# Curve Fitting using Least Square Piecewise Parametric Spline under Tension

[1]Sakendra Kumar, Dr. Rajendra Prasad Central Agricultural University, Pusa,Samastipur-848125, Bihar, India, Email: sakendrakmr@gmail.com

[2] Parag Jain, Roorkee Institute of Technology Roorkee-247667, Uttarakhand, India

[3] Amit Kumar Tanwar, Roorkee Institute of Technology Roorkee-247667, Uttarakhand, India

[4]Deepak Arya, Roorkee Institute of Technology Roorkee-247667, Uttarakhand, India

[5] Rachan Karmakar, NIMS University, Jaipur, Rajasthan, India

**Abstract:**
Curved shapes are represented across CAD and CAM using parametric piecewise cubic functions. It would be helpful to accurately create this representation from a sparse collection of points that approximate and extrapolate the desired curve, such as the input from a digitising tablet or a scanner, for various mechanical applications. The modelling of a piecewise parametric cubic spline under strain is presented in this study. To tackle the interpolation problem, we created an algorithm that first models the curve and sequence of function values at a given location.

**Keywords:** Modeling, Interpolation, Cubic spine, Least Square, Curve Fitting, Piecewise, Parametric, Curves, Tension spline

## 1. Introduction

Modeling's primary goal is function recognition, which entails learning a collection of data's geometric qualities and using that knowledge to recommend a suitable function that will share those properties. The achievement of an acceptable and theoretically straightforward curve-fitting function is a secondary goal.

The goal of this study might be to assess the model or to utilize it to derive values that would not otherwise be possible as functions of the observed curve. Thus, parameter determination is the main challenge in modeling through curve-fitting. The form of the calculated curve is influenced by parameters, which are known or unknowable integers. The challenge is to identify the parameter values that, in terms of least squares, bring the calculated curve the closest to the observed curve. For this reason, we talk about modeling by fitting curves.

In order to allow tension factors to change with intervals, Schweikert [1] and S path [15] created tension splines with uniform tension. Pruess [7] has studied the convergence rates and asymptotic behavior of interpolatory tension splines with varying tension, while Cline [2] has created software for both interpolation and smoothing using splines with uniform tension. Lynch [13] and Kumar [17] provide methods and software for selecting a

tension factor that satisfies bounds on function values and derivatives while maintaining local convexity. Spith [15] and Kumar [16] describe an iterative procedure for selecting tension factors to avoid unnecessary inflection points. However, because the latter approach only works in the scenario of uniform stress, it typically produces far more tension than is required at certain periods.

**1.1 Piecewise Parametric Cubic Spline in Tension**

A tension spline is a spline curve with a parameter that controls the curve's tension. This means that the curve that passes through the points of the data constraint might have a varied tension, making it more or less stiff. The tension spline behaves like a cubic spline when the tension is low (around 0). The tension spline approaches the piecewise linear function when the tension is high (moving to infinity). The cubic spline polynomial is not equivalent to a tension spline with tension greater than zero.

We have periodic boundary conditions and n data constraint points to deal with. See [4] through [6] for a more thorough and broad description. Here is a description of the tension spline curve's mathematical model. Given n knots and $d_1<d_2<\ldots\ldots<d_n$ data $y_i$ at each $d_i$ (i=1,2,....n). Periodicity is provided by the data value $y_n = y_1$ for the last knot dn. The function f with the following characteristics is the tension spline that we desire. [1]

i. $f \in C^2\ [d_1,d_n]$

ii. $f(d_i)= y_i \qquad (1 \le i \le n)$

iii. *On each open interval* $(d_i, d_{i+1})$, *f satisfies* $f^{(4)} - \tau f'' = 0$

The characteristics I to (iii) imply that f interpolates the provided data, has two continuous derivatives globally and satisfies a particular differential equation in each sub-interval. Since the solutions of the equation $f^{(4)}=0$ are cubic polynomials, it is evident that this prescription produces a cubic spline when $\tau = 0$. Zi is set to equal f"(di) in order to calculate f, and the requirements for f on the range [di, di+1] are written down.

$f^{(4)} - \tau f'' = 0$

$f(d_i) = y_i \qquad f(d_{i+1}) = y_{i+1}$

$f''(d_i) = z_i \qquad f''(d_{i+1}) = z_{i+1}$

Where, $z_n = f''(d_n) = f''(d_1) = z_1$. One can verify that the solution of this two-point boundary-value problem is $f(x) = \{ z_i \sinh[\tau(d_{i+1} - x)] + z_{i+1} \sinh [\tau(x-d_i)\}/ [ \tau^2 \sinh (\tau h_i)] + (y_i - z_i /\tau^2)(d_{i+1}-x)/h_i + (y_{i+1}-z_{i+1}/\tau^2)(x-d_i)/h_i$

Where $h_i = d_{i+1} - d_i$. After the coefficients $z_i$ have been determined, these equations will be used to compute values of*f* on the interval $[d_i, d_{i+1}]$. The function *f* is what we will call the tension spline function.

In order that *f* have $C^2$ global smoothness, the conditions

$lim_{x \uparrow d_i} f(x) = lim_{x \downarrow d_i} f(x) (1 \le i \le n)$ must be imposed at the knots. The tedious calculations involved in this are not given here.

The result is a tridiagonal system of equations for the unknowns $z_1, z_2 \ldots\ldots\ldots z_n$ that can be written in the form

$$\alpha_{i-1}z_{i-1} + (\beta_{i-1} + \beta_i)z_i + \alpha_i z_{i+1} = \gamma_i - \gamma_{i-1}$$

$$(1 \leq i \leq n-1) \qquad (1)$$

With these abbreviations

$\alpha_i = 1/h_i - \tau \sinh(\tau h_i)$

$\beta_i = \tau \cosh(\tau h_i) / \sinh(\tau h_i) - 1/h_i$

$\gamma_i = \tau^2(y_{i+1}-y_i)/h_i$

Since (1) produces (n-1) equations with n unknowns, it is noted that a further need is necessary to ascertain the z-vector. We now have the linear system of equations, which is given by the periodic boundary condition, defined by $y_1 = y_n$ and (really, it is the case that $y_i = y_i+(n-1)$and $z_i = z_i+(n-1)$).

$$\begin{bmatrix} (\beta_{n-1}+\beta_1) & \alpha_1 & \cdots & \alpha_{n-1} \\ \alpha_1 & (\beta_1+\beta_2) & \alpha_2 & \vdots \\ \vdots & \alpha_2 & \ddots & \vdots \\ \alpha_{n-1} & \cdots & \cdots & (\beta_{n-2}+\beta_{n-1}) \end{bmatrix} \begin{bmatrix} z_1 \\ \vdots \\ \vdots \\ z_{n-1} \end{bmatrix} = \begin{bmatrix} (\gamma_1-\gamma_0) \\ \vdots \\ \vdots \\ (\gamma_{n-1}-\gamma_{n-2}) \end{bmatrix} \quad (2)$$

Where$\gamma_0 = \tau^2 (y_1 - y_{n-1})$. By solving the equations (invert the symmetric matrix in (2)) all $z_i$ can be determined. That means the tension spline function $f$ will be given. The constant $\tau$ is called the tension.

## 2. Modules of Interpolator Splines under Tension

The modules of splines under tension includes the two algorithm, which solves the interpolation and determine a real valued function are $\{y_i\}_i^n=1$ at abscissas $\{x_i\}_i^n=1$.

### 2.1. Interpolation through sequence of function value

The first method uses functional values to define the parameters necessary to calculate an interpolatory spline under tension. To add points on the curve, the slopes at the two ends of the curve may be supplied or omitted.The algorithm's input parameters are Ni, the number of values to be interpolated, X, an array of the N functional values' rising abscissa, Y, an array of the N values' ordinates; and FD1&FDN, which have the predicted values for the curve's 1-derivative at X(l) and X(N), respectively. The tension factor is included in the arrays ARR, TEMP, and TF, each of which has a length of at least N and is used for scratch storage.

The output parameters ARR must have values proportionate to the curve's second derivative at the specified nodes.

*Algorithm:*

1. Function

```
FUNC1 (N, X, Y, FD1, FDN, ARR, TEMP, TF)
```

2. Input

```
INTEGER N
REAL X (N), Y (N), FD1, FDN, ARR (N), TEMP (N), TF
NM = N - 1
NP = N + 1
LXI = X(2) - X(1)
XI = (Y(2)-Y(1))/DELXI
```

3. Calculate slopes if required

```
IF (TF.LT.0.) GO TO 50
FDD1 = FD1
FDDN = FDN
```

4. Ab-normalize the stress factor

```
10 TFD=ABS (TF)*FLOAT (N-1)/(X (N)-X (1))
```

5. Set up the right side, the tridiagonal system, and make the progressive reduction.

```
      LS = TFD*LXI
      PS = EXP (LS)
      HS =0.5*(PS-1/PS)
      HIN = 1./(LXI*HS)
      GI = HIN*(LS*0.5*(PS*1.
/PS)-HS)
      GIN = 1. /GI
      ARR(1) = GIN*(XI-FDD1)
      DIAG =HIN*(HS-LS)
      TEMP(1) = GIN*DIAG
      IF (N.EQ.2) GO TO 30
      DO 20 I=2,NM
      LX2 = X(I+1) - X(I)
      X2 = (Y(I+1)-Y(1)/LX2
      LS = TED*LX2
      PS = EXP(LS)
      HS = .5*(PS-1./PS)
      HIN = 1./(LXI*HS)
      GG=HIN*(LS*(0.5*(PS*1./P
S))-HS)
      GIN =1./(GI+GG-
DIAG*TEMP(I-1))
      ARR(I) = GIN*(X2-X1-
DIAG*ARR(I-1))
      DIAG =HIN*(HS-LS)
      TEMP(1) = GIN*DIAG
      X1 = X2
      GI = GG
      20 CONTINUE
      30 GIN = 1./{GI-
DIAG*TEMP(NM))
      YP(N) = GIN*(FDDN-X2-
DIAG*ARR(NM))
```

6. Perform back substitution

```
      DO 40 I=2*N
      IK = NP-I
      ARR (IK) = ARR (IK)-TEMP
(IK)*ARR (IK+1)
      40 CONTINUE
      RETURN
      50 IF (N.EQ.0) G0 TO 60
```

7. Use the second-order polynomial regression on the input data to determine values at endpoints when no derivatives are provided.

```
      LX2 = X(3) - X(2)
      LX12 = X(3) - X(1)
      D1 = -
(LX12+LX1)/LX12/LX1
      D2 = LX12/LX1/LX2
      D3 = -LX1/LX12/LX2
      FDD1 = D1*Y (1) + D2*Y
(2) + D3*Y (3)
      LN = X(N) - X(NM)
      LNM = X(NM) - X(N-2)
      LNN = X(N) - X(N-2)
      D1 = (LNN+LN)/LNN/LN
      D2 = -LNN/LN/LNM
      D3 = LN/LNN/LNM
      FDDN = D3*Y (N-2) + D2*Y
(NM) ÷ D1*Y (N)
      GO TO 10
```

8. Use a straight line for a curve if there are just two points and no derivatives are provided.

```
      60 ARR (1) = 0.
      ARR (2) = 0,
      RETURN
```

9. End

## 2.2. Interpolate a curve at a given point

Using a spline under stress, this technique interpolates a curve at a specified point. The subroutine curve has to be called earlier to get some crucial parameters.

The algorithm requires the following input parameters: X and Y are arrays comprising the interpolated points' ordinates and abscissas. ARR is an array with values proportional to the second derivative of the curve at the nodes. TF is a numeric switch that includes the tension factor. T is a real value to map onto the interpolating curve. Output parameters contain the interpolated results for T fewer than X (1), function = Y (1), for T larger than X (N), function = Y (N).

*Algorithm:*

1. Function

```
      FUNC2 (T, N, X, Y, ARR,
TF, IT)
```

2. Input

```
      INTEGER N, IT
      REAL T, X (N), Y(N),
ARR(N), TF
      S = X (N) - X (1)
```

3. Denormalize sigma

```
      TFD= ABS (TF)*FLOAT (N-
1)/S
```

4. If IT.NE.1 resume the investigation where it was earlier stopped; otherwise, begin again

```
      IF (IT.EQ.1) I1 = 2
```

5. Explore for interval

```
      10 DO 20 I=I1, N
IF (X (I)-T) 20, 20, 30
      20 go on
      I=N
```

6. Check to insure correct interval

```
      30 IF (X(I-1).LE.T .OR.
T.LE.X(1)) GO TO 40
```

7. Resume explore and rearrange it

```
      (INPUT ''IT'' WAS
INCORRECT)
      I1 = 2
      GO T0 10
```

8. Set up and perform interpolation

```
      40 L1 = T - X(I-1)
      L2 = X(1) - T
      S1 = EXP(TFD*L1)
      D1 =.5*(S1-1*/S1)
      PS = EXP(TED*L2)
      D2 = .5*(PS-1./PS)
      PS = PS1*PS
      HS = .5*(PS-I./PS)
      FUNC2 =
      (ARR(1)*D1+ARR(I-
      1)*D2)/HS *((Y(I)-
      ARR(I))*L1+(Y(I-1)-
      ARR(I-1))*L2)/LS
      I1 = I
      RETURN
```

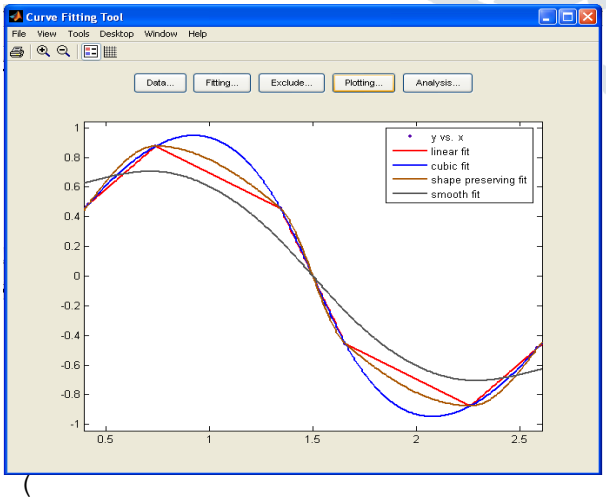
9. End

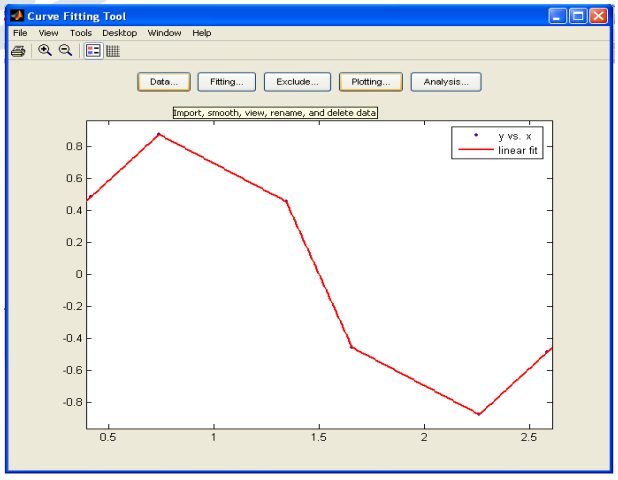
## 3. Modeling using MATLAB Programming

We consider the piecewise cubic spline curve with four two-dimensional position vectors $P_1$ [0 0], $P_2$ [1 1], $P_3$ [2 -1] & $P_4$ [3 0], tangent vectors at the ends are $P'_1$ [1 1] & $P'_4$ [1 1], tension factor $\tau$ is 0.2. The intermediate points at t = 1/3 & 2/3 for each segment are:

| Segment | 1 | | 2 | | 3 | |
|---|---|---|---|---|---|---|
| t | 1/3 | 2/3 | 1/3 | 2/3 | 1/3 | 2/3 |
| $P_x$ | 0.416 | 0.740 | 1.343 | 1.657 | 2.260 | 2.584 |
| $P_y$ | 0.484 | 0.876 | 0.457 | -0.457 | -0.876 | -0.484 |

We can import predictor $P_x$ (X) data, response $P_y$ (Y) data and weights, the fig.1 shows the curve fitting result of interpolatory cubic spline.




(
I-I)

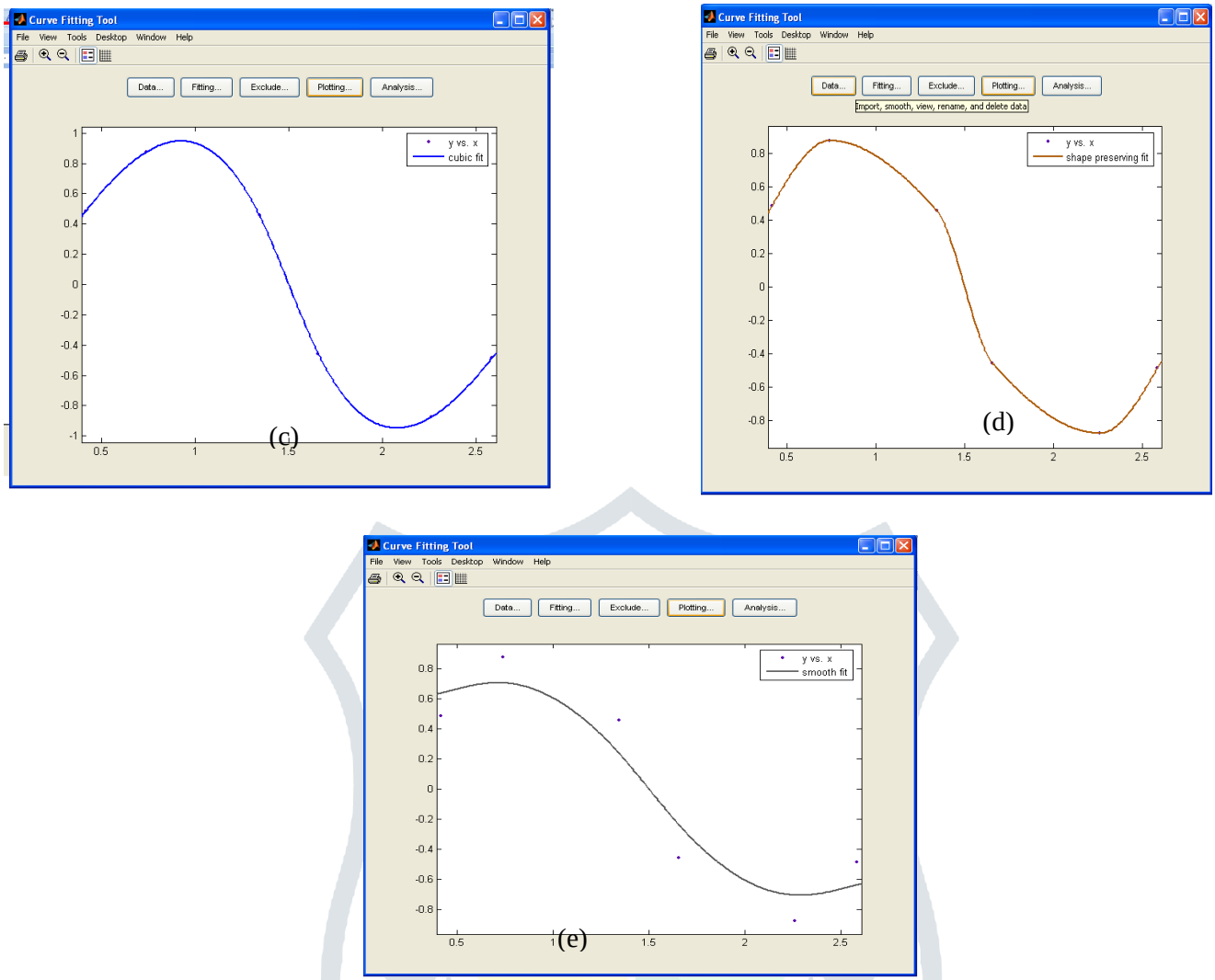



**Fig. 1:** Curve fitting graph (a) common (b) linear fit under tension (c) cubic spline fit under tension (d) shape preserving fit under tension (e) smooth fit under tension

We may estimate (interpolate or extrapolate), separate, or integrate a fit across a particular data arrangement with the analytical technique.

| $X_i$ | $f(X_i)$ | $df(X_i)/dX$ | $d^2f(X_i)/dX^2$ | Integral $f(X_i)$ |
|---|---|---|---|---|
| 0.416 | 0.484 | 1.55739 | -1.45861 | 0 |
| 0.6328 | 0.776567 | 1.09177 | -2.8368 | 0.138469 |
| 0.8496 | 0.935799 | 0.327357 | -4.21499 | 0.327084 |
| 1.0664 | 0.896917 | -0.73585 | -5.59318 | 0.529915 |
| 1.2832 | 0.595143 | -2.09784 | -6.97137 | 0.696989 |
| 1.5 | 0 | -3.10319 | 0 | 0.765635 |
| 1.7168 | -0.59514 | -2.09784 | 6.97137 | 0.696989 |
| 1.9336 | -0.89692 | -0.73585 | 5.59318 | 0.529915 |
| 2.1504 | -0.9358 | 0.327357 | 4.21499 | 0.327084 |
| 2.3672 | -0.77657 | 1.09177 | 2.8368 | 0.138469 |
| 2.584 | -0.484 | 1.55739 | 1.45861 | 0 |

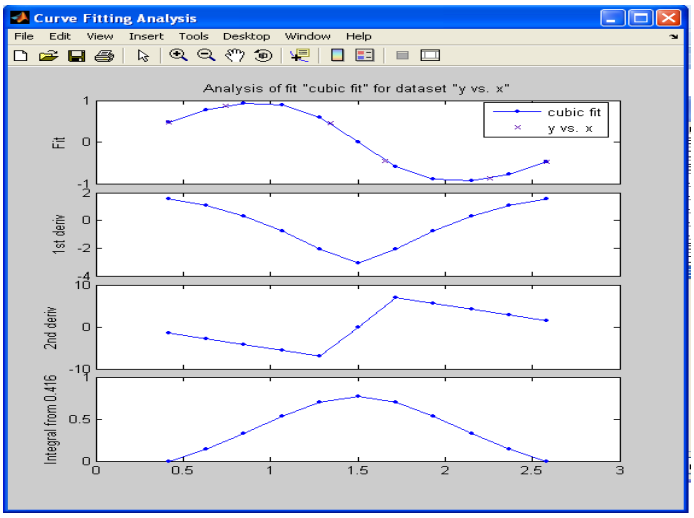



**Fig. 2:** Analysis of cubic fit under tension

## 4. Conclusion

This work describes the mathematical modeling of the least square piecewise parametric spline under strain. We've spoken about the conceptual and mathematical aspects of splines under tension Finally, the study's objective has been reached. This study aims to produce interpolatory spline modules under tension by interpolating a curve at a certain location using a series of functional values. Therefore, a suggested interpolatory method has been modeled in MATLAB programming. The other spline that is modeled under stress will be explored in the future study.

## 5. References

[1] Schweikert, D.G. “An interpolation curve using a spline in tension. J. Math. and Physics” 45 1966), 312 317.

[2] Cline, A.K. “Scalar and planar valued curve fitting using splines under tension”Comm. ACM 17, 4 (Apr. 1974), 218-220.

[3] R. Franke, "A critical comparison of some methods for interpolation of scattered data",Naval Postgraduate School Techn. Rep. NPS-53-79-003, March 1979)

[4] C. de Boor, "Efficient computer manipulation of tensor products"ACM Trans. Math. Software 5 (1979), 173–182; Corrigenda, 525

[5] Ahlberg, J.H., Nilson, E.N., and Walsh, J.L., “The Theory of Splines and their Applications”. Academic Press, New York, 1967.

[6] Kincaid, D and Cheney, W. 1996. “Numerical Analysis”, second edition. Brooks/Cole.

[7] Pruess, S. 1976. “Properties of splines in tension” JAT 17, 86-96.

[8] H. Akima, “A new method of interpolation and smooth curve fitting based on local procedures”, J. ACM,Comput. Math., 17 (1970), pp. 589-602.

[9] F. N. Fritsch and J. Butland, “A method for constructing local monotone piecewise cubic interpolants”, this Journal, 5 (1984), pp. 300-304.

[10] D. F. Mcallister, E. Passow and J. A. Roulier, “Algorithms for computing shape preserving spline interpolations to data”, Math. Comp., 31 (1977), pp. 717-725.

[11] R. J. Renka,“Interpolatory tension splines with automatic selection of tension factors”, SIAM J. SCI. STAT. COMPUT. Vol. 8, No. 3, May 1987

[12] PRICE, J. F., AND SIMONSEN, R. H. "Various methods and computer routines for approximation, curve fitting and interpolation" Doc. I) 1-82-0151 Rev., Boeing Sci. Res. Lab., Seattle, Wash., July 1963.

[13] R.W. LYNCH, A method for choosing a tension factor for spline under tension interpolation, M. A. Thesis, Univ. Texas, 1982.

[14] Aho, A.V., Hopcroft, J.E. and Ullman, J.D., "The Design and Analysis of Computer Algorithms", Addison-Wesley, Reading, Massachusetts, 1974.

[15] H. SPTH, "Exponential spline interpolation, Computing", 4 (1969), pp. 225-233.

[16] Kumar, S., & Tewari, S. P. (2018). Effect of mold oscillation on the metallurgical characterization and mechanical properties of A319 aluminum alloy casting. International Journal of Cast Metals Research, 31(1), 1-6.

[17] Kumar, S., & Tewari, S. P. (2018). Metallurgical and mechanical characterization of A319 aluminum alloy casting solidified under mold oscillation. International Journal of Metalcasting, 12(1), 28-35.