Toolbox Manual

Copyright Notice

Introduction

This toolbox is designed for form finding of large deployable pin-jointed (DPJ), structures, which implements the first part, determination of initial configuration, of the stochastic fixed nodal position method (SFNPM) proposed in Refs. [1-3]. The surface accuracy of the generated mesh geometry of large DPJ structures is evaluated by the methods given in Refs. [4,5]. A user guide is provided in the manual, containing the detail information of functions in the toolbox. The toolbox used large deployable mesh reflectors which is a typical large DPJ structures as an example for illustration of the SFNPM. ***But this toolbox can be easily generalized to apply to form finding of other types of DPJ structures.***

System Requirements for the MATLAB Toolbox

- PC with Win XP or Mac with OS 9.x and up
- The software MATLAB (version 2010 and up) on the computer.

Software Installation and Test

1. Drag the toolbox folder onto a hard disk of the computer;

2. Launch MATLAB and set a working path to the Toolbox folder; and
3. Test the toolbox by typing Demo_offsetpara.m in the MATLAB command window, which will launch a demo program showing how the toolbox works. The demo ends with a message: “The toolbox works properly.”

Optimal Design Procedure for the Toolbox

The Demo_offsetpara.m follows the design procedure in Figure 1 and it shows how to use this toolbox. Another demo, a mesh geometry design for center parabolic shape is in Demo_centerpara.m file.

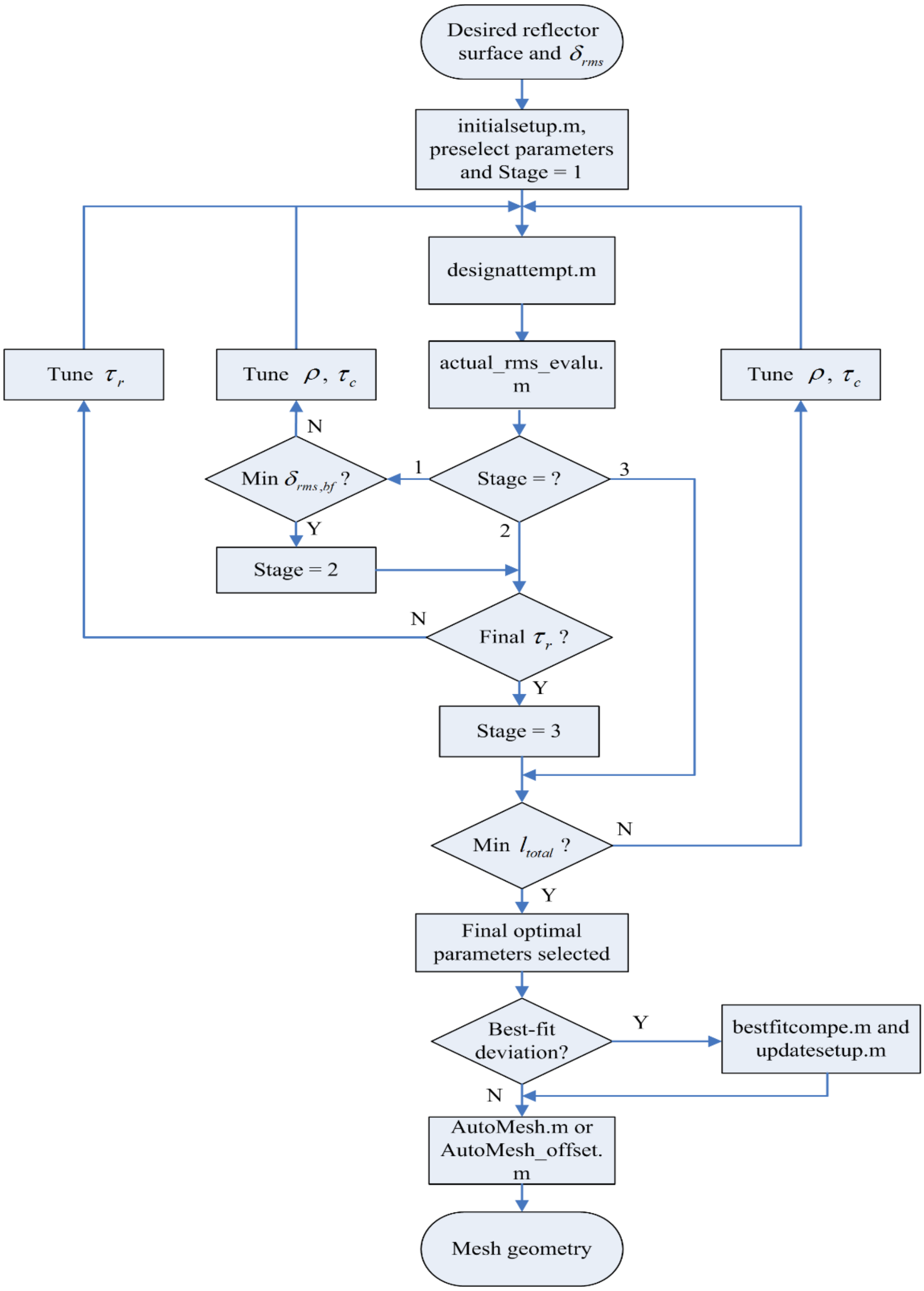

Figure 1 The complete design procedure.

Function Guide

1. MATLAB Function **initialsetup**

Purpose:

To set up the requirements for design of reflector surface geometry

Synopsis:

```
error_rms = initialsetup(F_required, D_required, e_required,
freq_op_required, flag_effect, flag_shape,rms_budget)
```

Inputs:

`F_required` required (parent) focal length.

`D_required` required (parent) diameter length.

`e_required` required offset distance.

`freq_op_required` required operating frequency.

`flag_effect` 0: whole surface is effective; 1: effective region are part of the surface.

`flag_shape` 1: spherical; 2: center parabolic; 3: offset parabolic.

`rms_budget` required surface accuracy budget.

Outputs:

`error_rms` allowable surface RMS error.

Note:

The toolbox is programmed based on the standard ISO units. The data in other units must be transferred into ISO system before it is input into the function.

2. MATLAB Function **designattempt**

Purpose:

To attempt a design implementing the design core twice.

Synopsis:

```
[n_r, n_c, zeta_final, wb_final, MemberL_prop, M_indx, Node_design,
Node_design_global, B_C, Indx_node_load, L_t, L_t_nobc, F, D, e_p] =
designattempt(tao_r, n_s, tao_c, rou, c_b, flag_wb, w_b)
```

Inputs:

`tao_r` the design parameter τ_r for number of rings.

n_s the design parameter n_s for number of subdivisions.

`tao_c` the design parameter τ_c for index of the first converging ring.

`rou` the design parameter ρ for the converging speed.

`c_b` the design parameter c_B for the boundary type.

`flag_wb` 0: the whole surface is effective and w_B is not needed; 1: effective surface is part of the working surface and w_B is needed.

`w_b` the design parameter w_B for the boundary thickness.

Outputs:

`n_r` number of rings.

`n_c` index of the first converging ring.

`zeta_final` final value of design parameter ς in the design attempt.

`wb_final` final value of design parameter w_B in the design attempt.

`MemberL_prop` vector of member lengths.

`M_indx` index matrix of member connectivity.

`Node_design` designed nodal coordinates (locally for the offset parabolic case).

`Node_design_global` designed global coordinates for offset parabolic case.

`B_C` the boundary conditions of designed reflector surface.

$$BC\_Spec = \begin{bmatrix} node\_no & support\_type \\ \vdots & \vdots \end{bmatrix}$$

where *node_no* is the node number of the support and the surface boundary of reflector is fixed or

$$support\_type = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}$$

`Indx_node_load` index vector of nodes except the boundary nodes.

`L_t` total length of members l_{total}.

`L_t_nobc` total length of members within the effective region.

`F` focal length of the whole surface.

`D` diameter of the whole surface.

`e_p` offset distance of the whole surface.

Note:

Please see Figure 1 for the completed design procedure and the role of the function within the procedure.

3. MATLAB Function **actual_rms_evalu**

Purpose:

To measure the exact best-fit surface RMS error.

Synopsis:

```
[error_rms, deta_bestfit, F_bestfit] = actual_rms_evalu(n_r, n_s,
w_b, Node_inuse, M_indx, center_lengthmem)
```

Inputs:

`n_r` number of rings.

n_s the design parameter n_s for number of subdivisions.

`w_b` the design parameter w_B for the boundary thickness.

`Node_inuse` nodal coordinate matrix (global nodal coordinates for offset parabolic case).

`M_indx` index matrix of member connectivity.

`center_lengthmem` the length of the first member in `MemberL_prop`.

Outputs:

`error_rms` exact best-fit surface RMS error $\delta_{rms,bf}$.

`deta_bestfit` δ in the exact formulation of best-fit surface RMS error.

`F_bestfit` focal length of the whole best-fit surface.

4. MATLAB Function **bestfitcompe**

Purpose:

To compensate the deviation between best-fit surface and the surface where the vertices of mesh facets are located.

Synopsis:

```
[D_bestfit, D_mod, F_mod, D_ca_bestfit, e_off_bestfit, D_p_bestfit,
F_p_mod, D_p_mod, e_off_mod] =
bestfitcompe(deta_bestfit,F_bestfit,F_surface,D_surface,e_surface)
```

Inputs:

`deta_bestfit` δ in the exact formulation of best-fit surface RMS error.

`F_bestfit` focal length of the whole best-fit surface.

`F_surface` focal length of the whole surface where the vertices of mesh facets are located.

`D_surface` diameter of the whole surface where the vertices of mesh facets are located.

`e_surface` offset distance of the whole surface where the vertices of mesh facets are located.

Outputs:

`D_bestfit` diameter of the whole best-fit surface in spherical or center parabolic case.

`D_mod` diameter of the modified desired surface in spherical or center parabolic case.

`F_mod` focal length of the modified desired surface in spherical or center parabolic case.

`D_ca_bestfit` circular diameter of the whole best-fit surface in offset parabolic case.

`e_off_bestfit` offset distance of the whole best-fit surface in offset parabolic case.

`D_p_bestfit` parent diameter of the whole best-fit surface in offset parabolic case.

`F_p_mod` parent focal length of the modified desired surface in offset parabolic case.

`D_p_mod` parent diameter of the modified desired surface in offset parabolic case.

`e_off_mod` offset distance of the modified desired surface in offset parabolic case.

5. MATLAB Function **updatesetup**

Purpose:

To update the modified desired surface as the new desired working surface.

Synopsis:

```
updatesetup(F_update, D_update, e_update)
```

Inputs:

`F_update` modified (parent) focal length of whole surface.

`D_update` modified (parent) diameter length of whole surface.

`e_update` modified offset distance of whole surface.

6. MATLAB Function **AutoMesh**

Purpose:

The design core of the toolbox for spherical and center parabolic cases.

Synopsis:

```
[MemberL_prop, M_indx, Node_design, B_C, Indx_node_load, L_t,
L_t_nobc, F, D] = AutoMesh(n_r, n_s, n_c, rou, c_b, ratio_zeta, w_b)
```

Inputs:

`n_r` the design parameter n_r as number of rings.

n_s the design parameter n_s as number of subdivisions.

`n_c` the design parameter n_c as index of the first converging ring.

`rou` the design parameter ρ as the converging speed.

`c_b` the design parameter c_B as the boundary type.

`ratio_zeta` the design parameter ς for the layer thicknesses.

`w_b` the design parameter w_B as the boundary thickness.

Outputs:

`MemberL_prop` vector of member lengths.

`M_indx` index matrix of member connectivity.

`Node_design` designed nodal coordinates (locally for the offset parabolic case).

`B_C` the boundary conditions of designed reflector surface.

`Indx_node_load` index vector of nodes except the boundary nodes.

`L_t` total length of members l_{total} .

`L_t_nobc` total length of members within the effective region.

`F` focal length of the whole surface.

`D` diameter of the whole surface.

7. MATLAB Function **AutoMesh_offset**

Purpose:

The design core of the toolbox for offset parabolic case.

Synopsis:

```
[MemberL_prop,M_indx, Node_design,Node_design_global, B_C,
Indx_node_load, L_t, L_t_nobc, F_p, D_p, e_p] = AutoMesh_offset(n_r,
n_s, n_c, rou, c_b, ratio_zeta, w_b)
```

Inputs:

`n_r` the design parameter n_r as number of rings.

n_s the design parameter n_s as number of subdivisions.

`n_c` the design parameter n_c as index of the first converging ring.

`rou` the design parameter ρ as the converging speed.

`c_b` the design parameter c_B as the boundary type.

`ratio_zeta` the design parameter ς for the layer thicknesses.

`w_b` the design parameter w_B as the boundary thickness.

Outputs:

`MemberL_prop` vector of member lengths.

`M_indx` index matrix of member connectivity.

`Node_design` designed nodal coordinates (locally for the offset parabolic case).

`Node_design_global` designed global coordinates for offset parabolic case.

`B_C` the boundary conditions of designed reflector surface.

`Indx_node_load` index vector of nodes except the boundary nodes.

`L_t` total length of members l_{total} .

`L_t_nobc` total length of members within the effective region.

`F_p` parent focal length of the whole surface.

`D_p` parent diameter of the whole surface.

`e_p` offset distance of the whole surface.

8. MATLAB Function **setup_struc_parameter**

Purpose:

To setup the reflector structure for plotting purpose.

Synopsis:

```
setup_struc_parameter(M_indx, Node_design, B_C)
```

Inputs:

`M_indx` index matrix of member connectivity.

`Node_design` designed nodal coordinates (locally for the offset parabolic case).

`B_C` the boundary conditions of designed reflector surface.

9. MATLAB Function **plot_desired**

Purpose:

To plot the desired the surface geometry of deployable mesh reflectors in local coordinates.

Synopsis:

plot_desired(display_node_no)

Description:

This function plots the mesh geometry of desired working surface of reflectors with the following options, in which a ball-and-socket joint is expressed by a star (*),

display_node_no = 0	do not show the member and node numbers
display_node_no = 1	show the node numbers
display_node_no = 2	show the member numbers
display_node_no = 3	show the member and node numbers

Solution Example

Here a large deployable mesh reflector example is presented to show how to use this toolbox.

MATLAB codes:

```
%   Copyright (c) 2022 by Sichen Yuan
%   Created: 2022/05/30
%   $Revision: 1.0 $  $Date: 2022/05/30 $

clear
clc

% Setup the desired shape and required operating frequency
F_required = 8;
D_required = 30;
e_required = 5;
freq_op_required = 3*1e+9;
flag_effect = 1;
flag_shape = 3;
rms_budget = 50;
% Initial problem setup
error_rms = initialsetup(F_required, D_required, e_required,
freq_op_required, flag_effect, flag_shape,rms_budget);
% Preselection of design parameters
n_s = 6;
c_b = 5;% must be 0 when flag_effect = 0
tao_r = 1.1;
tao_c = 0.1;
rou = 2.5;
```

```
% Design attempt
flag_wb = 1;
[n_r, n_c, ratio_zeta, w_b, MemberL_prop,M_indx,
Node_design,Node_design_global, B_C, Indx_node_load, L_t, L_t_nobc,
F_p, D_p, e_p] = designattempt(tao_r, n_s, tao_c, rou, c_b, flag_wb,
[]);
% RMS error evaluation
[error_rms_bestfit, deta_bestfit, F_bestfit] = actual_rms_evalu(n_r,
n_s, w_b, Node_design_global, M_indx, MemberL_prop(1));
% Best-fit compensation
[temp1, temp3, temp3, D_ca_bestfit, e_off_bestfit, D_p_bestfit,
F_p_mod, D_p_mod, e_off_mod] =
bestfitcompe(deta_bestfit,F_bestfit,F_p,D_p,e_p);
updatesetup(F_p_mod, D_p_mod, e_off_mod);
% Achieve the final design
[MemberL_prop,M_indx, Node_design,Node_design_global, B_C,
Indx_node_load, L_t, L_t_nobc, F_p_mod, D_p_mod, e_p_mod] =
AutoMesh_offset(n_r, n_s, n_c, rou, c_b, ratio_zeta, w_b);
% Measure the actual surface RMS error
[error_rms_mod, deta_bestfit_mod, F_bestfit_mod] =
actual_rms_evalu(n_r, n_s, w_b, Node_design_global,
M_indx,MemberL_prop(1));
% Evaluate the shape of best-fit surface
[temp1, temp3, temp3, D_ca_bestfit_mod, e_off_bestfit_mod,
D_p_bestfit_mod] =
bestfitcompe(deta_bestfit_mod,F_bestfit_mod,F_p_mod,D_p_mod,e_p_mod);

% Setup the reflector structure for plotting
setup_struc_parameter(M_indx,Node_design, B_C);
% Plot the desired profile
figure
plot_desired(0)
axis equal
view([0 0 1])
% The end of the toolbox demo
disp 'The toolbox works properly.'
```

MATLAB results:

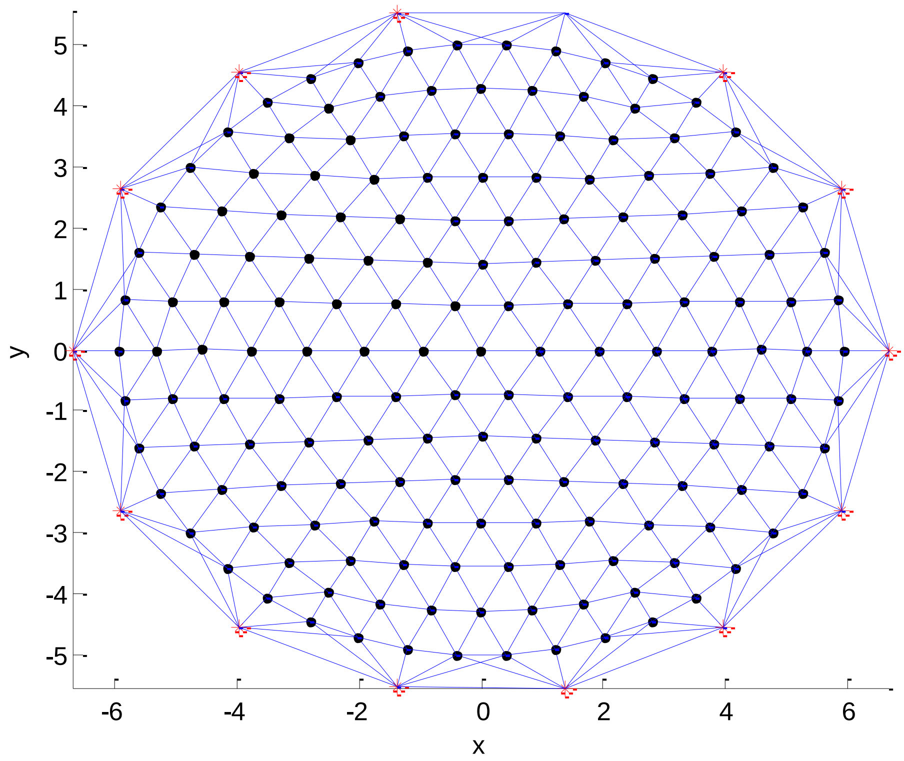

Figure 2 Generated mesh geometry of the deployable mesh reflector

References

[1] Shi, H., Yuan, S. and Yang, B., 2018. New methodology of surface mesh geometry design for deployable mesh reflectors. *Journal of Spacecraft and Rockets*, 55(2), pp.266-281.

[2] Yuan, S. and Yang, B., 2019. The fixed nodal position method for form finding of high-precision lightweight truss structures. *International journal of Solids and Structures*, 161, pp.82-95.

[3] Yuan, S. and Zhu, W., 2021. Optimal self-stress determination of tensegrity structures. *Engineering Structures*, 238, p.112003.

[4] Yuan, S., Yang, B. and Fang, H., 2020. Direct root-mean-square error for surface accuracy evaluation of large deployable mesh reflectors. *In AIAA SciTech 2020 Forum* (p. 0935).

[5] Yuan, S., 2022. Review of Root-Mean-Square Error Calculation Methods for Large Deployable Mesh Reflectors. *International Journal of Aerospace Engineering*.