

**APPLICATIONS OF POLYNOMIAL INTERPOLATION**

REPORT BY – ABHIRAM D V

COURSE TITLE - LINEAR ALGEBRA

PROJECT GROUP – 8

INSTITUTE - INTERNATIONAL INSTITUTE OF INFORMATION TECHNOLOGY, HYDERABAD

DATE – 21st November 2020

**ABSTRACT**

In this work, we will see what is interpolation of polynomials is and various techniques involved in solving of the interpolation of polynomials. We mainly look at Vandermonde matrix way of solving for interpolation and one of the application of interpolation where we can estimate an asteroids path given a set of points thatit has followed in the past.

**Interpolation of Polynomial**

Interpolation is a technique where we find the minimum ordered polynomial of order n that passes through a set of arbitrary n + 1 points. It is important to know that there will always be a unique polynomial of order n that passes through n + 1 arbitrary points.Δ

* One of the approaches to find the polynomial given the set of points is a vandremonde matrix. Lets take an example of a known polynomial, derive a set of points and use those to find the polynomial back. Lets take y = 3x + 4. This polynomial is simple enough, but instead of taking 2 points lets take 5 set of points and assume we might get a polynomial of order 4

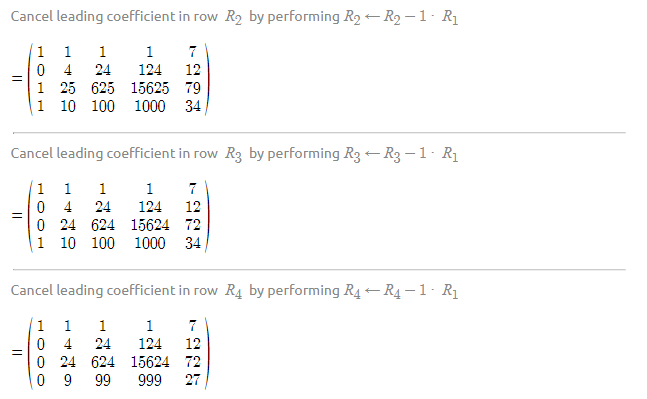
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | 1 | 5 | 25 | 10 |
| f(x) | 7 | 19 | 79 | 34 |

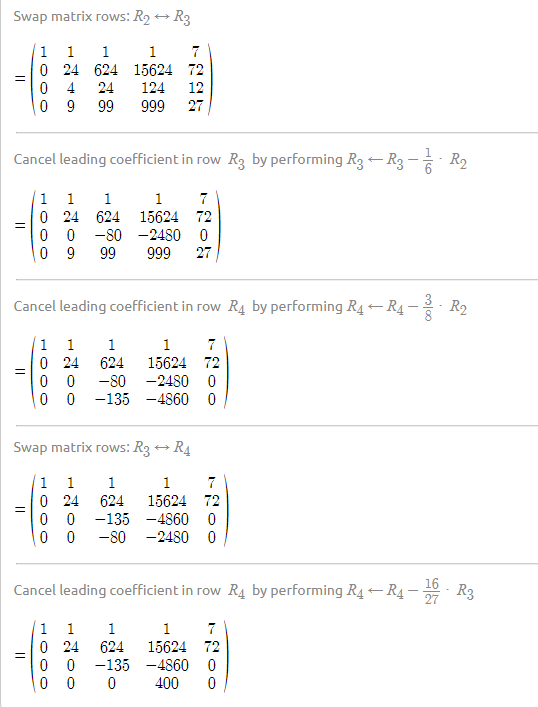
**Construction of Vandremonde Matrix**

We have already a fair idea as to how many points are present in the dataset. Since we have 4 data points there must be a polynomial of degree 4 satisfying this above points (Since we already know that the min order polynomial is order 1, we will observe how vandremonde method finally gets us to our original equation)

We have a system of equations now with four unknowns and 4 equations. The Augmented Matrix is given by

Solving the Matrix





Solving the equation with reduced matrix we get

Using vandremondes technique for polynomials of low degree is fine. But it becomes computationally hard as the order increases and here we will slightly touch base on other methods of interpolating polynomials

**Lagrange interpolation**

Define the interpolating polynomials on the interpolation

In other words, interpolates the special data with

equal to 1 and , Then the interpolating polynomial is simply represented as

It is easy to see that the special polynomials are linearly independent if are distinct, and hence form a basis for . Note that each and every of the basis polynomials is of degree N − 1.

The basis polynomials can be easily constructed as follows.

In General

The above general idea briefly describes the Lagrange’s method of solving for the polynomial that passes through given points.

**Coefficient determination with a fixed basis**

Instead of building the basis functions on every and each set of interpolation nodes, we let be a fixed basis of .Then the interpolating polynomial p can be represented as

The interpolation condition gives the interpolating equations for the combination coefficients

Note that if , then are the Lagrange polynomials at the nodes , and the matrix is the identity. When the set of interpolation nodes is changed, the matrix is changed as well. We find the coefficients by numerically solving the system of linear

equations. When the basis is , the matrix is known as Vandermonde matrix. It is nonsingular as long as the nodes are distinct. In other words, the interpolating coefficients can be determined for any set of data as long as xi are distinct. Once the coefficients are obtained, the evaluation at any point x can be done with operations.

**Applications of Interpolation of Polynomials and Vandermonde Matrix**

DFT Using Vandremonde Matrix of interpolation

In applied mathematics, a DFT matrix is an expression of a discrete Fourier transform (DFT) as a transformation matrix, which can be applied to a signal through matrix multiplication.

where is a primitive Nth root of unity in which . We can avoid writing large exponents for using the fact that for any exponent we have the identity . This is the Vandermonde matrix for the roots of unity, up to the normalization factor. Note that the normalization factor in front of the sum and the sign of the exponent in ω are merely conventions, and differ in some treatments. All of the following discussion applies regardless of the convention, with at most minor adjustments. The only important thing is that the forward and inverse transforms have opposite-sign exponents, and that the product of their normalization factors be 1/N. However, the choice here makes the resulting DFT matrix unitary, which is convenient in many circumstances.

Fast Fourier transform algorithms utilize the symmetries of the matrix to reduce the time of multiplying a vector by this matrix, from the usual O()

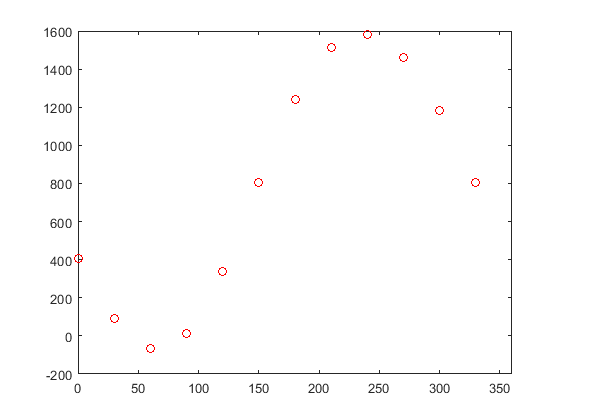
**Calculation of orbits of asteroids (Fourier and fast Fourier transforms)**

Usage of the fast Fourier transform (FFT) to estimate the coefficients of a trigonometric polynomial that interpolates a set of data is quite common. Many modules in Matlab and Simulink uses fast fourier transforms for plotting graphs and smoothing out surfaces.

Keeping several vast applications of fourier transformations in signals and systems aside, we will now look at a cool application where some set of points for an asteroid are taken into consideration and are interpolated to obtain a polynomial to estimate the trajectory. (Keep in mind that unlike planets asteroids are wild and are prone to significant change in path orbits even with a slight change in gravity).

The FFT algorithm is associated with applications in signal processing, but it can also be used more generally as a fast computational tool in mathematics. For example, coefficients ci of an nth degree polynomial  that interpolates a set of data are commonly computed by solving a straightforward system of linear equations. While studying asteroid orbits in the early 19th century, Carl Friedrich Gauss discovered a mathematical shortcut for computing the coefficients of a polynomial interpolant by splitting the problem up into smaller subproblems and combining the results. His method was equivalent to estimating the discrete Fourier transform of his data.

In a paper by Gauss, he describes an approach to estimating the orbit of the Pallas asteroid. He starts with the following twelve 2-D positional data points x and y.



**Path of Asteroid “PALLAS” observed and is plotted to scale**

The Gauss model of the asteroid is of the following form

Observe that we just need to find the co-efficients of the equations to get the complete polynomial which satisfies the equation.   
  
**Where does LA come here**

We may be going back and forth naming several methods like Vandermonde matrix, fast fourier transforms but in reality we are just using simple matrix eliminations to build DFT and using optimizations on DFT we are achieving fast fourier transforms in O(n log n) time complexity to solve for equations. The most crucial step which is finding the curve that fits the path is entirely dependent on the Linear algebra which helps us solve for the unknowns.

In Matlab we can find the fast fourier transform using fft()function.

**Matlab Code to simulate the orbit of asteroid**

x = 0:30:330;

y = [408 89 -66 10 338 807 1238 1511 1583 1462 1183 804];

plot(x,y,'ro')

xlim([0 360])

m = length(y);

n = floor((m+1)/2);

z = fft(y)/m;

a0 = z(1);

an = 2\*real(z(2:n));

a6 = z(n+1);

bn = -2\*imag(z(2:n));

px = 0:0.01:360;

k = 1:length(an);

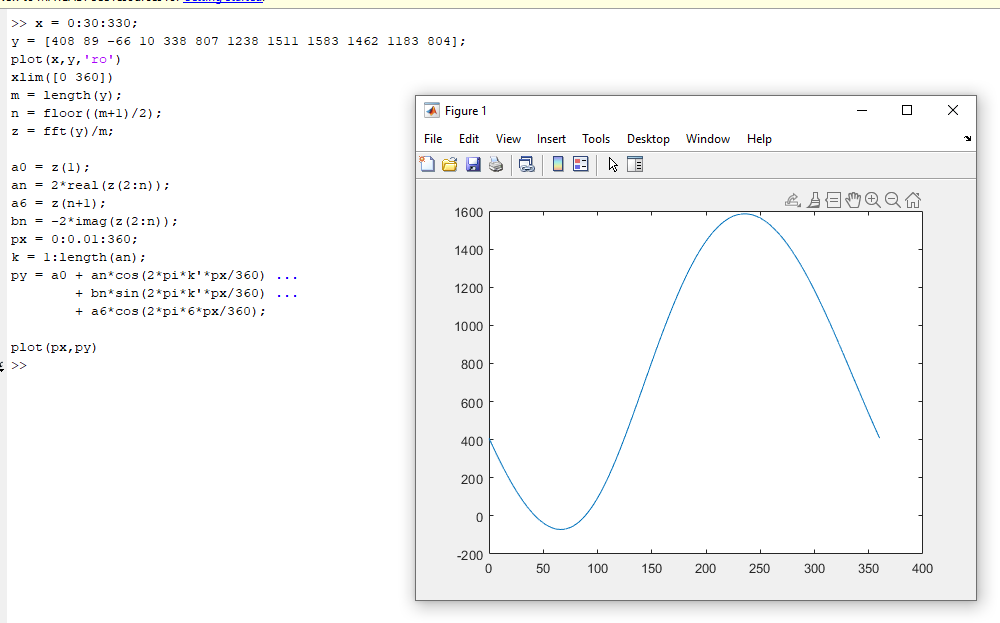
py = a0 + an\*cos(2\*pi\*k'\*px/360) ...

+ bn\*sin(2\*pi\*k'\*px/360) ...

+ a6\*cos(2\*pi\*6\*px/360);

plot(px,py)

**Matlab Output**



References

[1] Briggs, W. and V.E. Henson. The DFT: An Owner's Manual for the Discrete Fourier Transform. Philadelphia: SIAM, 1995.

[2] Gauss, C. F. “Theoria interpolationis methodo nova tractata.” Carl Friedrich Gauss Werke. Band 3. Göttingen: Königlichen Gesellschaft der Wissenschaften, 1866.

[3] Heideman M., D. Johnson, and C. Burrus. “Gauss and the History of the Fast Fourier Transform.” Arch. Hist. Exact Sciences. Vol. 34. 1985, pp. 265–277.

[4] Goldstine, H. H. A History of Numerical Analysis from the 16th through the 19th Century. Berlin: Springer-Verlag, 1977.