Appendix

APPLICATION OF GEOMETRIC TRANSFORM FOR OBTAINING SHAPES OF CURVES OF DENT WIRES FOR WEAVING 3D SHAPES

Geometric transforms can be used as a tool to obtain curve shapes of reed with shaped dent wires. Also skeleton grid view of the shape can be obtained.

A.1 Mathematical evaluation procedure:

Consider 2D fabric as a rectangular grid in 2-D plane. Apply geometric transforms to convert this 2D grid into a 3D shape. On doing so, lengths of sides of each cell change depending upon profile of 3D shape. These changes can be calculated mathematically.

For this the increments in lengths of sides of cells as compared to those of original lengths (those in 2D grid) is to be calculated. Reed dent wires are to be shaped that increased end spacing (or increased pick lengths) is obtained depending upon shape profile as explained in section BAKI.

Taking an example of a hemispherical shape and applying this fundamental, the incremental lengths of cells can be calculated and curve shapes for reed dent wires can be obtained with following methods:

i. By $s = r\theta$ concept:

   Knowing radius r and angle θ made by arc with horizontal line, arc length can be calculated. In this way increased arc lengths can be found.

ii. By line integration:

If end points of the curved line are known then arc length between those two points can be calculated.

In the developed program, first method in employed. However, second method is more generalized and can be used for shapes other than hemispherical shapes also.

Program is developed in MATLAB software.

A.2 MATLAB codes for generating curves for reed dent wire shapes as well as 3D grid view for hemispherical and pyramidal shapes:

A.2.1 Hemisphere program:

```
clc
clear
a=input('Enter lower limit=');
b=input('Enter upper limit=');
r=input('Enter radius=');
n=input('Enter the no. of threads=');
x=linspace(a,b,n);
y=linspace(a,b,n);
c=(a+b)/2;
%command for 2d mesh
[X Y]=meshgrid(x,y);
A=((X-c).^2+(Y-c).^2);
for i=1:n
```

```
  for j=1:n
    if A(i,j)<=r^2
      z(i,j)=sqrt((r^2)-A(i,j));
    else
      z(i,j)=0;
    end
  end
end
% 3d mesh generation
mesh(X,Y,z);
%surf(X,Y,z)
xlabel('x');
ylabel('y');
zlabel('z');

% arc length calcu for each cell
for i=1:n
  for j=1:fix(n/2)
    dx(i,j)=abs(X(i,j+1)-X(i,j));
      % Total cell length with addition of each cell
      if X(i,j+1)~=0|z(i,j+1)~=0
        arcdx(i,j)=max(z(i,:))*atan(abs(z(i,j+1)/X(i,j+1)));
      else
        arcdx(i,j)=dx(i,j);
      end
      % for individual cell length
      if j==1
        arcdx1(i,j)=arcdx(i,j);
      else
```

```
        arcdx1(i,j)=arcdx(i,j)-arcdx(i,j-1);
      end
      % to avoid zero arc length
      if arcdx1(i,j)==0
        arcdx1(i,j)=dx(i,j);
      end
      % difference between elongated cell and original flat : arc length
      I(i,j)=arcdx1(i,j)-dx(i,j);
    end
  end
end
% program for reed design
for i=1:n
  % to take 50% data from original calculated arc dx
  for j=1:fix(n/2)
    if arcdx(i,j)==0
      arcdx(i,j)=dx(i,j);
    else
      continue
    end
  end
end
u = arcdx;
%for knowing size of the matrix
[m n] = size(arcdx);
for i = 1:m
  for j = 1:n
    arcdx(i,j)=u(i,j) + (i)*min(min(dx));
  end
```

```
end
% to plot figure
figure
for i = 1:n
  plot(arcdx(i,:))
  hold on
end

% g1=[];
% g2=[];
% g3=[];
% g4=[];
% g5=[];
% g6=[];
% g7=[];
% g8=[];
% g9=[];
% g10=[];
% R=[0 .003 .008 .020 .060 .100 .250 .450 .600 .800]
% for i=1:n
%    for j=1:fix(n/2)
%       if I(i,j)==R(1)
%          g1=[ R(1); g1];
%       elseif R(1)<I(i,j)& I(i,j)<=R(2)
%             g2=[I(i,j); g2];
%       elseif R(2)<I(i,j)& I(i,j)<=R(3)
%             g3=[I(i,j) ;g3];
%       elseif R(3)<I(i,j)& I(i,j)<=R(4)
%             g4=[I(i,j) ;g4];
```

```
%       elseif R(4)<I(i,j)& I(i,j)<=R(5)
%             g5=[I(i,j) ;g5];
%       elseif R(5)<I(i,j)& I(i,j)<=R(6)
%             g6=[I(i,j) ;g6];
%       elseif R(6)<I(i,j)& I(i,j)<=R(7)
%             g7=[I(i,j) ;g7];
%       elseif R(7)<I(i,j)& I(i,j)<=R(8)
%             g8=[I(i,j) ;g8];
%       elseif R(8)<I(i,j)& I(i,j)<=R(9)
%             g9=[I(i,j) ;g9];
%       else
%             g10=[I(i,j) ;g10];
%       end
%    end
% end
```

A.2.2 Pyramid program:

```
clc
clear
a=input('Enter lower limit :');
b=input('Enter upper limit :');
n=input('Enter no. of threads=');
x=linspace(a,b,n);
y=linspace(a,b,n);
[X Y]=meshgrid(x,y);
for i=1:n
   for j=1:n
```

```
    if X(i,j)<=b  & X(i,j)>=0 & Y(i,j)<=b  & Y(i,j)>=0 & X(i,j)<=Y(i,j)
      z(i,j)=1-Y(i,j);
    elseif X(i,j)>=a   & X(i,j)<=0  & Y(i,j)<=b   & Y(i,j)>=0  & -X(i,j)<=Y(i,j)
      z(i,j)=1-Y(i,j);
    elseif X(i,j)<=b   & X(i,j)>=0  & Y(i,j)<=b   & Y(i,j)>=0  & X(i,j)>=Y(i,j)
      z(i,j)=1-X(i,j);
    elseif X(i,j)>=0  & X(i,j)<=b & Y(i,j)<=0  & Y(i,j)>=a & X(i,j)>=-Y(i,j)
      z(i,j)=1-X(i,j);
    elseif X(i,j)<=b  & X(i,j)>=0 & Y(i,j)<=0  & Y(i,j)>=a & X(i,j)<=-Y(i,j)
      z(i,j)=1+Y(i,j);
    elseif X(i,j)>=a   & X(i,j)<=0  & Y(i,j)<=0   & Y(i,j)>=a  & X(i,j)>=Y(i,j)
      z(i,j)=1+Y(i,j);
    elseif X(i,j)>=a   & X(i,j)<=0  & Y(i,j)<=0   & Y(i,j)>=a  & X(i,j)<=Y(i,j)
      z(i,j)=1+X(i,j);
    elseif X(i,j)>=a   & X(i,j)<=0  & Y(i,j)<=b   & Y(i,j)>=0  & -X(i,j)>=Y(i,j)
      z(i,j)=1+X(i,j);
    else
      z(i,j)=0;
    end
  end
end
mesh(z)
```

```
for i=1:fix(n/2)
   for j=1:fix(n/2)
      Z(i,j)=z(i,j);
      dx(i,j)=abs(X(i,j+1)-X(i,j));
   end
end
figure
for i=1:fix(n/2)
   P(i,:)=Z(i,:)+(i)*min(min(dx));
   plot(P(i,:))
   hold on
end
```

A.3 Role of differential geometry:

It is clear from above codes that calculating increments for a regular shape is easier than that for an irregular shape. For hemisphere, either $s = r\theta$ concept or line integration can be used to find incremental length. For pyramid incremental length can be calculated directly.

For irregular shapes only integral formula can be used. Differential geometry explains how this integral formula of calculating length is related to h\the space curves of 3D shape to be produced.

In the theory of plane curves, a curve is specified by means of a single equation or by parametric representations.

In 3D Euclidean space, a single equation would represent a surface while two equations are required to specify a curve. Thus a curve appears as an intersection of two surfaces.

In Cartesian co-ordinate system, a curve is specified by the equation $x = X(u)$; $y = Y(u)$; $z = Z(u)$ where X,Y,Z are real valued functions while u is a real parameter.

In vector notations, a curve is specified by a vector valued function, $r = R(u)$.

A.3.1 Functions of class m:

A vector valued function R=(X,Y,Z) defined on an interval I is said to be of class m if it has m^{th} derivative at every point and if this derivative is continuous on I or each of its components X,Y,Z is class of m.

Such function is specified in vector notations as R=(X,Y,Z) or in Cartesian coordinate system as $x = X(u)$; $y = Y(u)$; $z = Z(u)$

If the derivative $\frac{dR}{du} = r$ never vanishes on I, i.e. x, y, z never vanish simultaneously then the function is said to be *Regular*.

A regular vector valued function of class m is called *Path of class* .

A.3.2 Equivalent Paths:

Two paths R_1 and R_2 of the same class m in I_1 and I_2 are called equivalent if there exists a strictly increasing function ϕ of class m which maps I_1 onto I_2 and is such that $R_1=R_2\ {}_0\phi$. This is an equivalence relation. (Reflexive and transitive relation are obvious; symmetry follows from the fact that ϕ is never zero and hence admits an inverse function of same class m).

Curve:

An equivalent class of paths of class m determines a *curve of class m.*

A.3.3 Arc length:

Distance between two points $r_{1211}(x_1, y_1, z_1)$ and r_2 (x_2, y_2, z_2) in Euclidean space is defined as a number,

$$|r_1 - r_2| = \sqrt{(r_1 - r_2)^2} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

Arc:

If for a given path r = R(u) and two numbers a, b such that we have $a \le u \le b$ then the path r = R(u) is an arc of the original path joining the points corresponding to the points a and b.

Consider the subdivisions Δ of the interval (a, b) by the points, $a = u_0 \prec u_1 \prec \ldots\ldots \prec u_n = b$, then corresponding to every subdivisions the length given by,

$$L_\Delta = \sum_{i=1}^{n} |R(u_i) - R(u_{i-1})|$$

is the length of the polygon inscribed to the arc by joining the successive points. Clearly, addition of more points of subdivisions increases the length of the polygon. Length of the arc can be considered to be upper bound of L_Δ taken over all possible subdivisions of (a, b).

This upper bound is finite because for any Δ,

$$L_\Delta = \sum_{i=1}^{n} \left| \int_{u_{i-1}}^{i} \dot{R}(u)\,du \right|$$

$$\leq \sum_{i=1}^{n} \left| \int_{u_{i-1}}^{i} \dot{R}(u)\,du \right|$$

$$= \int_a^b \left| \dot{R}(u)\,du \right| \ \ldots\ldots\ldots\ldots\ldots(1)$$

Also RHS is finite and independent of Δ .

The upper bound is actually equal to the RHS of equation (1) so that it will give the formula for arc length.

Let s = S(u) be arc length from point a to any point u. Therefore, the arc length from u_o to u is given as,

$$S(u) - S(u_0)$$

From equation 1,

$$S(u) - S(u_0) \leq \int_a^o \left| \dot{R}(u)\, du \right|$$

From the definition of the arc length,

$$|R(u) - R(u_0)| \leq S(u) - S(u_0)$$

Hence,

$$\left| \frac{R(u) - R(u_0)}{u - u_0} \right| \leq \left| \frac{S(u) - S(u_0)}{u - u_0} \right| \leq \frac{1}{u - u_0} \int_{u_0}^{u} \left| \dot{R}(u)\, du \right| \quad \ldots\ldots\ldots\ldots (8.2)$$

The equation (8.2) holds true even if $u \prec u_0$. Now as u tends to u_o both extreme sides of equation (8.2) tend to the same limit i.e. $\left|\dot{R}(u_0)\right|$

Thus, the middle term $\dot{S}(u_0)$ also tends to this limit. This means that $\dot{S}(u_0)$ exists and;

$$\dot{S}(u_0) = \left|\dot{R}(u_0)\right|$$

Since this is true for any u_0 in the range I of the parameter, it follows that S is a function of same class of the curve. Thus,

$$S = S(u) = \int_{a}^{u} \left| \dot{R}(u) \right| du$$

For Cartesian co-ordinate system,

$$S = \int_{a}^{u} \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}\, du \quad \text{..................(8.3)}$$

Thus using the above formula of equation (8.3) the required increments of the space curves of which the 3D shape is comprised can be found. Thus using the above formula shapes of reed wires for regular or irregular shapes can be obtained.