Polynomial Method for Non Causal Constant Coefficient Difference Equations

Project under Prof. K S Venkatesh

December 2015

1 General Form of the Problem

We have a recurrence relation of the form

$$\Sigma_{i=n-N+1}^{n}\alpha_i x[n-i] = \Sigma_{i=n-N+1}^{n}\beta_i y[n-i] \tag{1}$$

where x[n] are the "known" inputs and y[n] are the response of the system that are to be known.
Also, there are η intial conditions of the form

$$\Sigma_{i=0}^{N}\gamma_{i,k}x[i] = \Sigma_{i=0}^{N}\delta_{i,k}y[i] + \sigma_i \tag{2}$$

η will be the difference between the coefficient of the maximum and the minimum non zero β_i.
Here,all α , β , γ , δ and σ belong in $\mathbf{R}$. Now we will construct a polynomial that when sampled at the discrete points will yield x[n] and y[n].

2 Constructing The Polynomial

We will use a popular numerical method tehnique to construct a N-1 degree polynomial.

2.1 Newton Divided Difference Method

Define a new notation of the form :

$$f[x_0, x_1..., x_k] = \frac{f[x_1, x_2..., x_{k-1}] - f[x_0, x_1, ..., x_k]}{k}$$

with the base case being :

$$f[x_1, x_0] = x[1] - x[0]$$

Then the required N-1 degree polynomial is:

$$p[n] = x[0] + (n - x[0])f[x_1, x_0] + (n - x[0])(n - x[1])f[x_2x_1x_0] + ... \tag{3}$$
$$+ (n - x[0])(n - x[1])(n - x[2]).....(n - x[N - 2])f[x_{N-1}, x_{N-2}, ..., x_0] \tag{4}$$

Also assume a N-1 degree polynomial for y[n] which when sampled at the points 0,1,2,3... will give us the output values.

$$Y(n) = a_{N-1}n^{N-1} + a_{N-2}n^{N-2}........... + a_1n + a_0$$

3 Finding the Coefficients of Y

3.1 Method of undetermined coefficients

N equations are needed to solve for the N unknown coefficients present in Y.We obtain those as follows.
Put the assumed polynomial Y in the RHS part of (1) i.e. in $\Sigma_{i=n-N+1}^{n}\beta_i y[n-i]$ and obtain an expression of form

$$\omega_{N-1}n^{N-1}..... + \omega_1 n + \omega_0 \tag{5}$$

where all are ω linear combination of $a_i's$ as per the binomial coefficients produced from y(n-i).
Similarly, obtain a polynomial from the LHS of (1) by putting X in $\Sigma_{i=n-N+1}^{n}\alpha_i x[n-i]$ as a polynomial and collecting the like powers of n.

$$\Omega_{N-1}n^{N-1}..... + \Omega_1 n + \Omega_0 \tag{6}$$

where all are Ω linear combination of $a_i's$ as per the binomial coefficients produced from y(n-i).
Now compare the coefficients of like powers of n on both sides to get N equations $\omega_i = \Omega_i$ i varies from 0 to N-1 giving us N linear equation in a_i which can be solved.
Now since are a_i are known the polynomial Y just needs to be sampled at discrete points to get y[n].

4 Another Approach

The above method can be optimized by bypassing the step to find a polynomial for X using the Newton Divided Difference approach described in Section 2.1

4.1 A more Direct Method

Assume a polynomial for Y of the form

$$Y(n) = a_{N-1}n^{N-1} + a_{N-2}n^{N-2}............ + a_1 n + a_0$$

and proceed as in Section 3.1 to obtain

$$\omega_{N-1}n^{N-1}..... + \omega_1 n + \omega_0$$

where all ω_i are functions of coefficients i.e. $\omega_i = f(a_0, a_1...., a_{N-1})$ But now, instead of proceeding to do the same for X form N different equation from the initial conditions and the recurrence relation and treat the value of X[n] in each of them as constants to obtain N different linear equations in a_i.
Now solve this system of equations to obtain a_i i varying from 0 to N-1 and sample at instants 0,1,2,3...N-1 to obtain the value of y.

5 Concluding Remarks

Both Methods - Matrix and Polynomial Method solve the problem with same accuracy and running time.However, Matrix method is much easier to implement and understand.Polynomial Method uses the analog world as a crutch to draw parallels and solve the discrete problem but it provides more insight into the nature of problem and allows us to predict the intermediate values between instants,if need be.
And more importantly,I want to thank my dearest mentor Prof. K S Venkatesh to allow me an opportunity to learn in depth about the subject and to bestow his guidance over me.

Matrix Method to solve constant coefficient Difference Equations

Winter Project Under Prof K S Venkatesh

December 2015

1 General Form Of The Problem

We have a recurrence relation of the form

$$\Sigma^{n}_{i=n-N+1}\alpha_i x[n-i] = \Sigma^{n}_{i=n-N+1}\beta_i y[n-i] \tag{1}$$

where x[n] are the "known" inputs and y[n] are the response of the system that are to be known.
Also, there are η intial conditions of the form

$$\Sigma^{N}_{i=0}\gamma_{i,k}x[i] = \Sigma^{N}_{i=0}\delta_{i,k}y[i] + \sigma_i \tag{2}$$

η will be the difference between the coefficient of the maximum and the minimum non zero β_i.
Here,all α , β , γ , δ and σ belong in $\mathbf{R}$.However, in further investigation σ_i are considered because they imply changing the input and the output by a constant and can be adjusted for later and would require just an extra row and an extra column in matrices.

1.1 An Example:

We have the recurrence relation

$$x[n] + 2x[n-1] = y[n] + y[n+1] + y[n+2] \tag{3}$$

say, we have N = 5. Hence the above relation is valid with n $\in$ [0,3]. So to have a unique solution we must have two intial conditions i.e. $\eta = 2$.
Let the intial conditions be

$$\begin{aligned} x[0] + x[1] &= y[2] - y[0] \\ x[5] &= y[2] + y[3] + 2y[5] \end{aligned}$$

2 Construction Of Matrices

Now treating the equations as Linear Equations in n variables we will construct two Coefficient matrix A and B.

2.1 Constructing the Input Matrix - A

$$A_{NxN} = \begin{pmatrix} \alpha_1 & \alpha_2 & ..Ncol... & \alpha_n \\ \alpha_1 & \alpha_2 & ..Ncol... & \alpha_n \\ \alpha_1 & \alpha_2 & ..Ncol... & \alpha_n \\ ... & ... & ...... & .... \\ ... & After & N-\eta+1 & rows \\ \alpha_1 & \alpha_2 & ..Ncol... & \alpha_n \\ \gamma_{1,1} & \gamma_{2,1} & ..Ncol... & \gamma_{N,1} \\ \gamma_{1,2} & \gamma_{2,2} & ..Ncol... & \gamma_{N,2} \\ ... & ... & ...... & .... \\ ... & After & \eta-1 & rows \\ \gamma_{1,N} & \gamma_{2,N} & ..Ncol... & \gamma_{N,N} \end{pmatrix}$$

So, A is constructed as described above from coefficients of the linear recursion and the initial conditions.

2.2 Constructing the Output Matrix - B

$$B_{NxN} = \begin{pmatrix} \beta_1 & \beta_2 & ..Ncol... & \beta_n \\ \beta_1 & \beta_2 & ..Ncol... & \beta_n \\ \beta_1 & \beta_2 & ..Ncol... & \beta_n \\ ... & ... & ...... & .... \\ ... & After & N-\eta+1 & rows \\ \beta_1 & \beta_2 & ..Ncol... & \beta_n \\ \delta_{1,1} & \delta_{2,1} & ..Ncol... & \delta_{N,1} \\ \delta_{1,2} & \delta_{2,2} & ..Ncol... & \delta_{N,2} \\ ... & ... & ...... & .... \\ ... & After & \eta-1 & rows \\ \delta_{1,N} & \delta_{2,N} & ..Ncol... & \delta_{N,N} \end{pmatrix}$$

The Output Matrix of the coefficients is constructed in a similar way.

3 Linear Algebra

3.1 Representation in Matrices

Construct two N length column matrices $X = [x[0], x[1], x[2]...x[N-1]]^T$ and $Y = [y[0], y[1], y[2]...y[N-1]]^T$. Now the complete problem to boils down to finding Y given X, A and B. The complete information can be represented in the matrix equation:

$$AX = BY$$

3.2 Final Solution:

For the system to have a unique solution A must me invertible,So we have the final solution y[n] at each discrete moment in time as:

$$Y = B^{-1}AX$$

3.3 Final Remarks

The cost of computation is $O(N^{\log_2^7})$ in the worst case.However, It can be significantly reduced on observing that A and B would be mostly sparse and utilizing this to select a more optimized algorithm for finding the inverse of B.
Further, the invertibility would be further investigated as to what other possibilities emerge for different possible ranks of B.
Thank you sir for agreeing to mentor me for the same.