

```

function npoints=EqSamp (points,numSamples)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   FUNCTION : EqSamp
%   parameters :
%       points - sequence of 2d points given as x,y,x,y...
%       numSmamples - number of points to sample.
%       npol - sequence of sampled points.
%   This function calculates the length of the contour
%   and samples the contour # "points" points at equal
%   distance starting with the first point.
%   Calls internal Function NewPoint
%
%   Comment: uses orig code that assumes input is array so
%   map sequence to array, compute and map resulting array
%   to output sequence.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% map point list to point array
arr = reshape(points,2,size(points,2)./2)';

n=size(arr,1); % number of input points
narr=zeros(numSamples,2); % create new points array

% Calculate length of boundary.
sum=0;
for i=1:n-1
    sum=sum+sqrt( (arr(i,1)-arr(i+1,1))^2+(arr(i,2)-arr(i+1,2))^2);
end;

% Boundary is assumed to be closed so add length from last point to first.
sum=sum+sqrt( (arr(1,1)-arr(n,1))^2+(arr(1,2)-arr(n,2))^2);

% Calc the length between sampled points
unit=sum/numSamples;

% Calc the sampled points
narr(1,1:2)=arr(1,1:2); % first sampled point is first input point
pos=1;
npos=2;
for i=2:numSamples
    flag=0;
    rdis=unit;
    dis=sqrt( (narr(i-1,1)-arr(npos,1))^2+(narr(i-1,2)-arr(npos,2))^2);
    while rdis>=dis
        rdis=rdis-dis;
        pos=pos+1;
        npos=npos+1;
        if npos>n
            npos=1;
        end;
        dis=sqrt( (arr(pos,1)-arr(npos,1))^2+(arr(pos,2)-arr(npos,2))^2);
        flag=1;
    end;
    if flag==1 % found segment in which npoint should be sampled at distance
rdis

```

```

        narr(i,1:2)=NewPoint(arr(pos,1),arr(pos,2),arr(npos,1),arr(npos,2),rdis
);
    else
        narr(i,1:2)=NewPoint(narr(i-1,1),narr(i-
1,2),arr(npos,1),arr(npos,2),rdis);
    end;
end;

npoints = reshape(narr',1,numSamples*2);

```

```

function npoint=NewPoint(x1,y1,x2,y2,dis)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
%   FUNCTION: NewPoint
%   This function finds a new point for the equal sampling.
%   The new point is dis distance along the segment from
%   x1,y1 to x2,y2
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

npoint=[0 0];
alfa=atan2((y1-y2),(x1-x2));
npoint(1)=x1-dis*cos(alfa);
npoint(2)=y1-dis*sin(alfa);

```