Laboratorium **Metod Obliczeniowych** Wydział Elektrotechniki Automatyki i Informatyki Politechnika Świętokrzyska	
Studia: **Stacjonarne I stopnia**	Kierunek: **Informatyka**
Data wykonania: **06.11.2017**	Grupa: **3ID13B**
Imię i nazwisko: **Bartłomiej Osak**	
Numer ćwiczenia: **4**	Temat ćwiczenia: **Interpolacja**

1. Interpolacja wielomianem Lagrange – program w środowisku MATLAB.

Polecenie: Wyznacz funkcje interpolacji $y = f(x)$ za pomocą wielomianu Lagrange'a.

x_i	1.41	1.46	1.52	1.60	1.65	1.72
y_i	3.57418	3.32513	3.09336	2.86203	2.74926	2.62098

Kod źródłowy:

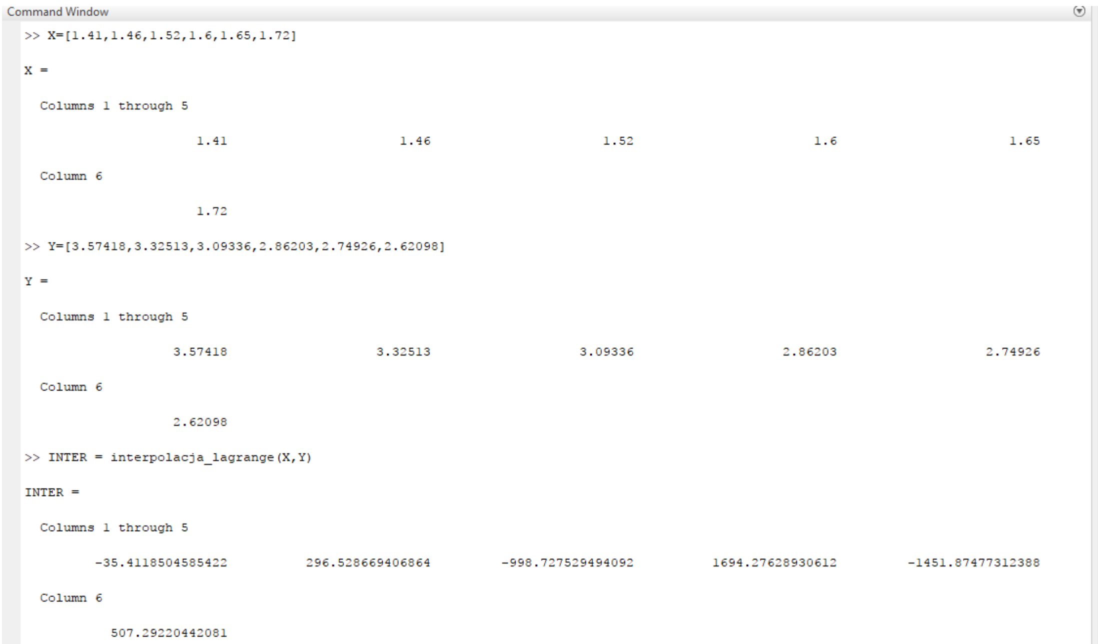
```
function[C] = interpolacja_lagrange(X,Y)
format long g
    C = 0;
    for i=1:size(X,2)
        P = 1;
        for j=1:size(X,2)
            if j~=i
                P = conv(P,[1 -X(j)])/(X(i)-X(j));
            end
        end
        C = C + Y(i) * P;
    end
end
```

Wywołanie:

- Przypisanie wartości do wektorów X oraz Y:
  X=[1.41,1.46,1.52,1.6,1.65,1.72]
  Y=[3.57418,3.32513,3.09336,2.86203,2.74926,2.62098]
- Wywołanie funkcji interpolacja_lagrange oraz przypisanie jej wyników do wektora INTER:
  INTER = interpolacja_lagrange(X,Y)
- Wynik zadziałania:

```
ans =
Columns 1 through 5
          -35.4118504585422          296.528669406864          -998.727529494092
          1694.27628930612          -1451.87477312388
Column 6
          507.29220442081
```

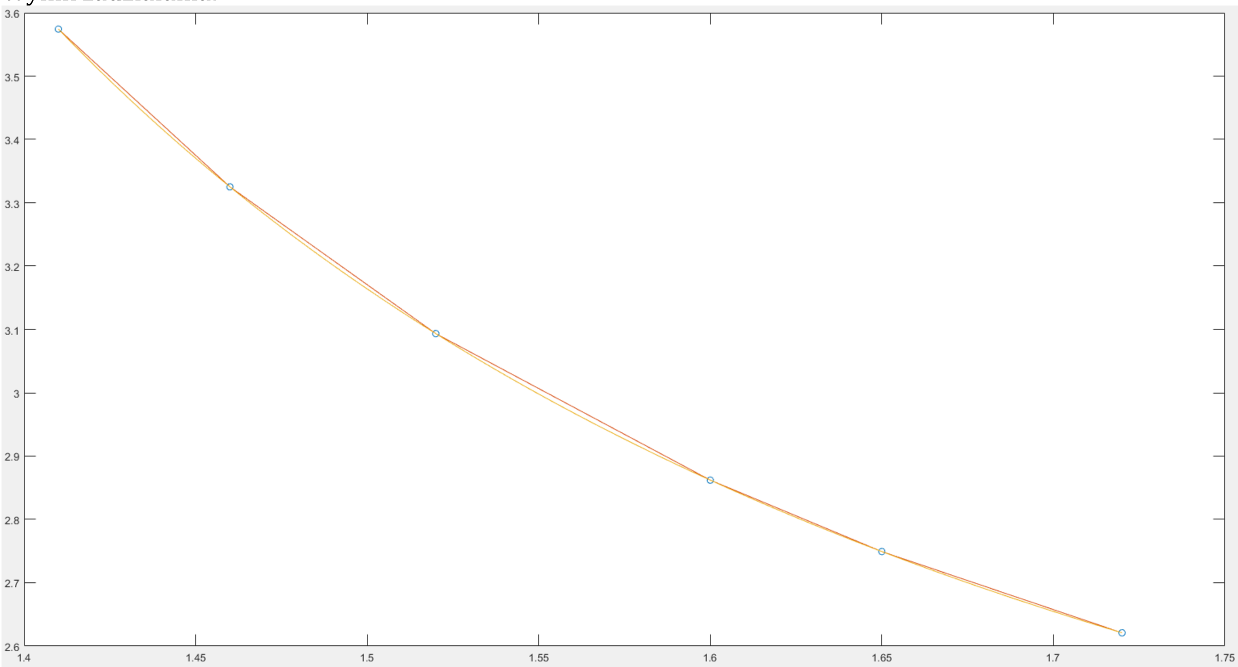
Zrzut ekranu:



```
Command Window
>> X=[1.41,1.46,1.52,1.6,1.65,1.72]

X =

  Columns 1 through 5

             1.41             1.46             1.52             1.6             1.65

  Column 6

             1.72

>> Y=[3.57418,3.32513,3.09336,2.86203,2.74926,2.62098]

Y =

  Columns 1 through 5

          3.57418          3.32513          3.09336          2.86203          2.74926

  Column 6

          2.62098

>> INTER = interpolacja_lagrange(X,Y)

INTER =

  Columns 1 through 5

       -35.4118504585422       296.528669406864       -998.727529494092       1694.27628930612       -1451.87477312388

  Column 6

       507.29220442081
```

Generowanie wykresu celem potwierdzenia interpolacji:

- Zapis do wektora XX sekwencji wygenerowanych wartości w zakresie od $< x_1, x_n >$:
  XX=1.41:0.001:1.72
- Zapis do wektora YY wartości wielomianu stworzonego na podstawie współczynników otrzymanych z wywołania funkcji interpolacja_lagrange (i zapisanych do zmiennej INTER) dla wartości z wektora XX:
  YY=(INTER(1)*XX.^5)+(INTER(2)*XX.^4)+(INTER(3)*XX.^3)+(INTER(4)*XX.^2)+(INTER(5)*XX) +INTER(6)
- Wygenerowanie wykresu:
  plot(X,Y,'o',X,Y,XX,YY)
- Wynik zadziałania:



Opis funkcji:

Funkcja interpolacja_lagrange przyjmuje dwa argumenty:

- X – wektor współrzędnych x danych punktów
- Y – wektor współrzędnych y danych punktów

Funkcja zwraca wektor o nazwie C. Początkowo inicjujemy go wartością równą zeru. Następnie inicjujemy pierwszą pętlę która iteruje od 1 do rozmiaru wektora X. W pętli tej inicjujemy wektor pomocniczy P wartością 1. Następnie inicjujemy drugą pętlę iterującą również od 1 do rozmiaru wektora X. W pętli tej występuje instrukcja warunkowa sprawdzająca, czy iterator pętli wewnętrznej jest różny od wartości iteratora pętli zewnętrznej. Jeśli warunek jest spełniony następuje przypisanie do wektora pomocniczego P wartości ilorazu splotu wektorów P oraz [1 -X(j)] i różnicy X(i)-X(j). Następnie w pętli zewnętrznej następuje zapis do wektora wynikowego C sumy poprzedniej wartości wektora wynikowego i wartości wektora Y(i), która jest pomnożona przez wartość wektora pomocniczego P. Całość algorytmu oparta jest zgodnie ze wzorem na interpolację Lagrange'a:

$$W_n(x) = \sum_{j=0}^{n} y_j * \frac{(x - x_0) * (x - x_1) *..* \left(x - x_{j-1}\right) * \left(x - x_{j+1}\right) *..* (x - x_n)}{\left(x_j - x_0\right) * \left(x_j - x_1\right) *..* \left(x_j - x_{j-1}\right) * \left(x_j - x_{j+1}\right) *..* \left(x_j - x_n\right)}$$

2. Interpolacja wielomianem Lagrange – program w języku R.

Polecenie: Wyznacz funkcje interpolacji $y = f(x)$ za pomocą wielomianu Lagrange'a.

x_i	1.41	1.46	1.52	1.60	1.65	1.72
y_i	3.57418	3.32513	3.09336	2.86203	2.74926	2.62098

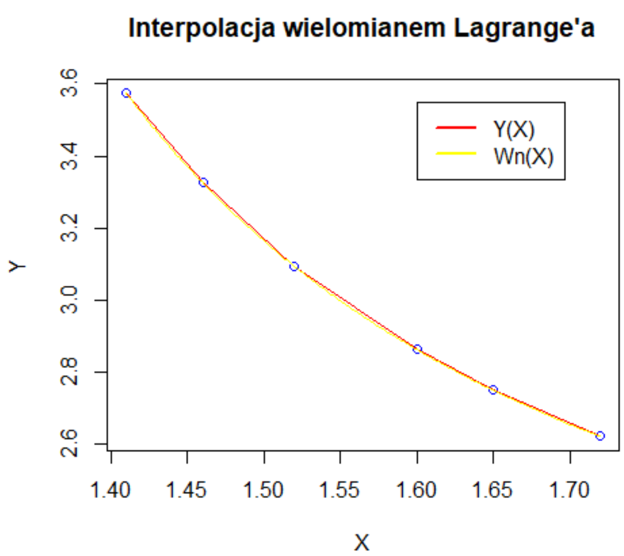
Kod źródłowy – wersja 1 (z domyślną funkcją conv()):

```
function.lagrange <- function(X,Y)
{
  C <- c(0);
  for(i in 1:length(X))
  {
    P <- c(1)
    for(j in 1:length(X))
    {
      if (j!=i)
      {
        W <- c(1,-X[j])
        P <- c(conv(P,W))/(X[i]-X[j])
      }
    }
    C <- c(C + Y[i] * P)
  }
  return (C);
}
```

Wywołanie:

- Przypisanie wartości do wektorów X oraz Y:
  X<-c(1.41,1.46,1.52,1.6,1.65,1.72)
  Y<-c(3.57418,3.32513,3.09336,2.86203,2.74926,2.62098)
- Wywołanie funkcji interpolacja_lagrange oraz przypisanie jej wyników do wektora INTER:
  INTER<-c(function.lagrange(X,Y))
- Zapis do wektora XX sekwencji wygenerowanych wartości w zakresie od $< x_1, x_n >$:
  XX<-seq(1.41,1.72,0.001)
- Zapis do wektora YY wartości:
  YY<-(INTER[1]*XX^5)+(INTER[2]*XX^4)+(INTER[3]*XX^3)+(INTER[4]*XX^2)+(INTER[5]*XX) +INTER[6]
- Wygenerowanie wykresu:
  plot(X,Y,col='blue')
  lines(X,Y,col='red')
  lines(XX,YY,col='yellow')
  title(„Interpolacja wielomianem Lagrange'a")
  legend(1.6,3.55,c('Y(X)','Wn(X)'),
  +lwd=c(2,2),col=c('red','yellow'))
- Wynik zadziałania:

```
Console ~/
> X<-c(1.41,1.46,1.52,1.6,1.65,1.72)
> Y<-c(3.57418,3.32513,3.09336,2.86203,2.74926,2.62098)
> INTER<-c(function.lagrange(X,Y))
> XX<-seq(1.41,1.72,0.001)
> YY<-(INTER[1]*XX^5)+(INTER[2]*XX^4)+(INTER[3]*XX^3)+(INTER[4]*XX^2)+(INTER[5]*XX)
> +INTER[6]
[1] 507.2922
>
> plot(X,Y,col='blue')

> plot(X,Y,col='blue')
> lines(X,Y,col='red')
> lines(XX,YY,col='yellow')
> title("Interpolacja wielomianem Lagrange'a")
> legend(1.6,3.55,c('Y(x)','Wn(x)')),

> legend(1.6,3.55,c("Y(x)","Wn(x)")),

> legend(1.6,3.55,c("Y(x)","Wn(x)"),
+ lwd=c(2,2),col=c('red','yellow'))
> |
```



Kod źródłowy – wersja 2:

Można również napisać samodzielnie metodę conv() wykonującą splot macierzy, a następnie wywołać ją w funkcji głównej interpolacyjnej:

- **Funkcja function.conv():**

```
function.conv <- function(U,V)
{
  m<-length(U)
  n<-length(V)
  t<-c()
  for(k in 1:m+n)
  {
    t=0.0
    for(j in 0:m)
    {
      if((k-j)>=1 && (k-j)<=n)
      {
        t=c(t)+(U[j+1]*V[k-j])
        print(t)
      }
    }
    w=c(w)+c(t);
  }
  return (w)
}
```

- **Funkcja function.lagrange z wywołaniem function.conv:**

```
function.lagrange <- function(X,Y)
{
  C <- c(0);
  for(i in 1:length(X))
  {
    P <- c(1)
    for(j in 1:length(X))
    {
      if (j!=i)
      {
        W <- c(1,-X[j])
        P <- c(eval(function.conv(P,W))/(X[i]-X[j])
      }
    }
    C <- c(C + Y[i] * P)
  }
  return (C);
}
```

3. Interpolacja wielomianem Newtona – program w środowisku MATLAB.

Kod źródłowy:

```
function[C] = interpolacja_newton(X,Y)
    n = length(X);
    D = zeros(n,n);
    D(:,1) = Y;
    for i=2:size(X,2)
        for j=i:size(X,2)
            D(j,i) = (D(j,i-1)-D(j-1,i-1))/(X(j)-X(j-i+1));
        end
    end
    C = D(n,n);
    for k=(n-1):-1:1
        C = conv(C,poly(X(k)));
        m = length(C);
        C(m) = C(m) + D(k,k);
    end
end
```

Wywołanie:

- Przypisanie wartości do wektorów X oraz Y:

  ```
  X=[1.41,1.46,1.52,1.6,1.65,1.72]
  Y=[3.57418,3.32513,3.09336,2.86203,2.74926,2.62098]
  ```

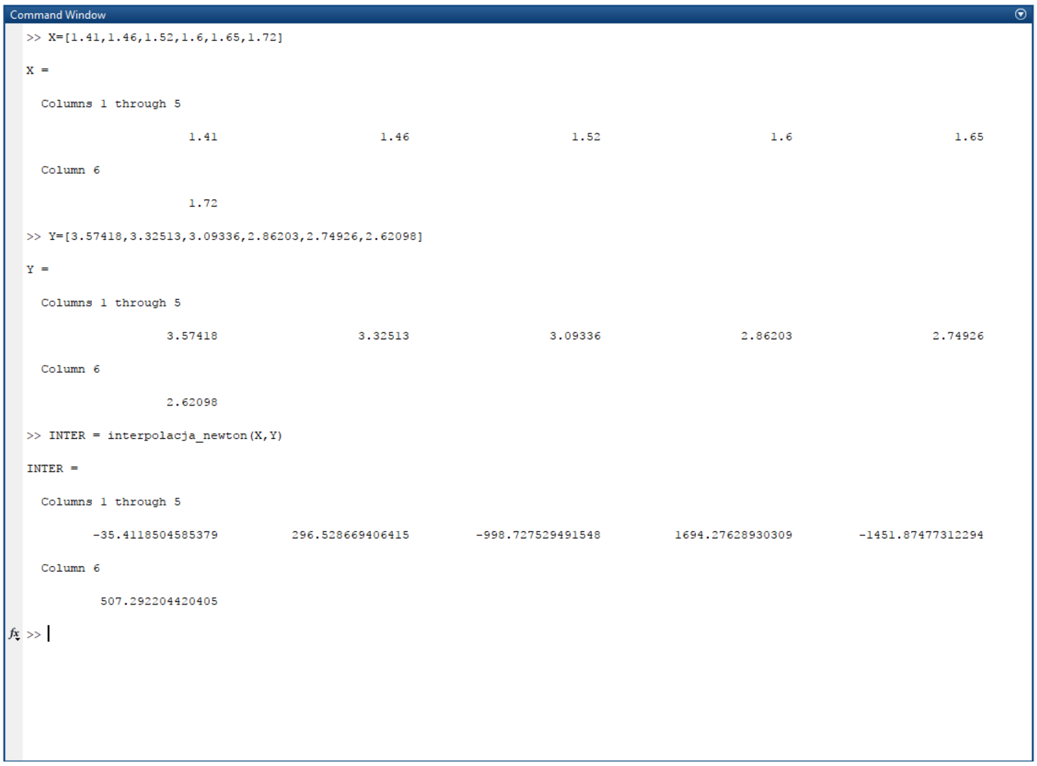
- Wywołanie funkcji interpolacja_lagrange oraz przypisanie jej wyników do wektora INTER:

  ```
  INTER = interpolacja_newtona(X,Y)
  ```

- Wynik zadziałania:

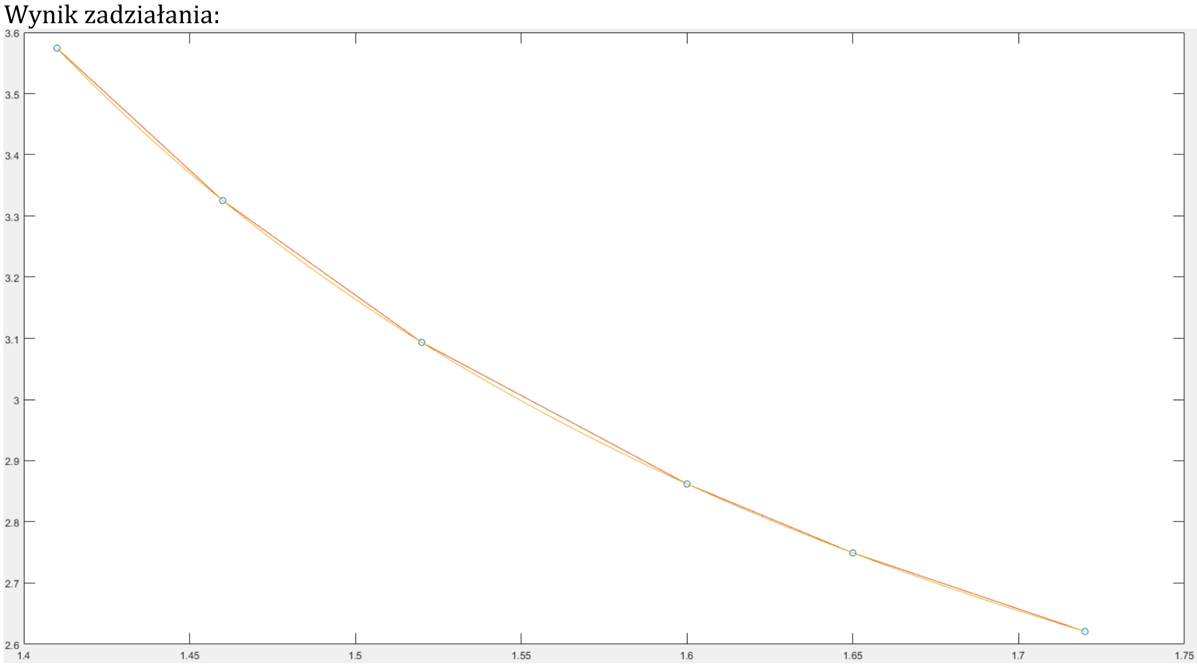
  ```
  ans =
  Columns 1 through 5
           -35.4118504585422          296.528669406864          -998.727529491548
           1694.27628930309          -1451.87477312294
  Column 6
           507.292204420405
  ```

Zrzut ekranu:



Generowanie wykresu celem potwierdzenia interpolacji:

- Zapis do wektora XX sekwencji wygenerowanych wartości w zakresie od $< x_1, x_n >$:
  `XX=1.41:0.001:1.72`
- Zapis do wektora YY wartości wielomianu stworzonego na podstawie współczynników otrzymanych z wywołania funkcji interpolacja_newtona (i zapisanych do zmiennej INTER) dla wartości z wektora XX:
  `YY=(INTER(1)*XX.^5)+(INTER(2)*XX.^4)+(INTER(3)*XX.^3)+(INTER(4)*XX.^2)+(INTER(5)*XX)+INTER(6)`
- Wygenerowanie wykresu:
  `plot(X,Y,'o',X,Y,XX,YY)`
- Wynik zadziałania:



Opis funkcji:

Funkcja interpolacja_newton przyjmuje dwa argumenty:

- X – wektor współrzędnych x danych punktów
- Y – wektor współrzędnych y danych punktów

Funkcja zwraca wektor o nazwie C. Na początku do zmiennej n przypisujemy rozmiar wektora X oraz inicjujemy wektor D, który jest zerowany przez funkcję zeros. Następnie do wyzerowanego wektora D przypisujemy transponowany wektor Y. Dalej inicjujemy pętlę, która iteruje od 1 do rozmiaru wektora X. W pętli zewnętrznej inicjujemy pętlę wewnętrzną iterującą od i, czyli od iteratora pętli zewnętrznej do rozmiaru wektora X. W pętli wewnętrznej obliczamy tablicę różnic dzielonych. Po wyjściu z obu pętli następuje pobranie wartości z tabeli różnic dzielonych, czyli z macierzy D. Jest to ostatnia wartość różna od 0 (znajduje się na poziomie n x n). Następnie definiujemy pętlę iterującą od rozmiaru wektora X pomniejszonego o 1 do wartości 1 co skok wynoszący -1. W pętli do wektora wynikowego C przypisywana jest wartość zwracana poprzez wykonanie funkcji splotu macierzy. Splot macierzy występuje dla poprzedniej wartości wektora wynikowego C oraz dla rezultatu funkcji poly wywoływanej dla k-tego elementu (iterator k) wektora X, która zwraca współczynniki wielomianu naturalnego. Na końcu w pętli wykonywane jest przypisanie do zmiennej m aktualnej długości wektora wynikowego C oraz przypisanie na pozycję m-tą wektora wynikowego C sumy poprzedniej wartości spod m-tej pozycji oraz wartości z macierzy różnic dzielonych o współrzędnych k,k.

4. Interpolacja trygonometryczna.

W wielu przypadkach konieczna jest interpolacja funkcji okresowej $y = f(t)$. O funkcji takiej zakłada się, że jest okresowa o okresie głównym $[0; 2\pi]$ lub w przypadku ogólniejszym: $[a; b]$. Ponieważ jako bazę funkcji stosuje się funkcje trygonometryczne postaci:

$$1, \cos(x), \sin(x), \cos(2x), \sin(2x), \ldots, \cos(mx), \sin(mx)$$

których okresem głównym jest przedział $[0; 2\pi]$ (lub jego całkowite „wielokrotności”) w przypadku okresu głównego $[a; b]$ konieczne jest przeskalowanie przedziału za pomocą zależności:

$$t = a + \frac{b-a}{2\pi}x, gdzie\ (a \leq t \leq b\ wtedy\ i\ tylko\ wtedy, gdy\ 0 \leq x \leq 2\pi)$$

Zatem funkcja interpolująca jest wielomianem trygonometrycznym o postaci:

$$F(x) = \frac{a_0}{2} + \sum_{k=1}^{m} (a_k \cos(kx) + b_k \sin(kx))$$

Przy czym $n = 2m$ (liczba węzłów interpolacji jest równa n+1!). Współczynniki $a_0, a_1, a_2, .., a_m, b_m$ wyznacza się na podstawie rozwiązania układu równań:

$$\begin{cases} F(x_0) = a_0 + a_1 \cos(x_0) + b_1 \sin(x_0) +..+a_m \cos(mx_0) + b_m \sin(mx_0) = y_0, \\ F(x_1) = a_0 + a_1 \cos(x_1) + b_1 \sin(x_1) +..+a_m \cos(mx_1) + b_m \sin(mx_1) = y_1, \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots.. \\ F(x_{2m}) = a_0 + a_1 \cos(x_{2m}) + b_1 \sin(x_{2m}) +..+a_m \cos(mx_{2m}) + b_m \sin(mx_{2m}) = y_{2m} \end{cases}$$

W praktyce najważniejszy jest przypadek, gdy węzły interpolacji są równoodległe tj.

$$x_i = \frac{2i}{2m+1}\pi, dla\ i = 0,1,..,2m$$

W tym przypadku macierz główna układu równań pomnożona przez $\sqrt{\frac{2}{2m} + 1}$ staje się macierzą ortogonalną. Dzięki temu można podać bezpośrednie wzory na obliczanie współczynników $a_0, a_1, b_1, .., a_m, b_m$.

$$a_0 = \frac{2}{2m+1} * \sum_{i=0}^{2m} y_i$$

$$a_k = \frac{2}{2m+1} * \sum_{i=0}^{2m} y_i coskx_i$$

$$b_k = \frac{2}{2m+1} * \sum_{i=0}^{2m} y_i coskx_i$$