1.2) Applying Taylor's series to $F(x+\beta_i \Delta x)$

$$F(x+\beta_i \Delta x) = F(x) + (\beta_i \Delta x) F'(x) + \frac{(\beta_i \Delta x)^2 F''(x)}{2!} + \cdots$$

substituting into the approximation we have,

$$F'(x) \simeq \frac{1}{\Delta x} \sum_{i=1}^{N} \alpha_i F(x+\beta_i \Delta x) = \sum_{i=1}^{N} \Big(\frac{\alpha_i F(x)}{\Delta(x)} + \cancel{\alpha_i + \beta_i F}$$

$$\alpha_i \beta_i F'(x) + \frac{\alpha_i \beta_i^2 \Delta x F''(x)}{2!} + \cdots$$

~~since we are only interested in~~

since we are only interested in $F'(x)$
setting it up as a system of equations in matrix form

$$\begin{pmatrix} \frac{\alpha_1}{\Delta x} & \frac{\alpha_2}{\Delta x} & \cdots & \frac{\alpha_{N_p}}{\Delta x} \\ \alpha_1 \beta_1 & \alpha_2 \beta_2 & \cdots & \alpha_{N_p} \cdot \beta_{N_p} \\ \alpha_1 \beta_1^2 \frac{\Delta x}{2!} & \alpha_2 \beta_2^2 \frac{\Delta x}{2!} & \cdots & \alpha_{N_p} \cdot \beta_{N_p}^2 \frac{\Delta x}{2!} \\ \vdots & \vdots & & \vdots \\ \alpha_1 \beta_1^p \frac{\Delta x^{p-1}}{(p-1)!} & \alpha_2 \beta_2^p \cdot \frac{\Delta x^{p-1}}{(p-1)!} & \cdots & \alpha_p \beta_{N_p}^p \frac{\Delta x^{p-1}}{(p-1)!} \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Each colum corresponds to a value of i
and each row represents each derivative each derivative
i.e 2nd row is the first derivatie, 2nd row is the 3rd etc

Subject:.. No:................ Date:

the matrix can be simplified further, since δx is small and everything sums to 0, furthermore notice there are no δx values for the ~~first deriv~~ $F'(x)$

Then, we are left with only α_i and β_i values within the matrix. which can be split in the form of $Ax = b$.

A is the matrix of all β values

x is the alpha values

b is the vector on the right hand side.

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ \beta_1 & \beta_2 & \cdots & \beta_{N_p} \\ \beta_1^2 & \beta_2^2 & \cdots & (\beta_{N_p})^2 \\ \vdots & \vdots & & \vdots \\ \beta_1^p & \beta_2^p & \cdots & \beta_{N_p}^p \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_{N_p} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

using the code np.linalg.solve(A, b) will give the alpha values.

we find the square matrix with $p+1$ rows and N_p columns. $\Rightarrow$ $N_p = p+1$
where N_p is the min # of points required to find an approximation which is p'th order accurate.