3D-curves play an important role in the engineering, design and manufacture in Shipbuilding.

Prior of the development of *mathematical and computer models* to support engineering, design and manufacturing descriptive geometry was used.

Today, many of these geometric design techniques have been carried over into *Computer Aid Geometric Design.*
Surfaces are frequently described by a *net of curves* lying in orthogonal cutting planes plus three-dimensional (3D) feature or detail lines.

This curve are obtained by *digitizing on a physical model or a drawing* and then fitting a mathematical curve to the digitized data.
Interpolation curve are characterized by the fact that the derived mathematical curve passes through each and every data point. Ex: Cubic spline, Parabolic blended curve, B-spline, etc.

Fitting curve are generated without any prior knowledge of curve shape or form. These are characterized by the fact that few if any point on the curves pass through the control points used to define the curve.
Ship Calculation (1)

Curves: \[ \text{Plane curve} \quad - \quad \text{Space curve} \]

Intersecting two planes \[ \rightarrow \text{Straight line} \]
Intersecting a plane with a curves surface \[ \rightarrow \text{Plane curve} \]
Intersecting two curved surfaces \[ \rightarrow \text{Space curve} \]

Form of Representation

<table>
<thead>
<tr>
<th>Non parametric</th>
<th>Parametric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Implicit</td>
<td>Plane curve</td>
</tr>
<tr>
<td>( f(x,y)=0 )</td>
<td>( x=f(t) )</td>
</tr>
<tr>
<td>( f(x,y,z)=0 )</td>
<td>( y=f(t) )</td>
</tr>
<tr>
<td>( g(x,y,z)=0 )</td>
<td>( y=g(x) )</td>
</tr>
<tr>
<td></td>
<td>( x=f(t) )</td>
</tr>
<tr>
<td></td>
<td>( z=h(t) )</td>
</tr>
<tr>
<td>Explicit</td>
<td>Space curve</td>
</tr>
<tr>
<td>( y=f(x) )</td>
<td></td>
</tr>
<tr>
<td>( y=f(x) )</td>
<td></td>
</tr>
<tr>
<td>( z=g(x) )</td>
<td></td>
</tr>
</tbody>
</table>
Non-parametric implicit:

\[ F(x,y) = 0 \]

→ Problem in finding the correct root of an algebraic equation.

Non-parametric explicit:

\[ y = f(x) \]

→ Axis dependent no multiple valued curves *problems with vertical tangent.*

Parametric:

→ Each point on the curve is defined by parameter \( t \)

→ Multiple-valued curves

→ Axis independent

→ *Problem in case find value of \( y(x) \) with a given \( x \)
When the composite curve are combined. The curves are determined by the properties according to the continuity condition at the given \textit{joint point} position.

To the \( n \)th derivative are \textit{continuous} at that point.

The following table shows the major continuity conditions.

\begin{itemize}
  \item \( C^\text{none} \) positional discontinuity , discontinuous
    (non continuity)
  \item \( C^0 \) positional continuity , continous \( y \), slope discontinuity
    (zero order continuity)
  \item \( C' \) slope continuity , continous \( y' \), curve discontinuity
    (first order continuity)
  \item \( C^2 \) curvature, continous \( y' \), curve contintuity \( y'' \)
    (second order continuity)
  \item \( C^3 \) curvature, continous \( y' \), curve contintuity \( y''' \)
    (third order continuity)
  \item \( C^n \) all derivative are continous.
\end{itemize}
Ship Calculation (1)

No Continuity
C0 Continuity (positional)
C1 Continuity (tangential)
C2 Continuity (curvature)
Lagrange Polynomial Curve

- Given \((x_o, y_o), (x_1, y_1), \ldots, (x_n, y_n)\) \(x_i < x_j\) and \(i < j\)
- \(n - \) order interpolational polynomial equation

\[ y = f(x) = \sum_{i=0}^{n} y_i \cdot L_i(x) \]

\[ L(x) = \frac{(x - x_1)(x - x_2)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)}{(x_i - x_1)(x_i - x_2)\ldots(x - x_{i-1})(x - x_{i+1})\ldots(x - x_n)} \]

\[ = \prod_{j=0}^{n} i \neq j \frac{x - x_j}{x_i - x_j} \]

Ex: \(L_0(x) = \frac{x - x_1}{(x_0 - x_1)\ldots(x - x_n)}\)
Property

→ If the number of points are increased to enhance the precision of curve. The degree of polynomial is increased but make increase a possibly of oscillation. *The curve is too perfect.*

→ Below these oscillations in an interpolating curve of $n = 7$

→ The variation of curve affects a whole curve
This particular value of the parameter $t$ is associated with each interpolating point $P_i$ and is called knot

As normal: we will give the values of $t$ such as $t_i = t_{i-1} + 1$

or $t_{i-1} - t_i = “1”$

“1” named: grid uniform which can be “2” or “3” ...const... The Curve will be in same.

Another way: If we give various grid space – We will take different Curve such below:
Basic idea:
For 4 given points $P_0, P_1, P_2, P_3$,
Find the two overlapping parabolas $P(n)$ through $P_0, P_1, P_2$ and $G(s)$ through $P_1, P_2, P_3$. 
Then find a smooth \( C(t) \) between the two interior points \( P_1 \) and \( P_2 \) by blending the two overlapping parabolic segments.

This method was first suggested by Over Houser (1968).

Parabolic blending is a measure to draw a smooth curve between 2 interior points from 4 given points.
Definition

Parabolic blended curve is given by

\[ \vec{C}(t) = (1-t) \vec{p}(n) + t \vec{g}(s) \]  

(linear form)

Where \( \vec{p}(n), \vec{g}(s) \) are parametric parabolas though \( \vec{P}_0, \vec{P}_1, \vec{P}_2 \) and \( \vec{P}_1, \vec{P}_2, \vec{P}_3 \), respectively.

The parametric representation of \( \vec{p}(n), \vec{g}(s) \) is

\[ \vec{p}(n) = [n^2 n 1][B] \]

\[ \vec{g}(s) = [s^2 s 1][D] \]
Where $[B]$ & $[D]$ are metrics involving the position vectors $\vec{P}_0, \vec{P}_1, \vec{P}_2$ and $\vec{P}_1, \vec{P}_2, \vec{P}_3$.

To determine $[B]$ & $[D]$ that it necessary to establish the relationship between $r$, $t$ and $s$.

Assumed: 

$\text{Assumed: } \quad n = k_1 t + k_2 \quad s = k_3 t + k_4$

Voting that data is frequently evenly or nearly evenly spaced and assuming that the parameter range are normalized:

$$0 \leq n \cdot s \cdot t \leq 1$$
To calculate $K_i$, we will write the condition at the end of two parabolas and the curve segment.

**Boundary condition:**

\[
\begin{align*}
P(0) &= P_0 & P(0.5) &= P_1 & P(1) &= P_2 \\
Q(0) &= P_1 & Q(0.5) &= P_2 & Q(1) &= P_3 \\
C(0) &= P_1 & C(1) &= C_2
\end{align*}
\]

This result in:

\[
n(t) = \frac{1}{2}(t+1) ; \ s(t) = \frac{1}{2}t
\]

Then:

\[
\overrightarrow{C(t)} = [t^3 \ t^2 \ t \ 1]^t \cdot \frac{1}{2} \begin{bmatrix}
-1 & 3 & -3 & 1 \\
2 & -5 & 4 & -1 \\
-1 & 0 & 1 & 0 \\
0 & 2 & 0 & 0
\end{bmatrix} \begin{bmatrix}
P_0 \\
P_1 \\
P_2 \\
P_3
\end{bmatrix}
\]

\[
\tilde{C}(t) = [t^3 \ t^2 \ t \ 1][A][G]
\]
Normal parabolic blending is assumed a specific value of \( \frac{1}{2} \) for parameter \( r \) and \( s \) at \( P_1 \) and \( P_2 \).

If the position vector (data) to be fit are not nearly evenly spaced, more reasonable assumption is to use a normalized chord length approximation.

\[
\alpha = \frac{|P_2 - P_1|}{|P_3 - P_2| + |P_2 - P_1|} \quad 0 < \alpha < 1.
\]

\[
\beta = \frac{|P_3 - P_2|}{|P_3 - P_2| + |P_4 - P_2|} \quad 0 < \beta < 1.
\]
Generalized Parabolic Blending

- Using the Boundary condition as concerned we will take the relationship between \( n, s \) with \( t \) parameter.

\[
\begin{align*}
n_{(t)} &= (1 - \alpha)t + \alpha \\
s_{(t)} &= \beta t
\end{align*}
\]

\[
\overrightarrow{C(t)} = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{-(1-\alpha)^2}{\alpha} & \frac{(1-\alpha)+\alpha\beta}{\alpha} & \frac{-(1-\alpha)-\alpha\beta}{\alpha} & \frac{\beta^2}{1-\beta} \\
\frac{2(1-\alpha)^2}{\alpha} & \frac{2(1-\alpha)-\alpha\beta}{\alpha} & \frac{1}{1-\beta} & \frac{-\beta^2}{1-\beta} \\
\frac{-(1-\alpha)^2}{\alpha} & \frac{(1-2\alpha)}{\alpha} & \frac{1}{1-\beta} & 0 \\
\frac{\alpha}{0} & \frac{\alpha}{0} & \frac{\alpha}{0} & 0
\end{bmatrix}
\begin{bmatrix}
P_0 \\
P_1 \\
P_2 \\
P_3
\end{bmatrix}
\]
Generalized Parabolic Blending

**ASSUMPTION:**
The Blending Curve (Cat-Mull Curve) starts at the point $P_1$ and ends at $P_{i-1}$. To make it pass through over all initial given points, we must add more two points called: Imagination Points

Example:
Mathematical spline: piecewise polynomical with continuous derivatives at knots.
Knots (given points or offsets)

From: Plastic strip → drafting spline. (Deforme to minimize energy)
Use: Duck → lead weight.

“Theory of spline” is a certain generalization of the behavior of the elastic spline.
Linearized beam theory:

\[ \frac{M(x)}{EI} = y'' \quad (Euler\ Equation) \]

E: young’s modulus (*determined by material properties of the beam*)
I: Moment of Inertia (*determined by the cross-sectional shape of the beam*)

Assuming that ducks acts as simple supports, the bending moment \( M(x) \) is know to way linearly between supports.
Substituting: \( M(x) = Az + B \). Euler’s equation becomes:

\[ y = Ax^3 + Bx^2 + Cx = D. \] for the deflection of the beam.

The results show that the shape of the physical spline *between ducks* is mathematically discribled by cubic polynomial.
A piecewise 3\textsuperscript{rd} degree polynomial with the following characteristics:
- Between Ducks are knots.
- We need Ducks to describe the curve.

\begin{align*}
\text{Between Duck} & & \text{Between Knots} \\
y''' & \quad \text{constant} & & \text{discontinuous} \\
y'' & \quad \text{linear} & & \text{continuous} \\
y' & \quad \text{guaradtic} & & \text{continuous} \\
y & \quad \text{cubic} & & \text{continuous} \\
\rightarrow \quad C^2 & \text{ (second order) continuing at knots.} \\
\text{General mathematical spline:} & & \text{a piecewise polynomical of degree } m \text{ with contiuity of derivatives of order } m-1 \text{ at knots}
\end{align*}
The equation for a single parametric cubic spline segment is given by:

\[ \overrightarrow{P(t)} = \sum_{i=1}^{4} B_i t^{i-1} \]

Where \( t_i \), \( t_{i+1} \) are the parameter values at the beginning and the end of segment.
is the position vector of any point on the cubic segment

\[ P(t) = [x(t) \ y(t) \ z(t)] \]: Vector values function or cartesian coordinate of position vector.

The constant coefficient \( B_i \) are determined by specified Four Boundary Conditions for Spline Segment.

\[
\overrightarrow{P'(t)} = \sum_{i=1}^{4} (i - 1)B_i t^{i-2}
\]

Or

\[
\overrightarrow{P'(t)} = [x'(t) \ y'(t) \ z'(t)]
\]
Assuming without loss of generality, that \( t_i = 0 \) and applying the **Four Boundary Conditions**.

**Position Condition**

\[
\overrightarrow{P(0)} = \overrightarrow{P_1} \quad \overrightarrow{P'(0)} = \overrightarrow{P'_1}
\]

**Tangent Condition**

\[
\overrightarrow{P(t_2)} = \overrightarrow{P_2} \quad \overrightarrow{P'(t_2)} = \overrightarrow{P'_2}
\]

The resultant

\[
\begin{align*}
\overrightarrow{B_1} &= \overrightarrow{P_1} \\
\overrightarrow{B_2} &= \overrightarrow{P'_1} \\
\overrightarrow{B_3} &= \frac{3(\overrightarrow{P_2} - \overrightarrow{P_1})}{t_2^2} - \frac{2\overrightarrow{P_1}}{t_2} - \frac{\overrightarrow{P_2}}{t_2} \\
\overrightarrow{B_4} &= \frac{2(\overrightarrow{P_1} - \overrightarrow{P_2})}{t_2^3} + \frac{\overrightarrow{P_1}}{t_2^2} + \frac{\overrightarrow{P_2}}{t_2^2}
\end{align*}
\]
Cubic Spline with Internal points

Generalizing for \( n \) data points to give \( n-1 \) piecewise cubic spline segments with position, slope and curvature i.e \( C^2 \)-continuity at the internal joints.

The generalized equations for any two adjacent cubic spline segments:

\[
\overrightarrow{P_k}(t) = \overrightarrow{P_k} + \overrightarrow{P'_k}(t) + \left[ \frac{3(\overrightarrow{P_{k+1}}(t) - \overrightarrow{P_k})}{t_{k+1}^2} - \frac{2P'_k}{t_{k+1}} - \frac{P'_{k+1}}{t_{k+1}} \right] t^2 \\
+ \left[ \frac{2(\overrightarrow{P_k} - \overrightarrow{P_{k+1}}(t))}{t_{k+1}^3} + \frac{P'_k}{t_{k+1}^2} + \frac{P'_{k+1}}{t_{k+1}^2} \right] t^3
\]
For any two adjacent spline segment equating the second derivatives at the common interal joints, i.e., letting

\[ \overrightarrow{P_k''}(t_k) = \overrightarrow{P_k''}(0) \]

\[ t_{k+2}P_k' + 2(t_{k+1} + t_{k+2})\overrightarrow{P_{k+1}'} + t_{k+1} + P_{k+2}' = \frac{3}{t_{k+1} \cdot t_{k+2}} \left[ t_{k+1}^2(P_{k+2} + P_{k+1}) + t_{k+2}^2(P_{k+1} - P_k) \right] \]

Applying this equation recursively over all the spline segment yields \( n-2 \) equations for the tangent vectors

\[ \overrightarrow{P_k'}, 2 \leq k \leq n - 1 \]
Ship Calculation (3) – Spline Curve

\[
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 \\
t_3 & 2(t_2 + t_3) & t_2 & 0 & 0 & 0 \\
0 & t_4 & 2(t_3 + t_4) & t_3 & 0 & 0 \\
0 & 0 & t_5 & 2(t_4 + t_5) & t_4 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & t_n & 2(t_n + t_{n+1}) & t_{n-1} \\
0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\begin{bmatrix}
P_1' \\
P_2' \\
P_3' \\
P_4' \\
\vdots \\
P_n'
\end{bmatrix}
\]

\[
\begin{bmatrix}
\frac{3}{t_2 t_3} \left[ t_2^2 (\vec{P}_3 - \vec{P}_2) + t_3^2 (\vec{P}_2 - \vec{P}_1) \right] \\
\frac{3}{t_3 t_4} \left[ t_3^2 (\vec{P}_4 - \vec{P}_3) + t_4^2 (\vec{P}_3 - \vec{P}_2) \right] \\
\vdots \\
\frac{3}{t_{n-1} t_n} \left[ t_{n-1}^2 (\vec{P}_n - \vec{P}_{n-1}) + t_n^2 (\vec{P}_{n-1} - \vec{P}_{n-2}) \right]
\end{bmatrix}
\]
Once $P'_k$ are known,

\[
\begin{align*}
\overrightarrow{B_{1k}} &= \overrightarrow{P_k} \\
\overrightarrow{B_{2k}} &= \overrightarrow{P'_k} \\
\overrightarrow{B_{3k}} &= \frac{3(\overrightarrow{P_{k+1}} - \overrightarrow{P_k})}{t_{k+1}^2} - \frac{2\overrightarrow{P'_k}}{t_{k+1}} - \frac{\overrightarrow{P'_k+1}}{t_{k+1}} \\
\overrightarrow{B_{4k}} &= \frac{2(\overrightarrow{P_k} - \overrightarrow{P_{k+1}})}{t_{k+1}^3} + \frac{\overrightarrow{P'_k}}{t_{k+1}^2} + \frac{\overrightarrow{P'_k+1}}{t_{k+1}^2}
\end{align*}
\]
Alternate cubic Spline end conditions.

<table>
<thead>
<tr>
<th>End condition</th>
<th>[M]</th>
<th>[R]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clamped</td>
<td>$M(1,1) = 1$</td>
<td>$R(1,1) = P_1$</td>
</tr>
<tr>
<td></td>
<td>$M(n,m) = 1$</td>
<td>$R(1,n) = P_n$ by user definition.</td>
</tr>
<tr>
<td>Relaxed</td>
<td>$M(1,1) = 1$</td>
<td>$R(1,1) = \frac{3}{2t_2} (\vec{P}_2 - \vec{P}_1)$</td>
</tr>
<tr>
<td></td>
<td>$M(1,1) = 1$</td>
<td>$R(n,1) = \frac{6}{t_n} (\vec{P}<em>n - \vec{P}</em>{n-1})$</td>
</tr>
<tr>
<td></td>
<td>$M(n,n-1) = 2$</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$M(n,n) = 4$</td>
<td></td>
</tr>
<tr>
<td>Cyclic</td>
<td>$M(1,1) = 2 \left(1 + \frac{t_m}{t_2}\right)$</td>
<td>$R(1,1) = 3 \left(\frac{t_n}{t_2^2}\right) (\vec{P}_2 + \vec{P}<em>1) - \frac{3}{t_n} (\vec{P}</em>{n-1} + \vec{P}_n)$</td>
</tr>
<tr>
<td></td>
<td>$M(1,2) = \frac{t_n}{t_2}$</td>
<td>$R(m,1) =$ undefined.</td>
</tr>
<tr>
<td></td>
<td>$M(1,n-1) = 1$</td>
<td></td>
</tr>
</tbody>
</table>
A Bezier curve is determined by a defining

A parametric Bezier Curve is defined by

\[
\vec{P}(t) = \sum_{i=0}^{n} B_i J_{ni}(t) \quad 0 \leq t \leq 1 \quad (1)
\]

Where the Bezier or Bernstein basic or blending.

\[
J_{n,i}(t) = \binom{n}{i} t^i (1 - t)^{n-i}
\]

With \[
\binom{n}{i} = \frac{n!}{i!(n-i)!}
\]

Note that 0 ≡ 1, 0’ ≡ 1
The properties of Bezier Curve
- The basic functions are real.
- The degree of the polynomial defining the curve segment is one less than the number of defining polygon points.
- The curve generally follows the shape of the defining polygon.
- The first and last points on the curve are coincident with the first and last points of the defining polygon.
- The tangent vectors at the ends of the curve have the same direction as the first and last polygon segments respectively.
- The curve is contained within the convex hull of the polygon obtainable with the defining polygon vertices.
- The curve exhibits the variation diminishing property.
The properties of B-Spline Curve

B- Spline curve of order K:

→ Piecewise continuous polynomial of degree K-1.
→ Continuously differentiable K-2 times.
→ $C^2$ continuity
→ As with Bezier, B-spline is uniquely defined by the vertices of a defining polygon

→ The order K of the B-spline and the number of vertices m may be chosen independently of each other as long as $m \geq K$.

  Bezier curve : $m = K$
  B-spline curve : $m > k$

  $m > k$ allows local modification
Ship Calculation (5) – B-Spline

B-spline curve, order k, m vertical (m ≥ K)

\[ P(t) = \sum_{i=0}^{n} B_i N_i, K(t) \]

\[ t_{\text{min}} \leq t \leq t, 2 \leq k \leq n+1 \]

Where the \( B_i \) : position vectors of the \( n+1 \) defining polygon vertices.

The \( N_i, K \) : normalized B-spline basic function. The basic function \( N_i, k \) are defined by the Cox de boor recursion formulas.

\[ N_{i,1}(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_{i,k-1} + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1,k-1} \]
Ship Calculation (5) – B-Spline

\[ N_{i,k}(t) = \begin{cases} 1; & 0 \leq t \leq t_{i+1} \\ 0; & \sum_{i=0}^{n} N_{i,k}(t) \end{cases} \]

An additional variable must be introduced.

The knot vector: \( J = (t_0, t_1, t_2, ..., t_n) \quad t_2 \leq t_{i+1} \)

It integers are used for \( t_i \) so called “uniform” B-spline

Example (0 1 2 3 4) -> T=(0 0 0 1 2 3 3 4 4 )
Ship Calculattion (5) – B-Spline

Vertices : \( K=4 \) m=4 (cubic Bezier curve) \( 0 \leq t_{i-1} \leq 1 \)

In general : total number of knots \( t_i, n \) knot.

Vector T: \( n-1 \)

Total number spans (including zero – length spans \( n+k-1 \))

- Number of interior knots
- Number of spans with non-zero length: \( m-k+1 \)
- Each blending function \( N_i,k \) is valued by using \( k+1 \) values from the knot vector \( (t_i, t_i+k) \)
- \( K=3, m=4 \) spans with nonzero length.