygon, w_i are the weights and $R_{i,p}(u)$ are the p^{th} degree rational basis functions defined over a non-uniform knot vector $u \in [0\ 1]$ with u as non-dimensional curve parameter [5].

Equation (5) shows that the NURBS curve is in the form of a coefficient sum of control point coordinate vectors. Therefore, it is difficult to realize a curve passing through all control points unless each coefficient, $R_{i,p}(u)$ is unrelated to other coefficients.

Figure 4 shows the obtained 21 data one-dimensional strings (red circles) and control points (blue circles) by Eqs.(5) and (6). The given control points, [8.232, 10.728, 9.041] are same with the above example by DFT based technique. The weights, w_i are all 1.0. The knot vector u is [0, 0, 0, 1, 1, 1]. The numerical values of 21 data strings are [**8.232**, 8.471, 8.690, 8.887, 9.063, 9.219, 9.353, 9.467, 9.560, 9.631, 9.682, 9.712, 9.721, 9.710, 9.677, 9.623, 9.549, 9.453, 9.337, 9.200, **9.041**]. The obtained data strings pass through the end control points but not pass through the middle control point.



Coordinates corresponding to index number of Z

Figure 4: *Obtained 21 data strings (red circles) and control points (blue circles) by Eqs.(5) and (6)*

3.2. Comparison of generated surfaces

Figures 5-7 show curved surfaces with height coordinates (red circles) on the XY plane over a 20 × 20 grid, generated by the DFT based technique. The number of control points is set to nfx = 3 in the X direction and nfy = 3 in the Y direction, for 9

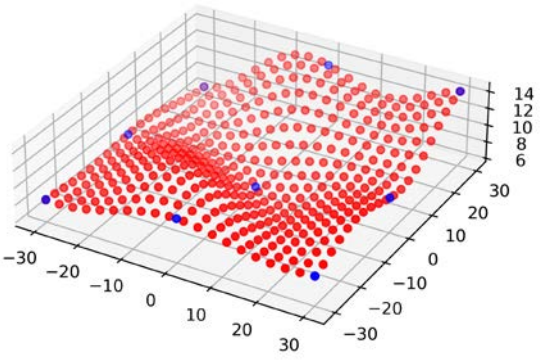


Figure 5: *Generated surface (DFT based technique, nfx=3, nfy=3, random sequence#0)*

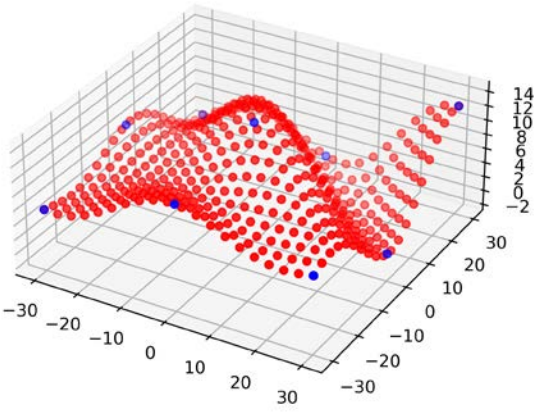


Figure 6: *Generated surface (DFT based technique, nfx=3, nfy=3, random sequence#1)*

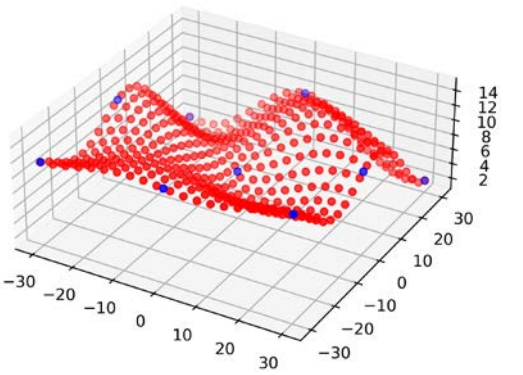


Figure 7: *Generated surface (DFT based technique, nfx=3, nfy=3, random sequence#2)*

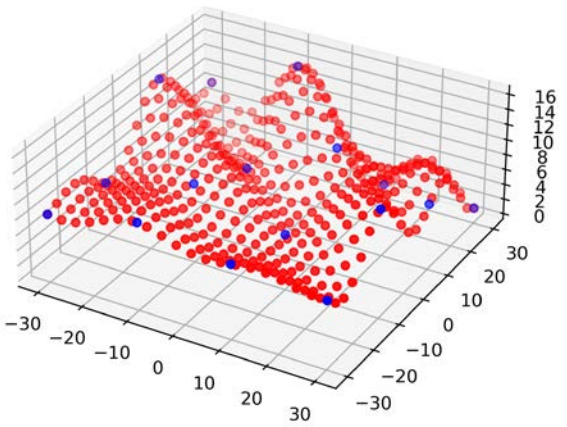


Figure 8: *Generated surface (DFT based technique, nfx=4, nfy=4, random sequence#0)*

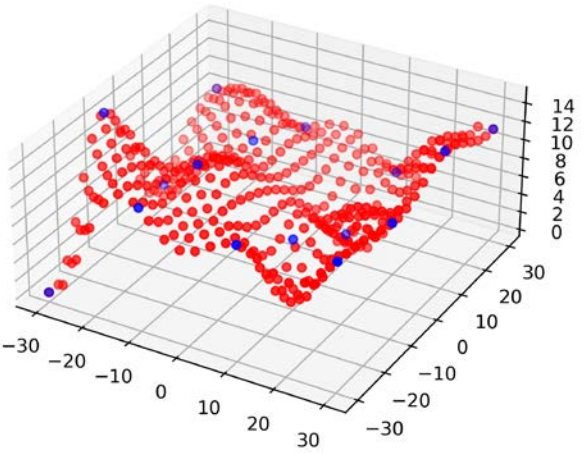


Figure 9: *Generated surface (DFT based technique, nfx=4, nfy=4, random sequence#1)*

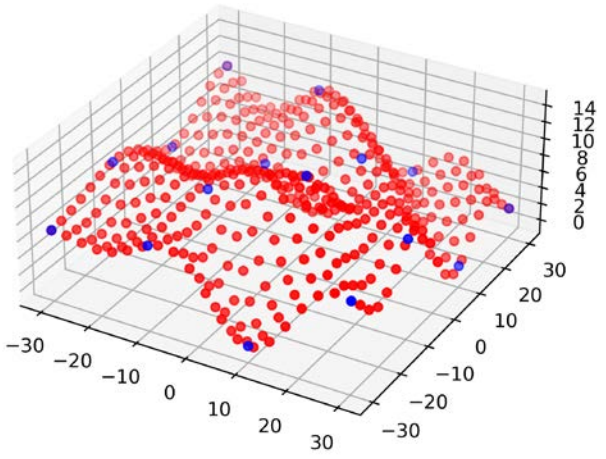


Figure 10: *Generated surface (DFT based technique, nfx=4, nfy=4, random sequence#2)*

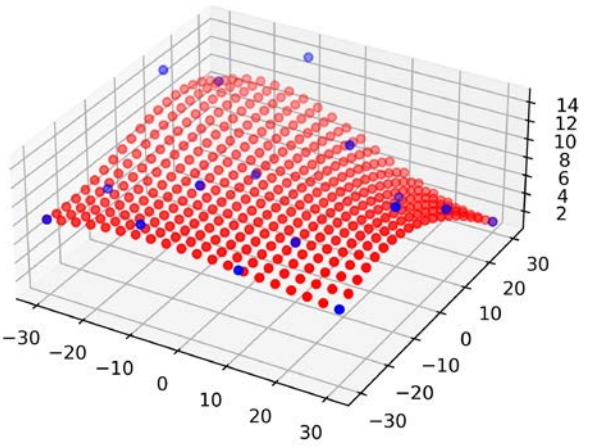


Figure 11: *Generated surface (NURBS, nfx=4, nfy=4, weighting factors=1.0, random sequence#0)*

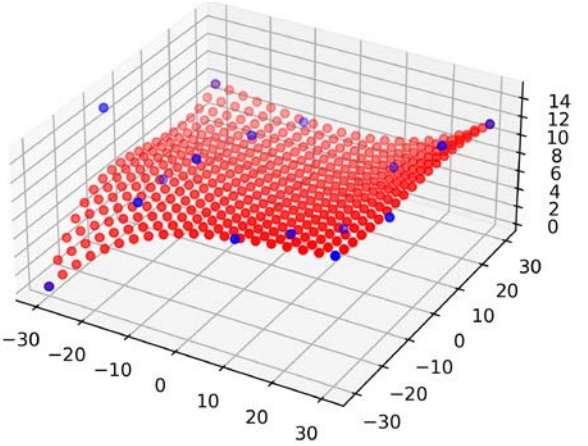


Figure 12: *Generated surface (NURBS, nfx=4, nfy=4, weighting factors=1.0, random sequence#1)*

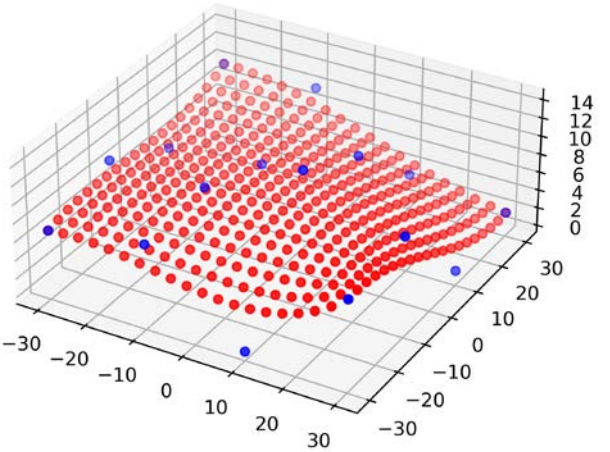


Figure 13: *Generated surface (NURBS, nfx=4, nfy=4, weighting factors = 1.0, random sequence#2)*

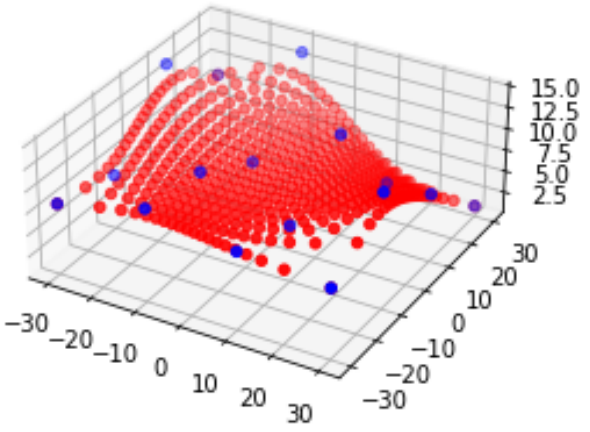


Figure 14: *Generated surface (NURBS, nfx=4, nfy=4, weighting factors=5.0, random sequence#0)*

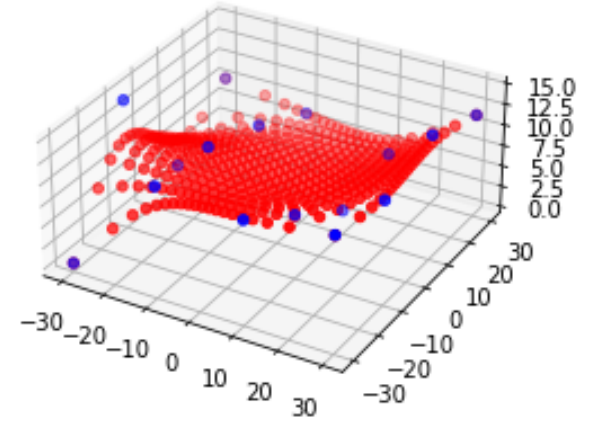


Figure 15: *Generated surface (NURBS, nfx=4, nfy=4, weighting factors=5.0, random sequence#1)*

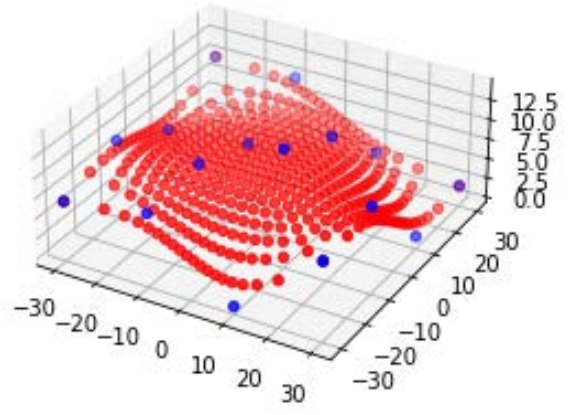


Figure 16: *Generated surface (NURBS, nfx=4, nfy=4, weighting factors=5.0, random sequence#2)*

points (blue circles), which are generated by using pseudo random numbers from 0.0 to 15.0 to validate on geometrical and structural (section 3.3) aspects against any combination. Figures 8 to 10 also show curved surfaces generated by the DFT based technique. The conditions are the same as those shown in Figures 5 to 7, except that nfx = 4 and nfy = 4. However, in the discrete inverse transform in the procedure in Section 2, in order to make the outer control point position correspond to the grid endpoint of the outermost edge, the following inverse transform equation with the correction coefficient r_{0x} and r_{0y} is used.

$$Z_{KL} = \frac{1}{n_{fx} \cdot n_{fy}} \cdot \sum_{I=0}^{n_x-1} \sum_{J=0}^{n_y-1} z_{IJ} e^{2\pi j (r_{0x} \cdot I \cdot K/nx + r_{0y} \cdot J \cdot L/ny)}$$

(K=0,…n_x-1, L=0,…,n_y-1) (7)

$$r_{0x} = \frac{n_{fx}-1}{n_{fx}} \cdot \frac{n_x}{n_x-1} \quad (8)$$

$$r_{0y} = \frac{n_{fy}-1}{n_{fy}} \cdot \frac{n_y}{n_y-1} \quad (9)$$

Here, n_{fx} is total number of control points in the X direction, n_{fy} is total number of control points in the Y direction, n_x is total number of grid nodes in the X direction, and n_y is total number of grid nodes in the Y direction.

The results showed that created surfaces pass through all the specified control points.

Figures 11-13 show the curved surfaces generated by NURBS [4]. The number of control points are set to nfx=4 in the X direction and nfy=4 in the Y direction, for 16 points, also generated using the pseudo random numbers. Degrees and knot vectors of each direction are respectively 3 and [0,0,0,0,1,1,1,1]. The weighting factors are 1.0 in both the corners and the middle. Comparing figures 8-10 with figures 11-13, it is visually observed that DFT based technique gives a larger curvature than NURBS. Figures 14 to 16 also show curved surfaces generated by NURBS. Conditions are the same as those in Figures 8-10 except that weighting factors are 5.0 in the middle control points. The control points are some distance away from the generated surface even for higher weighting factors. Moreover, as the weighting factors increase, the grid points appear to deviate from the even arrangement while the generated curved surface approach the control points.

3.3. Comparison of structural performance

Figures 17-20 show average vertical displacements over all the nodes of curved surfaces generated by random control points. They have same condition with section 3.2. The displacements are obtained by FEM analysis software Lisa 8.0 [14]. Four nodes shell element is adopted. Assuming concrete material, Young's modulus of 20000 MPa, Poisson's ratio of 0.2, and the density of 2400 kg/m^3 are used. The edge nodes (X=-30(m) or X=30(m)) are fixed in all the direction. The shell thickness is 300 mm, and the load condition is the weight of the concrete. The horizontal axes in the Figures 19 and 20 are maximum or minimum of Gauss curvatures of the shell surfaces. The Gauss curvatures of the shell surfaces are calculated by Rhinoceros ver.7 software after fitting the discrete nodal coordinates to NURBS functions on the software. It seems from the figures that the DFT surfaces have smaller vertical displacements than the NURBS surfaces. However, for both the surfaces, larger the number of control points given, smaller the vertical displacements obtained as well as higher the Gauss curvatures. Especially for NURBS, it seems that larger number of control points is needed to obtain higher curvatures of surfaces or smaller vertical displacements than DFT based technique.

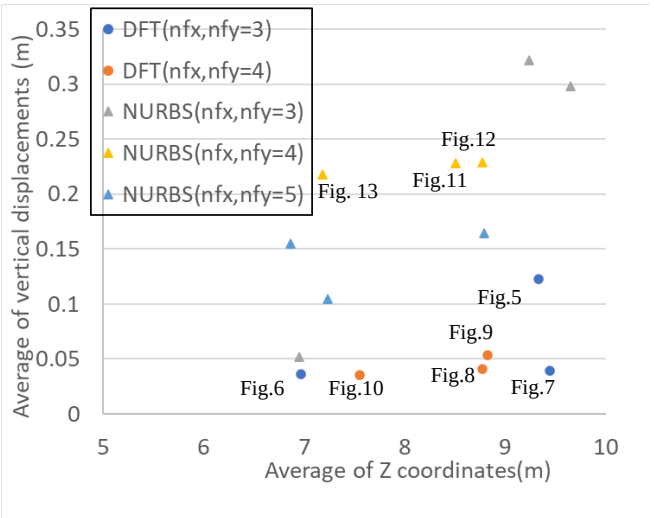


Figure 17: *The relationship between average of vertical displacement and average of Z coordinates*

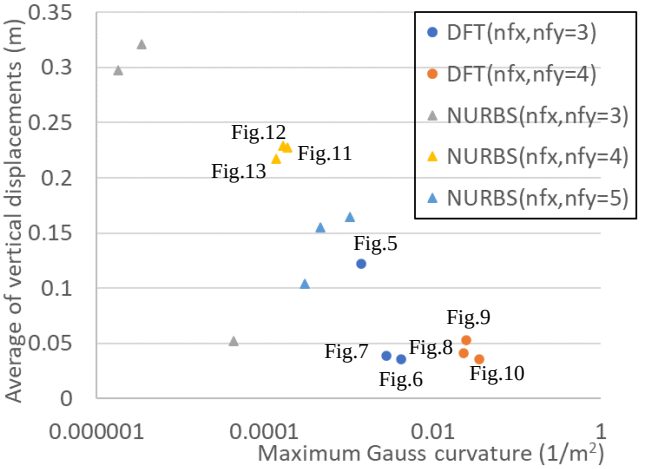


Figure 19: *The relationship between average of vertical displacement and Maximum Gauss curvature*

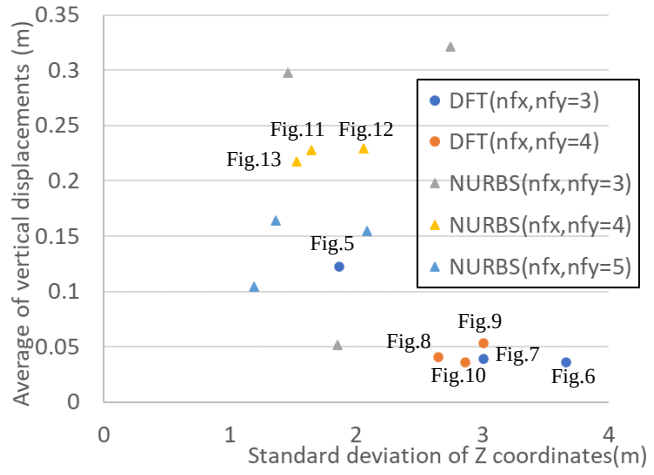


Figure 18: *The relationship between average of vertical displacement and standard deviation of Z coordinates*

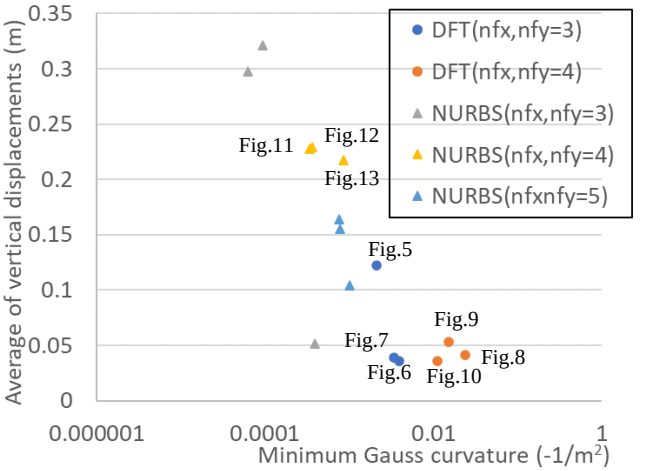


Figure 20: *The relationship between average of vertical displacement and Minimum Gauss curvature*

4. EXAMPLES FOR MORHPOGENESIS OF FREE-FORM SHELL ROOFING

4.1. Example 1

(1) Basic circular grid plane

In the X–Y plane, a polar coordinate is set with the distance R from the origin and the rotation angle P, as shown in Figure 21, and a basic circular grid with an inner radius (l_0 = 5 (m)) and an outer radius (l_1 = 20 (m)) is generated. The grid has 72 nodes in the angular direction and 11 nodes in the radial direction, as shown in Figure 22.

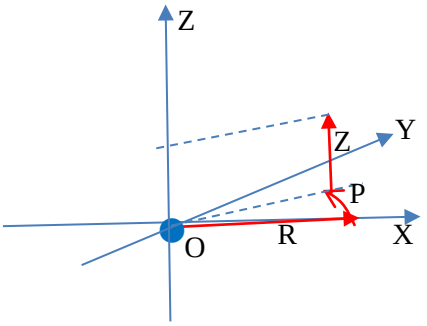


Figure 21: *Coordinate system (X-Y plane and R-P polar system)*

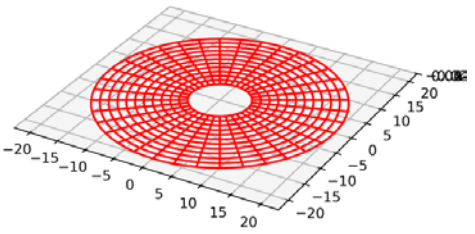


Figure 22: *Basic circular grid plane (Example 1)*

(2) Conversion to arbitrary shaped plane

The control magnifications are set in each angle direction on the basic circular plane, and conversion to an arbitrary shaped plane is conducted by the procedure in Section 2 as shown in Figure 23. In the calculation example here, control is performed in four directions of P = 0°, 90°, 180°, and 270°, and the control magnification in each direction is 4.0, 2.0, 2.5, 1.0.

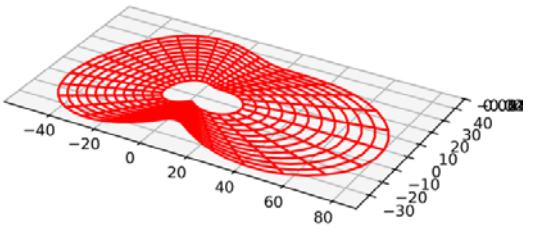


Figure 23: *Conversion to arbitrary shaped plane (Example1)*

(3) Morphogenesis of free-form shell roofing

The height coordinate control values are set in each angular direction in the arbitrary plane, and a free-form surface shell can be generated using the procedure in Section 2, as shown in Figure 24. In the example, the control height at the center side is 10 m, the height at the intermediate position is 15 m, and the height at the outside is 3 m in each direction (0°, 90°, 180°, and 270°).

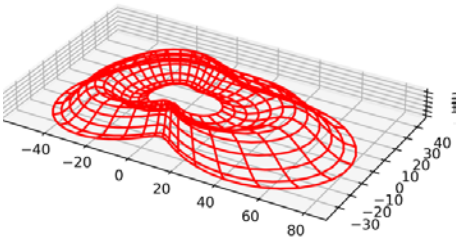


Figure 24: *Free form shell roofing (Example 1)*

To make the outer control point position correspond to the grid endpoint of the outermost edge, the inverse transform equation with the correction coefficient r0 is used:

$$Z_{KL} = \frac{1}{n_{ft} \cdot n_{fr}} \cdot \sum_{I=0}^{n_t-1} \sum_{J=0}^{n_r-1} z_{IJ} e^{2\pi j (I \cdot K/nt + r_0 \cdot J \cdot L/nr)} \quad (K=0,\ldots n_t\text{-}1,\ L=0,\ldots,n_r\text{-}1) \tag{4}$$

$$r_0 = \frac{n_{fr}-1}{n_{fr}} \cdot \frac{n_r}{n_r-1} \tag{5}$$

where, n_{ft} is total number of control points in the angular direction, n_{fr} is total number of control points in the radial direction, n_t is total number of grid nodes in the angular direction, and n_r is total number of grid nodes in the radial direction

The following sequence shows the generated height coordinate values of each grid node (11 nodes) from the center side to the outside in the 0° direction.

10.0, 8.69554, 8.65074, 10.4251, 13.1378, **15.0**, 14.4974, 11.441, 7.18259, 3.83596, **3.0**

The same height coordinate sequence as above was obtained in all angular directions. It is observed that the coordinate height values of the center side, the outer endpoints and the middle exactly correspond to the heights of the specified control point.

(4) FEM analysis of the generated shell roofing

Figures 25 and 26 show the displacement magnitude and maximum principal stress on bottom surface. These are obtained by FEM analysis software Lisa 8.0 [14]. Eight nodes solid element is adopted. Assuming concrete material, Young's modulus of 20000 MPa, Poisson's ratio of 0.2, and the density of 2400 kg/m^3 are used. The nodes on the outer circumference are fixed in all the direction. The shell thickness is 300 mm, and the load condition is the weight of the concrete. The large displacement or stress are locally observed in the red element (low curvature areas) indicated by the black arrows in the figures.

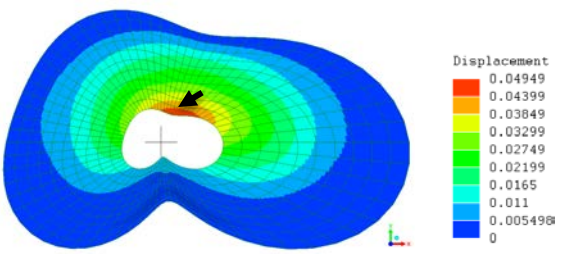


Figure 25: *Displacement magnitude (Example 1)*

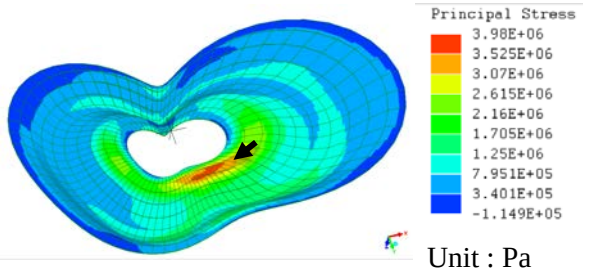


Figure 26: *Maximum principal stress viewed from bottom surface (Example 1)*

4.2. Example 2

Concerning the same basic circular plane as in Example 1, the control is performed in eight directions (0°, 45°, 90°, 135°… 270°, 315°), and the control magnification in each direction (2.0, 4.0, 2.0, 4.0, 2.0, 4.0, 2.0, 4.0) is set and converted to the arbitrary shaped plane shown in Figure 27 using the procedure in Section 2. Next, the height coordinate control values are set in each angular direction in the plane, and the free-form surface shell shown in Fig.28 is generated by the procedure in Section 2. In the calculation example here, in the directions of 0°, 90°, 180°, and 270°, the height on the center side edge is set to 20 m and height on the outside edge is set to 0 m. On the other hand, in the directions of 45 °, 135°, 225° and 315°, the height on the center side is set to 20 m, the height on the outside is set to 30 m. Figures 29 and 30 show the displacement magnitude and maximum principal stress on upper surface over all the element of the generated shell roof structure. The same concrete material is assumed as Example 1. Fixed nodes are indicated by triangular marks in the figures. The shell thickness is 300 mm and the load condition is the same as the weight of the concrete used in Example 1.

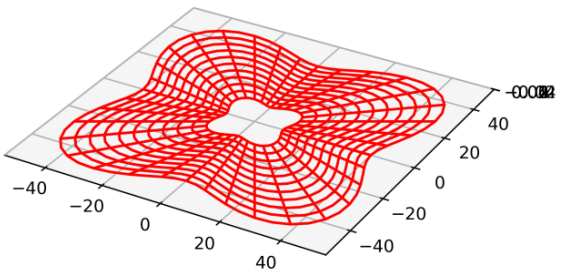


Figure 27: *Conversion to arbitrary shaped plane (Example 2)*

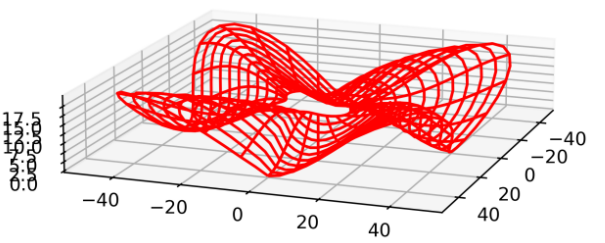


Figure 28: *Free form shell roofing (Example 2)*

Since the shell morphology does not have symmetrical axis but tilts like a propeller, the stress and displacement distribution also does not have symmetrical axis. The large displacement or stress are locally observed in the red region indicated by the black arrows in the figures.

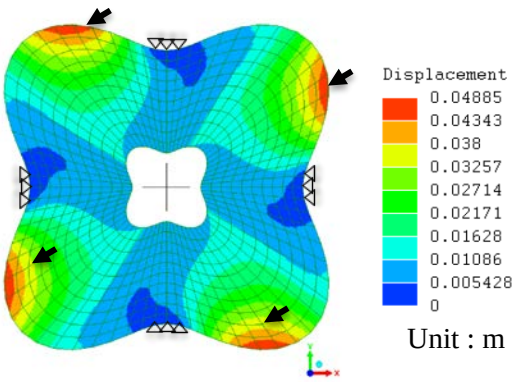


Figure 29: *Displacement magnitude (Example 2)*

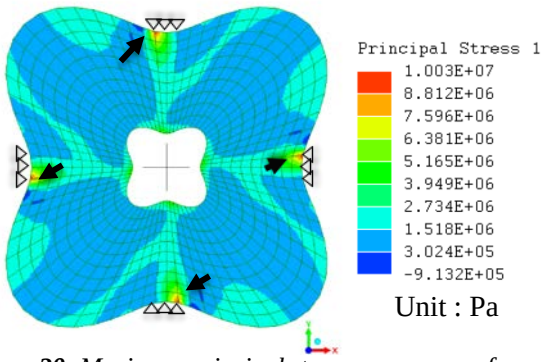


Figure 30: *Maximum principal stress on upper surface (Example 2)*

4.3. Example 3

Also, concerning the same basic circular plane as Example 1 and 2, the control is performed in 3 directions (0°, 120°, and 240°), and the control magnification in each direction is 2.0,3.0, and 2.0 are set and converted to the arbitrary shaped plane shown in Figure 31 by the procedure in Section 2. Next, the height coordinate control values are set in each angular direction in the plane, and the free-form surface shell shown in Figure 32 is generated. In this example, in the direction of 0°, the height on the center side is set to 5 m, the height on the outside is set to 3 m, and the height in the middle is set to 10m. In the direction of 120°, the height on the center side is set to 10 m, the height on the outside is set to 3 m and the height on the middle is set to 15m. In the direction of 240°, the height on the center side is set to 15 m, the height on the outside is set to 3 m and the height on the middle is set to 5m. Figures 33 and 34 show the displacement magnitude and the maximum principal stress on bottom surface over all the elements of the generated shell roof structure. The same concrete material is assumed as Examples

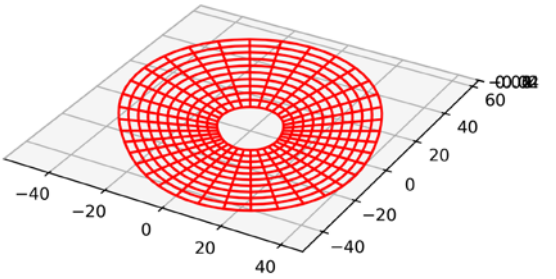


Figure 31: *Conversion to arbitrary shaped plane (Example 3)*

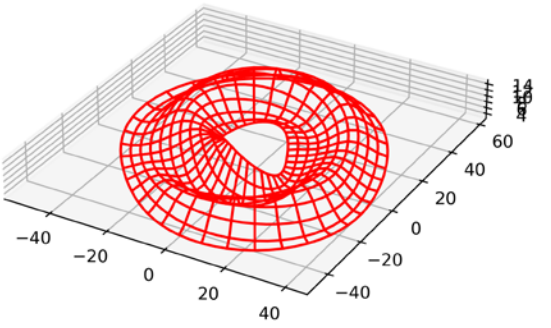


Figure 32: *Free form shell roofing (Example 3)*

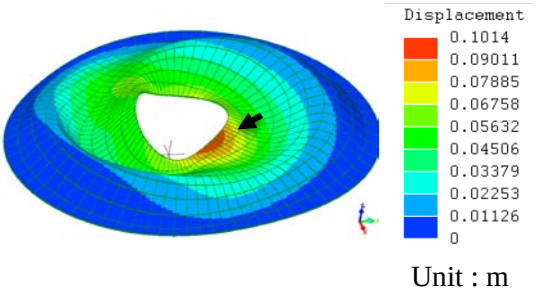


Figure 33: *Displacement magnitude (Example 3)*

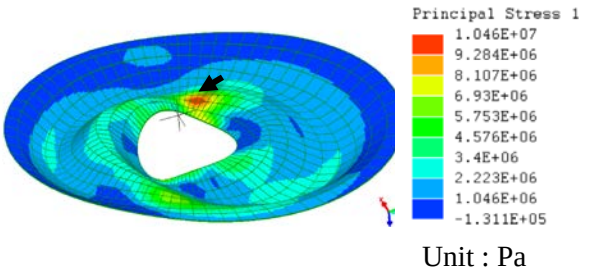


Figure 34: *Maximum principal stress viewed from bottom surface (Example 3)*

1 and 2. The nodes on the outer circumference are fixed. The shell thickness is 300 mm and the load condition equals the weight of the concrete used in Examples 1 and 2. The large displacement or stress is locally observed in the comparatively low curvature area (the red region indicated by the black arrows in the figures).

5. CONCLUSIONS

An initial-morphogenesis technique is proposed for free-form shell roofing with DFT. Concluding remarks can be summarized as follows.

(1) The control points are some distance away from the NURBS's surface even for weighting factor of 5.0. Moreover, as the weighting factor increases, the grid points appear to deviate from the even arrangement while the NURBS's surface approaches the control points.

(2) Contrary to NURBS surface, the surfaces created by DFT based technique pass through all the specified control points.

(3) The control points must be evenly distributed in the DFT-based technique, while, in NURBS, the control points are not necessarily evenly distributed.

(4) The DFT based technique can generate high curvatures or expressive surfaces with simple parameter settings.

(5) For both the surfaces, larger the number of control points given, smaller the vertical displacements obtained as well as higher the Gauss curvatures.

(6) The initial morphology of the shell roofs obtained in the three examples is not ideal from the structural aspect because of the large local stress and deformation on the low curvature areas in FEM analyses. Therefore, a comprehensive design method, which can consider not only the initial form requirements but also other requirements such as structural performance, is required. Multi-objective optimization using these requirements as objective functions is envisioned.

(7) In the examples of section 4, the load condition of the weight of the concrete was just considered. However, it should be noted that a significant proportion of the prime cause of failures is improper loading conditions [1]. Moreover, it is interesting subject that the DFT based technique and NURBS are compared on the aspect of structural performance. Reliability analyses under various loadings is also envisioned for the shells by the two techniques.

ACKNOWLEDGMENTS

Example 1 is inspired by the esquisse plan of Mr. Hikaru Seto, a fourth-year student in the Department of Architectural Design, Interdisciplinary Faculty of Science and Engineering, Shimane University, who visited for consultation on architectural design and drafting. I would also like to express deep gratitude to Professor emeritus Toshio Honma, Associate Professor Yohei Yokosuka, Kagoshima University, and Assistant Professor Nguyen Thu Lan, Shimane University, for valuable discussions.

REFERENCES

[1] M. Majowiecki, "Ethics and structural reliability in free-form design (FFD)," *Journal of the International Association for Shell and Spatial Structures*, 48(4), 29-50, 2007

[2] M. Eekhout, (Ed.), "Free form technology from Delft," (Vol. 14). IOS Press, 2016

[3] M. Sasaki, T. Itō, and A. Isozaki, "Morphogenesis of flux structure," *Aa Publications*, 2007

[4] O. R. Bingol and A. Krishnamurthy, "NURBS-Python: An open-source object-oriented NURBS modeling framework in Python," *SoftwareX*, Vol.9, Pages 85-94, January–June,2019 (DOI: 10.1016/j.softx.2018.12.005)

[5] S. Mohan, S. H. Kweon, D. M. Lee, and S. H. Yang, "Parametric NURBS curve inter-polators: a review," *International Journal of Precision Engineering and Manufacturing*, 9(2), 84-92, 2008

[6] Y. Okita and T. Honma, "Structural Morphogenesis for Asymmetric Free-Form Grid Shell Using NURBS with Manipulation of Decent Solutions Search," *Proceedings of IASS Annual Symposia*, IASS 2016 Tokyo Symposium: Spatial Structures in the 21st Century – Form Finding & Optimization, pp. 1-9(9)

[7] T. Kimura and H. Ohmori, "Computational Morphogenesis of Free Form Shell," *Journal of the International Association for Shell and Spatial Structures*, Vol. 49, No. 3 December n. 159, pp. 175-180(6), 2008

[8] W. C. James and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Math. Comput.* 19: 297-301, 1965 (DOI: 10.1090/S0025-5718-1965-0178586-1)

[9] J. O. Smith, “Mathematics of the discrete Fourier transform (DFT): with audio applications,” *CCRMA*, Stanford, July 2002

[10] Y. Toyoshima, S. Yamada, K. Sato, T. Nagai and M. Araya, “Study on Structural Morphogenesis by Applying of Fourier Series to Circular Structure,” *Summaries of Technical Papers of Annual Meeting*, AIJ, pp.835-836, 2009. (in Japanese)

[11] K. Sawada, “Topology optimization of large deformable elastic plates,” *Journal of Structural and Construction Engineering (Transactions of AIJ)*, Vol.85, No.771, pp.683-692, 2020.5 (in Japanese) (DOI: 10.3130/aijs.85.683)

[12] *K. Sawada, "Seismic Response Analyses of RC Portal Frames with Large Deformable Elastic Braces." Seismic Resistant Structures*: 197, 2018

[13] K. Sawada, K. Kajitani, T. Uno, J. Teramoto, S. Komatsu, “A Study on Multi-Objective Optimization of Large Deformable Elastic Plates,” *Buildings*, 12(9), 1323, 2022 (DOI: 10.3390/buildings12091323)

[14] https://lisafea.com/ (Final access on Jan.10, 2022)

APPDENDIX A: PYTHON PROGRAMMING CODE-SECTION 3.1

The following script is a Python programming code created by the author for DFT based technique described in Section 3.1.

```
"""
DFT curve @author: kiichiro sawada
"""
import matplotlib.pyplot as plt
import numpy as np

i_seed = 0 ; nx = 21 ; nfx = 3
a = np.linspace(0,nx-1,nx)
ia = np.int16(a[:])
a_c = np.linspace(0,nx-1,nfx)

# generate the control points' coordinate values by
# random generator.
np.random.seed(i_seed)
r = 15.0*np.random.random([nfx])
Z_c = r[:]
zf0 = r[:]

# DFT on the sequence of the control points
zf1 = np.fft.fftn(zf0)

# Add the zero values as higher-order components to
# the generated DFT sequence.
zfa = np.array([zf1[i] if i<nfx else 0.0 for i in range(nx)])

def ifftcon(zfa):
    r0x = (nfx-1)/nfx*nx/(nx-1)
    izfs = np.zeros([nx],dtype='complex128')
    for i in range(nx):
        izfs = izfs + zfa[i]*np.exp(2.0j*np.pi*\
            (i*ia[:]/nx*r0x))
    izfs = izfs/nfx
    izfsr = izfs.real
    return izfsr

# Inverse transform on the DFT sequence
Z = ifftcon(zfa)

# Plot figures
fig = plt.figure(dpi=400)
plt.scatter(a,Z, color='red')
plt.scatter(a_c,Z_c, color='blue')
plt.show()
```

APPDENDIX B: PYTHON PROGRAMMING CODE-SECTION 3.2

The following script is a Python programming code created by the author for DFT based technique described in Section 3.2.

```
"""
DFT Surface @author: kiichiro sawada
"""
import matplotlib.pyplot as plt
import numpy as np

i_seed = 0
nx = 20 ; ny = 20
nfx = 3 ; nfy = 3
lx = 30.0 ; ly = 30.0
a = np.linspace(0,nx-1,nx)
b = np.linspace(0,ny-1,ny)
ia = np.int16(a[:])
ib = np.int16(b[:])
iaa,ibb = np.meshgrid(ia,ib)
x = np.linspace(-lx, lx, nx)
y = np.linspace(-ly, ly, ny)
x_c = np.linspace(-lx, lx, nfx)
y_c = np.linspace(-ly, ly, nfy)
X, Y = np.meshgrid(x, y)
X_c, Y_c = np.meshgrid(x_c, y_c)

# Generate the height coordinates for control points
# by pseudo random numbers

np.random.seed(i_seed)
r = 15.0*np.random.random([nfx*nfy])
Z_c = r.reshape(nfy,nfx)
zf0 = r.reshape(nfy,nfx)
#DFT on the matrix of the control points information
zf1 = np.fft.fftn(zf0)

# Add the zero values as higher-order components to
# the generated DFT matrix
zfa = np.array([[zf1[i,j] if j<nfx and i<nfy else 0.0 \
for j in range(nx)] for i in range(ny)])

# Inverse transform on the DFT matrix added the
# zero values
def ifftcon(zfa):
    r0x = (nfx-1)/nfx*nx/(nx-1)
    r0y = (nfy-1)/nfy*ny/(ny-1)
    izfs = np.zeros([ny,nx],dtype='complex128')
    for i in range(ny):
        for j in range(nx):
            izfs = izfs + zfa[i,j]*np.exp(2.0j*np.pi*\
            (i*ibb[:,:]/ny*r0y+j*iaa[:,:]/nx*r0x))
    izfs = izfs/nfx/nfy
    izfsr = izfs.real
    return izfsr
Z = ifftcon(zfa)

# 3D plots by Matplotlib
ix = int(np.amax(X)-np.amin(X))
iy = int(np.amax(Y)-np.amin(Y))
iz = int(np.amax(Z_c)-np.amin(Z_c))
fig,ax= plt.subplots(subplot_kw={'projection': '3d'})
ax.set_box_aspect((ix,iy,iz*2))
ax.scatter(X,Y,Z, color='red')
ax.scatter(X_c,Y_c,Z_c, color='blue')
plt.show()
```