Transfinite Interpolation

Siva Srinivas Kolukula
Structural Mechanics Laboratory
IGCAR, Kalpakkam
INDIA
allwayzitzme@gmail.com

1. Introduction

Grid generation is very important in solving partial differential equations in computational fluid dynamics, aerodynamics, tidal and estuary flow, plasma physics, electromagnetic and structures. The generation of grid is the first and foremost step in finite element method, computational fluid dynamics, finite volume method, finite difference method etc. The accuracy of the solution of partial differential equation depends on how fine and sensible is grid for the problem domain. Discretizing the domain is very challenging if the domain is complex. Generating a grid would be easy if analytic expressions are known for the geometries of the domain. Fundamentals of grid generation start with generating a grid when analytic expressions for the domain are known. Readers can refer [1] for this method of grid generation. One of the methods among many other methods of grid generation is by interpolation.

2. Transfinite Interpolation

Grid generation based on interpolation has two basic advantages:

1. Rapid computation of grids
2. Direct control over grid point locations

These advantages are offset by the fact that interpolation methods may not generate smooth grids, in particular, when the problem domain has steep curves or bends. In these cases grid gets folded across the bends of the domain. Grid generation methods by interpolation also called algebraic methods. The standard method of algebraic grid generation is known as transfinite interpolation (TFI).

Any 2D grid generation problem demands the description of four boundaries of the region, i.e. four parametric equations for the boundaries.

$$\begin{aligned} &\mathbf{X}_b(\xi), \quad \mathbf{X}_t(\xi), \quad 0 \le \xi \le 1, \\ &\mathbf{X}_l(\eta), \quad \mathbf{X}_r(\eta), \quad 0 \le \eta \le 1 \end{aligned} \tag{1}$$

The subscripts on **X** stand for bottom, top, left and right boundaries of the logical domain as shown in Figure 1. Where $\mathbf{X}_i = (x, y)_i$, it represents it have two components in it.

The vector notation shown in Eq. (1) should be converted to component form to implement in a computer program. There should be four important consistency checks for the boundary formulas at four corners of the region, as shown below.

(i) $\mathbf{X}_b(0) = \mathbf{X}_l(0)$

(ii) $\mathbf{X}_b(1) = \mathbf{X}_r(0)$

(iii)$\mathbf{X}_r(1) = \mathbf{X}_t(1)$
(iv) $\mathbf{X}_l(1) = \mathbf{X}_t(0)$

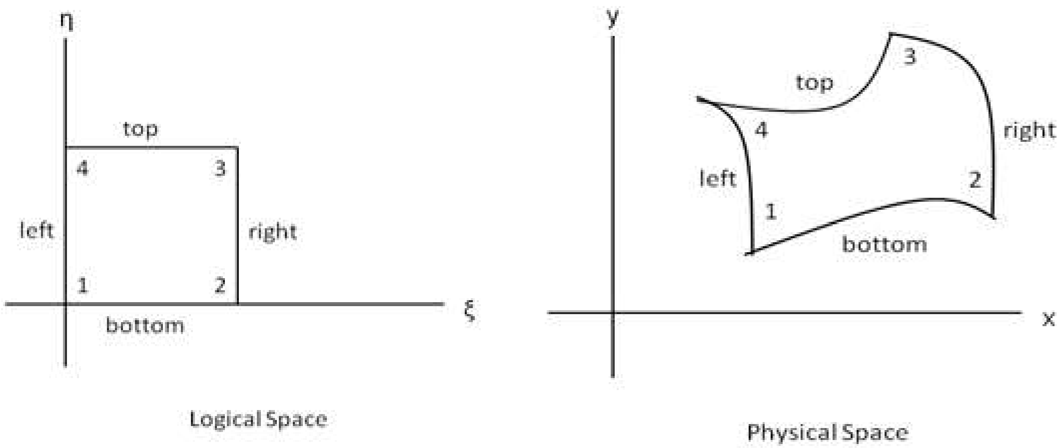

Figure 1. Boundaries of Planar region

The first degree Lagrange polynomials 1-ξ, ξ, 1-η and η are used as blending functions in the basic transfinite interpolation formula. The TFI formula is

$$\mathbf{X}(\xi,\eta) = (1-\eta)\mathbf{X}_b(\xi) + \eta\mathbf{X}_t(\xi)+(1-\xi)*\mathbf{X}_l(\eta)+\xi\mathbf{X}_r(\eta) \\ -\{\xi\eta\mathbf{X}_t(1)+\xi(1-\eta)\mathbf{X}_b(1)+\eta(1-\xi)\mathbf{X}_t(0)+(1-\xi)(1-\eta)\mathbf{X}_b(0)\} \qquad (2)$$

4. Examples of TFI

The following three examples of TFI show power of the method and its limitations. The three examples are Modified Horseshoe, Swan and Chevron.

4.1 Modified Horseshoe

A TFI grid for the Horseshoe is shown in Figure 2; the TFI grid is smooth and unfolded. The following are the boundary parameterizations

Bottom boundary:

$$x_b(s) = \rho b_0 cos\left(\frac{\pi}{2}(1-2s)\right), \quad y_b(s) = b_0 sin\left(\frac{\pi}{2}(1-2s)\right) \qquad (3)$$

Top boundary:

$$x_t(s) = \rho b_1 cos\left(\frac{\pi}{2}(1-2s)\right), \quad y_t(s) = b_1 sin\left(\frac{\pi}{2}(1-2s)\right) \qquad (4)$$

Left boundary:

$$x_l(s) = 0, \ \ y_l(s) = b_0 + (b_1 - b_0)s \qquad (5)$$

Right boundary:

$$x_r(s) = 0,\ y_r(s) = -(b_0 + (b_1 - b_0)s) \tag{6}$$

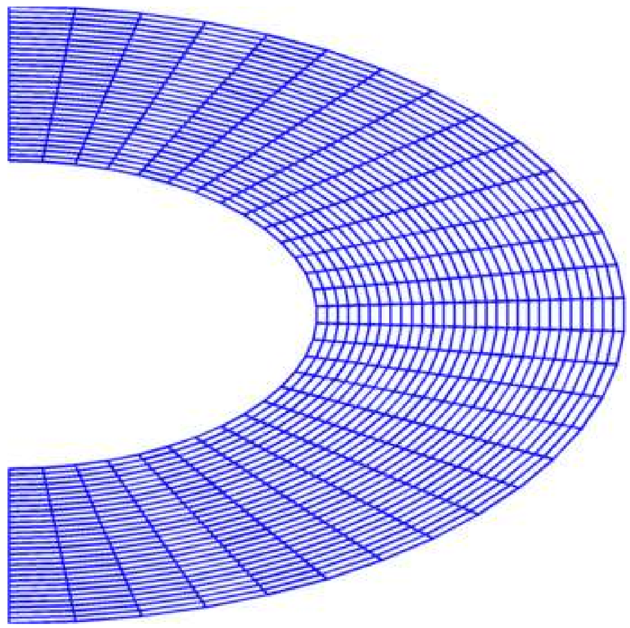

Figure 2: TFI grid on the horseshoe

4.2 Swan:

The TFI grid for the swan is shown in Figure 3. From figure 3 it is clear that the grid is folded. The boundary parametrizations are given by

Bottom boundary:

$$x_b(s) = s,\ y_b(s) = 0 \tag{7}$$

Top boundary:

$$x_t(s) = s,\ y_t(s) = 1 - 3s + 3s^2 \tag{8}$$

Left boundary:

$$x_l(s) = 0,\ y_l(s) = s \tag{9}$$

Right boundary:

$$x_r(s) = 1 + 2s - 2s^2,\ y_r(s) = s \tag{10}$$

4.3 Chevron

The TFI grid for the chevron is shown in Figure 4; the slope discontinuity on the boundary of the Chevron is propogated into the interior of the physical region by TFI. The boundary parametrizations of the Chevron are as follows

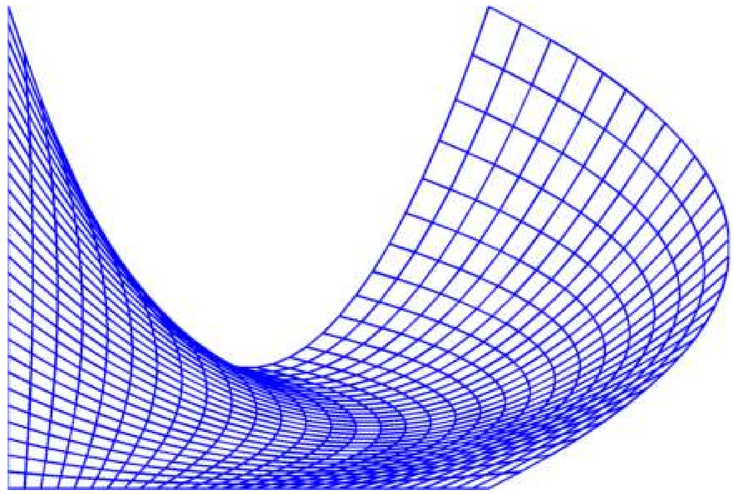

Figure 3: TFI grid on the swan

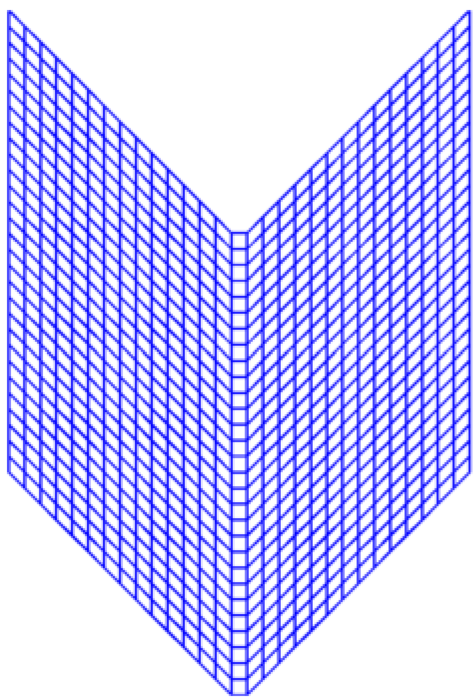

Figure 1: TFI grid on the Chevron

Bottom boundary:

$$x_b(s) = s, \quad y_b(s) = \begin{cases} -s, & s \leq \frac{1}{2} \\ s-1, & s > \frac{1}{2} \end{cases} \tag{11}$$

Top boundary:

$$x_t(s) = s, \quad y_t(s) = \begin{cases} 1 - s, & s \leq \frac{1}{2} \\ s, & s > \frac{1}{2} \end{cases} \tag{12}$$

Left boundary:

$$x_l(s) = 0, \; y_l(s) = s \tag{13}$$

Right boundary:

$$x_r(s) = 1, \; y_r(s) = s \tag{14}$$

Transfinite interpolation produces excellent grids on many regions but is inadequate on the Swan, Chevron, Airfoil, Backstep, Plow and C. The main disadvantages of TFI are lack of smoothness and a tendency to fold grids on complex doamins.

For any discussions, advice, bugs, developing the code please feel free to mail me. Please share your experience with the code by commenting or rating.

References:

[1] Grid generation (To demonstrate grid generation using analytic coordinate systems)

Link: (http://www.mathworks.in/matlabcentral/fileexchange/40618-grid-generation)

[2] Fundamentals of Grid Generation – Patrick Knupp, Stanley Steinberg