IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Analysis Of Non-Linear 2-Dimensional Motion Of A Rotating Member By The Newton-Gregory Interpolation Method And Java Application

[1]Sudha Patil, [2]Kshitijarun Y. Bidari, [3]Supriya Rajput
[1]Assistant Professor, [2]Assistant Professor, [3]Assistant Professor
[1]Applied Science Department,
[1]Maratha Mandal's Engineering College, Belagavi, Karnataka, India.

Abstract: Accurate analysis of motion parameters is essential in various engineering applications involving rotating mechanical systems. This study presents a numerical approach for analyzing non-linear two-dimensional rotational motion using the Newton–Gregory forward interpolation method. The research considers time and corresponding angular displacement data of a wind turbine blade to determine the unknown motion parameters such as angular velocity and angular acceleration. The mathematical formulation is validated through computational implementation using the Java programming language to improve numerical accuracy and computational efficiency. The results obtained from the computational model show strong agreement with the classical numerical method, with minimal percentage error, demonstrating the reliability of the proposed approach. The study highlights the effectiveness of combining numerical interpolation techniques with computational tools for real-time engineering motion analysis and data-driven system modeling.

***Index Terms* -** Newton Gregory Interpolation, Non-Linear Motion Analysis, Angular Velocity, Angular Acceleration, Wind Turbine Blade Motion, Numerical Methods, Java Programming, Computational Modeling, Engineering Motion Analysis.

I. Introduction

The modern fast advancing world demands the use of modern methods and tools for various applications. In earlier times the analysis of any member along certain motion was done using classical method in the field of theoretical mathematics. This research article attempts to give solution to a similar kind of mathematical problem that involves the motion of a 1D element in a 2-Dimensional space.

The present world is experiencing a huge burst of data which if left un-analyzed can lead to detrimental losses to environment and life. The field of data science makes use of mathematical models which, when applied to software tools like Python, R, Java, C can give highly beneficial results that can help increase the efficiency of performance of a system by leveraging the data. This research paper involves applying the Newton-Gregory Forward Interpolation technique to create a mathematical model to analyze a given set of data by the classical method and by the Java application tool , thereby obtaining the values for certain unknown parameters. There are a number of tools that can analyze data in the form of known values to obtain new unknown parameters, but this research uses Java software application to apply the mathematical model, interpolate it and obtain the unknown data.

It is seen that with the stacking of huge amounts of data, day by day its becoming more and more complex to analyze and utilize the same to obtain meaningful insights or results. If we consider the real world motion of a rocket, missile or an aero-plane, it is clear that the motion is not always linear especially in case of an aero-plane or a jet and it is required to analyze the different parameters of motion like position, velocity and acceleration at different time periods which if not done can lead to detrimental effects on human life. This research involves analyzing the motion of a similar kind in a rotational plane where the velocities and accelerations are obtained for various time internals at specific positions.

It is observed that in the real-world there are numerous examples of machines and objects that exhibit physical linear, rotational, trajectory and planar motions. Some of the crucial motions of objects that need to be analyzed are the motion of vanes/blades of a wind mill, blades of a jet engine, reciprocating motion of a piston, rotational motion of a crank shaft and camshaft. Few of the fore-mentioned examples exhibit high speed but are dimensionally small in size like the crankshaft and camshaft; and few of them exhibit low speeds comparatively but are considerably large in size for instance the blades/vanes of a wind mill. This research specifically deals in analyzing the rotation of vanes of a wind turbine where the motion exhibited is non-linear due to the varying of the wind flow density at different times in day.

Initially a wind turbine is considered of standard size and dimensions and the different positions of a single blade at different time periods are taken using which the N-G Interpolation method and java application tool are applied to obtain the corresponding angular velocities and accelerations.

II. PROBLEM DEFINITION

A standard wind turbine will have three to four blades based on the conditional requirement. As all the blades will be moving at the same velocity and acceleration with respect to each other, we are considering the positions of only 1 blade for the analysis. Because all the blades are fixed on the same center hub, there is no need to consider all the blades.

The problem considers a blade of unit mass of a wind mill rotating at nominal speed of 40 to 50 rpm where the wind velocity ranges between 20 to 25 kmph. Here the rotational motion of the blades is low as compared to the other mechanical rotating members, whereas the size of the blade is large. But that will not be an affecting parameter for the analysis as the analysis in itself will not consider the weight and size, as in the case of study of kinematics of motion where the mass and size are also considered for the analysis. The problem here, takes the different positions and their respective time periods as the input and obtains the respective angular velocities and accelerations.

A blade of the wind mill which is considered as a one-dimensional rod rotates in a plane of two-dimensions (i.e., a plane) and exhibits non-linear motion. Non-Linear motion in this case indicates that the change in the position is not uniformly increasing or decreasing with respect to time. The positions and the respective time periods of the moving rod element are as given below:

Table 2.1:The positions and the respective time periods of the moving rod.

time = t	**0**	**0.2**	**0.4**	**0.6**	**0.8**	**1**	**1.2**
Angle = Ɵ	**0**	**0.12**	**0.49**	**1.12**	**2.02**	**3.2**	**4.67**

The above dataset is obtained from a standard wind mill of 1 KW capacity located at an approximate altitude of 300 feet above the ground level in the North Karnataka Region.

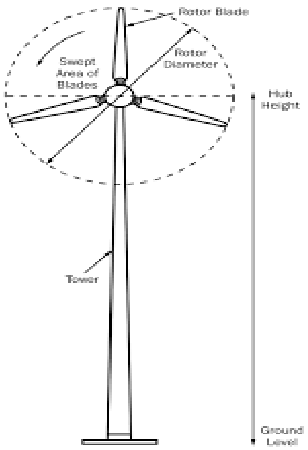

Fig. 2.1 Wind Mill Blade Used for Rotational Motion Analysis

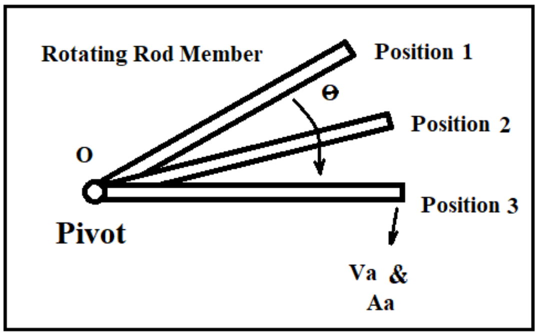

Fig. 2.2 Rotational Motion Representation of Wind Mill Bladein 2-D Plane

III. METHODOLOGY

The methodology of research basically is divided into 2 categories the first one deals with obtaining the solution through the Newton-Gregory Interpolation formula which takes the above parameters as input and generates the difference table. The values from the difference table are substituted in the formula and the corresponding angular velocities and acceleration are obtained. The process of obtaining the solution involves performing multiple iterations with different input values.

The second part of the study involves writing the Java code for the numerical method in the core Java format. The inputs are taken as the values of time period (t) and their corresponding positions (Θ) in radians. The code is executed and the output is obtained to give the values of velocities and accelerations.

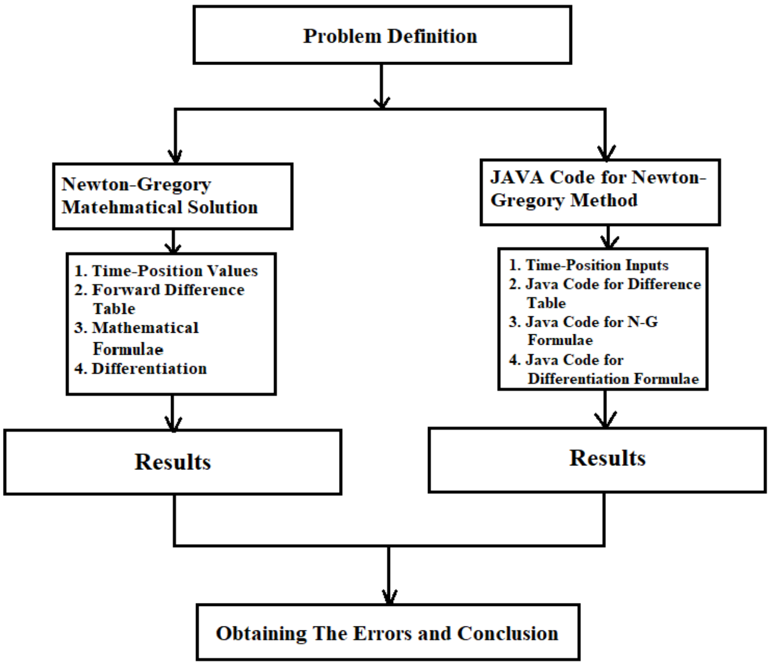

Fig3.1.Block diagram of Proposed Methodology.

3.1 Solution using newton-gregory interpolation formula

The Newton-Gregory Forward Interpolation Formula can be used to determine specific parameters from a given set of data by the method of forward interpolation. Here the problem aims at obtaining the angular velocity and acceleration at any given point or a number of points by using the above formula.

Newton-Gregory Forward Interpolation Formula to compute angular velocity = $d\Theta/dt$ and angular acceleration = $d^2\Theta/dt^2$ for a given point or a specific set of points.

$$f(x_0 + rh) = y_0 + r\Delta y_0 + (r(r-1)/2!)*(\Delta^2 y_0) + ((r(r-1)(r-2))/3!)*(\Delta^3 y_0) + ((r(r-1)(r-2)(r-3))/4!)*(\Delta^4 y_0) + \ldots$$

The above formula in general form can be written as:

$$f(x_0 + rh) = \Theta_0 + r\Delta\Theta_0 + (r(r-1)/2!)*(\Delta^2\Theta_0) + ((r(r-1)(r-2))/3!)*(\Delta^3\Theta_0) + ((r(r-1)(r-2)(r-3))/4!)*(\Delta^4\Theta_0) + \ldots$$

The notations in the above formula are as below:

x = required value whose angular velocity and acceleration needs to be obtained

$x_0 = t_0$ = initial value of the Time period

$t_1, t_2, t_3, t_4, t_5, t_6, t_7$ - values of time periods

$\Theta_1\ \Theta_2\ \Theta_3\ \Theta_4\ \Theta_5\ \Theta_6\ \Theta_7$ – values of position/displacement

$y_0 = \Theta_0$ = initial value of the position of rotation

$\Delta y_0 = \Delta\Theta_0$ = initial value of the 1st forward difference (as shown in the table)

$\Delta^2 y_0 = \Delta^2\Theta_0$ = initial value of the 2nd forward difference (as shown in the table)

$\Delta^3 y_0 = \Delta^3\Theta_0$ = initial value of the 3rd forward difference (as shown in the table)

$\Delta^4 y_0 = \Delta^4\Theta_0$ = initial value of the 4th forward difference (as shown in the table)

$\Delta^5 y_0 = \Delta^5\Theta_0$ = initial value of the 5th forward difference (as shown in the table)

h = Step size of time period

r = ratio of required value minus the initial value to the time period step size

$r = (x - x_0)/h = (0.6 - 0)/0.2 = 3$

Table 3.1 The Forward difference table is as follows:

t	**y or Θ**	**$\Delta\Theta$**	**$\Delta^2\Theta$**	**$\Delta^3\Theta$**	**$\Delta^4\Theta$**	**$\Delta^5\Theta$**	**$\Delta^6\Theta$**
t0 = 0	$\Theta0 = 0$						
		$\Delta\Theta_0 = 0.12$					
t1 = 0.2	$\Theta1 = 0.12$		$\Delta^2\Theta_0 = 0.25$				
		$\Delta\Theta1 = 0.37$		$\Delta^3\Theta_0 = 0.01$			
t2 = 0.4	$\Theta2 = 0.49$		0.26		$\Delta^4\Theta_0 = 0$		
		0.63		0.01		$\Delta^5\Theta_0 = 0$	
0.6	1.12		0.27		0		$\Delta^6\Theta_0 = 0$
		0.9		0.01		0	
0.8	2.02		0.28		0		
		1.18		0.01			
1	3.2		0.29				
		1.47					
1.2	4.67						

The values in the table are obtained as follows:

$\Delta\Theta_0 = \Theta1 - \Theta0 = 0.12 - 0 = 0.12$

$\Delta^2\Theta_0 = \Delta\Theta1 - \Delta\Theta_0 = 0.37 - 0.12 = 0.25$

$\Delta^3\Theta_0 = \Delta^2\Theta_1 - \Delta^2\Theta_0 = 0.26 - 0.25 = 0.01$

$\Delta^4\Theta_0 = \Delta^3\Theta_1 - \Delta^3\Theta_0 = 0.01 - 0.01 = 0$

(Iterations need not be continued as the value 0 is attained which indicates no further iterations needed)

3.2 Obtaining the angular velocity

Initially the Newton-Gregory Interpolation Equation needs to be differentiated for obtaining the Angular Velocity.

$f(x_0 + rh) = y_0 + r\Delta y_0 + (r(r-1)/2!)*(\Delta^2 y_0) + ((r(r-1)(r-2))/3!)*(\Delta^3 y_0))$ ------------ (a)

Simplifying the above equation, we get :

$f(x_0 + rh) = y_0 + r\Delta y_0 + ((r^2 - r)/2!)*(\Delta^2 y_0) + ((r^3 - 3r^2 + 2r)/3!)*(\Delta^3 y_0))$

Differentiating the above equation, we get:

$f'(x_0 + rh).h = \Delta y_0 + ((2r-1)*(\Delta^2 y_0 / 2) + ((3r^2 - 6r + 2))*(\Delta^3 y_0 /6))$ --------------- (b)

Substituting "ϴ" in place of "y" because the dependent variable "y" here is referred as "ϴ"

$f'(x_0 + rh).h = \Delta \Theta_0 + ((2r-1)*(\Delta^2 \Theta_0 / 2) + ((3r^2 - 6r + 2))*(\Delta^3 \Theta_0 /6))$ ---------- (c)

Substituting the values of "r" and "h" in the above equation:

$f'(0.6)*(0.2) = 0.12 + ((2(3)-1)*(0.25 / 2))+(((3(3)^2 - 6(3) + 2))*(0.01 /6))$

$f'(0.6)*(0.2) = 0.12 + (5*0.125) + (11*0.00167)$

$f'(0.6)*(0.2) = 0.12 + 0.625 + 0.01837$

$f'(0.6)*(0.2) = 0.764$

$f'(0.6) = 0.764 / 0.2$

$f'(0.6) = 3.82$

The Angular Velocity (dϴ/dt) of the rod element at position 0.6 radians is 3.82 rad/sec

3.3 Obtaining the angular acceleration

Consider the fore-mentioned equation (b), which is:

$f'(x_0 + rh).h = \Delta y_0 + ((2r-1)*(\Delta^2 y_0 / 2) + ((3r^2 - 6r + 2))*(\Delta^3 y_0 /6))$

This equation has to be differentiated again to obtain the angular acceleration. The differentiated equation is as follows:

$f''(x_0 + rh).h^2 = \Delta^2 y_0 + ((6r-6)*(\Delta^3 y_0 / 6)$

$f''(x_0 + rh).h^2 = \Delta^2 y_0 + ((r-1)*(\Delta^3 y_0))$

Substituting "ϴ" in place of "y" because the dependent variable "y" here is referred as "ϴ"

$f''(x_0 + rh).h^2 = \Delta^2 \Theta_0 + ((r-1)*(\Delta^3 \Theta_0))$

Substituting the values of "r" and "h" in the above equation:

$f''(0.6)*(0.2)^2 = 0.25 + ((3 - 1)*(0.01))$

$f''(0.6)*(0.04) = 0.25 + 0.02$

$f''(0.6) = 0.27 /0.04$

$f''(0.6) = 6.75$

The Angular Acceleration ($d^2\Theta/dt^2$) of the rod element at position 0.6 radians is 6.75 rad/sec^2

3.4 Results obtained through classical method.

After following the steps as mentioned in the previous head for the other values of time and position, the different angular velocities and accelerations obtained are as listed below:

Table.3.2 – Shows the Velocity and Acceleration values as obtained by the Classical Method

Sl.No	Time(t)	Displacement(Θ)	Velocity	Acceleration
1	0	0	0	0
2	0.2	0.12	1.3	6.3
3	0.4	0.49	2.5	6.5
4	0.6	1.12	3.82	6.75
5	0.8	2.02	5.2	7
6	1	3.2	6.62	7.25
7	1.2	4.67	8.1	7.5

Computational Implementation Using Java Programming Language

To obtain accurate numerical solutions for the motion parameters, a computational approach is adopted using the Java programming language for mathematical modeling and numerical analysis. The interpolation method is implemented in a structured Java-based computational environment to efficiently process the given time and displacement data. The program is designed to accept experimental or observed values as inputs and perform numerical computations based on the Newton–Gregory interpolation technique. This approach helps in minimizing manual calculation errors and enables faster evaluation of angular velocity and angular acceleration at any desired time instant. The use of Java-based computational tools improves the reliability, repeatability, and accuracy of the results, making the method suitable for practical engineering applications and real-time motion analysis.

IV. RESULT

The below table shows the results as obtained by the Java application in which the values are obtained accurate upto 7 decimal places which cannot be easily possible with the classical method being solved manually and doing which for all the different positions seems to be cumbersome.

Table 3.3 –Results obtained by the Java Application

Sl.No	Time(t)	Displacement(Θ)	Velocity	Acceleration
1	0	0	0	0
2	0.2	0.12	1.2166667	6.2499995
3	0.4	0.49	2.4916666	6.4999995
4	0.6	1.12	3.816666	6.749999
5	0.8	2.02	5.1916666	6.9999986
6	1	3.2	6.6166663	7.2499986
7	1.2	4.67	8.091666	7.499998

As can be observed in the above table the results of the Java application are closely precise accurate as compared to the values obtained by the classical method. Moreover, they are much more accurate than the former values.

The results as obtained by both the methods are given below and their comparison shows that the difference between both the corresponding values is negligible showing that the numerical method and the Java code corroborate each other. The average error obtained below is seen to be 1.3% for the angular velocity and 0.714% for the angular acceleration. Due to the error being so low it can be concluded that the method of analysis is acceptable.

Table.3.4 – Shows the Error of the Velocity and Acceleration values as obtained by the Classical and Java Method.

Sl.No	Methods				% Error	
	Classical Method		Java Code			
	Velocity	Acceleration	Velocity	Acceleration	Velocity	Acceleration
1	0	0	0	0	0	0
2	1.3	6.3	1.2166667	6.2499995	8.33%	5.0005%
3	2.5	6.5	2.4916666	6.4999995	0.3%	0.000007%
4	3.82	6.75	3.816666	6.749999	0.087%	0.0000148%
5	5.2	7	5.1916666	6.9999986	0.16%	0.000002%
6	6.62	7.25	6.6166663	7.2499986	0.15%	0.000019
7	8.1	7.5	8.091666	7.499998	0.1%	0.0000267
					Avg =1.3%	Avg = 0.714%

V. CONCLUSION

The present study successfully demonstrates the effectiveness of the Newton Gregory interpolation technique for analyzing non-linear rotational motion parameters using both classical mathematical formulation and computational implementation through the Java programming language. The developed computational model enabled precise estimation of angular velocity and angular acceleration from discrete time displacement datasets, thereby validating the suitability of numerical interpolation methods for real-world dynamic motion analysis. The results obtained from the Java-based computational approach showed high accuracy and strong agreement with the classical method, with minimal percentage error, confirming the reliability and robustness of the proposed methodology. Furthermore, the use of computational tools significantly reduces manual calculation complexity, enhances computational efficiency, and enables high-precision analysis. Therefore, the proposed approach can be effectively applied to various engineering domains such as wind energy systems, rotating machinery, aerospace motion analysis, and real-time mechanical system monitoring, thereby contributing to advanced data-driven engineering analysis and design.

REFERENCES

[1] Kreyszig, E. Advanced Engineering Mathematics, 10th Edition, Wiley, New York, 2011.

[2] Grewal, B. S. Higher Engineering Mathematics, Khanna Publishers, New Delhi, 2017.

[3] Thomas, G. B., Weir, M. D., and Hass, J. Thomas' Calculus, 12th Edition, Pearson, Boston, 2010.

[4]4 eisstein, E. W. MathWorld – A Wolfram Web Resource, Wolfram Research, 1999.