|  |  |  |  |
| --- | --- | --- | --- |
| **AKADEMIA NAUK STOSOWANYCH W NOWYM SĄCZU**  Wydział Nauk Inżynieryjnych, Katedra informatyki | | | |
| Przedmiot: | Metody numeryczne – laboratorium, dr hab. Jabłoński Zenon | | |
| Grupa: | L2 | Nr sprawozdania: | 7 |
| Imię i Nazwisko: | Bartłomiej Cetera | Data: | 15.05.2023 |

**Zadanie (1) Algorytm Hornera**

**FUNKCJA**

function retval = Hor (a,y)

n=length(a);

t(n)=a(n);

for k=n-1:-1:1

t(k)=t(k+1)\*y+a(k);

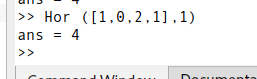
endfor

retval=t(1);

endfunction

**SKRYPT**

Hor ([1,0,2,1],1)



**WNIOSEK**

* W pierwszej linijce kodu definiowana jest funkcja "Hor", która przyjmuje dwa argumenty wejściowe: "a" i "y". "a" jest wektorem zawierającym współczynniki wielomianu, a "y" jest wartością, w której chcemy obliczyć wartość wielomianu interpolacyjnego.
* Następnie w drugiej linijce kodu obliczana jest długość wektora "a" za pomocą funkcji "length(a)", co pozwala na określenie stopnia wielomianu.
* W trzeciej linijce kodu tworzona jest pomocnicza zmienna "t", która jest wektorem o takiej samej długości jak wektor "a". Pierwszy element wektora "t" przyjmuje wartość ostatniego elementu wektora "a" (co odpowiada warunkowi brzegowemu dla interpolacji).
* Następnie w pętli "for" iterującej po wartościach od "n-1" do "1", w każdej iteracji obliczana jest wartość kolejnego elementu wektora "t" zgodnie ze wzorem Hornera, gdzie wartość poprzedniego elementu wektora "t" jest mnożona przez "y" i dodawana jest wartość odpowiedniego współczynnika z wektora "a".
* Na koniec, w ostatniej linijce kodu, zwracana jest wartość wielomianu interpolacyjnego dla punktu "y", która jest równa pierwszemu elementowi wektora "t".
* **Wielomian interpolacyjny może być łatwo obliczony za pomocą algorytmu Hornera, który pozwala na szybkie i efektywne obliczenie wartości wielomianu w danym punkcie.**

**Zadanie (2) Algorytm Neville’a**

**FUNKCJA**

function retval = Nev (x,w,y)

n=length(x)-1;

for r=0:n

t(r+1)=w(r+1);

for k=r-1:-1:0

t(k+1)=t(k+2)+(t(k+2)-t(k+1))\*(y-x(r+1))/(x(r+1)-x(k+1));

endfor

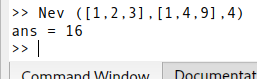
endfor

retval=t(1);

endfunction

**SKRYPT**

Nev ([1,2,3],[1,4,9],4)



**WNIOSEK**

* W pierwszej linijce kodu definiowana jest funkcja "Nev", która przyjmuje trzy argumenty wejściowe: "x", "w" i "y". "x" jest wektorem zawierającym węzły interpolacji, "w" jest wektorem wartości funkcji w tych węzłach, a "y" jest wartością, w której chcemy obliczyć wartość interpolowanej funkcji.
* Następnie w drugiej linijce kodu obliczana jest długość wektora "x" za pomocą funkcji "length(x)", co pozwala na określenie liczby węzłów interpolacji.
* W pętli "for" iterującej po wartościach od 0 do "n", w każdej iteracji tworzona jest pomocnicza zmienna "t", która jest wektorem o długości "r+1" (gdzie "r" jest wartością iteracji) i przyjmuje początkowo wartości funkcji w węzłach interpolacji "w(r+1)".
* W pętli "for" wewnętrznej iterującej po wartościach od "r-1" do 0, w każdej iteracji obliczana jest wartość kolejnego elementu wektora "t" zgodnie ze wzorem Neville'a, który polega na interpolacji wielomianowej pomiędzy dwoma punktami za pomocą wielomianu interpolacyjnego w punkcie pośrednim. W tym celu korzysta się z wartości wcześniej obliczonych elementów wektora "t".
* Na koniec, w ostatniej linijce kodu, zwracana jest wartość interpolowanej funkcji dla punktu "y", która jest równa pierwszemu elementowi wektora "t".
* **Metoda Neville'a pozwala na interpolację wartości funkcji w punkcie "y" na podstawie węzłów interpolacji "x" i wartości funkcji w tych węzłach "w" z użyciem wielomianu interpolacyjnego.**

**Zadanie (3) Wielomian Lagrange’a**

**FUNKCJA**

function retval = Lagr (x,f)

n=length(x)-1;

for k=1:n

for l=n:-1:k

f(l+1)=(f(l+1)-f(l))/(x(l+1)-x(l-k+1));

endfor

endfor

retval=f;

a(n+1)=f(n+1);

for i=n-1:-1:0

a(i+1)=f(i+1);

for k=i:n-1

a(k+1)=a(k+1)-x(i+1)\*a(k+2);

endfor

endfor

retval=a;

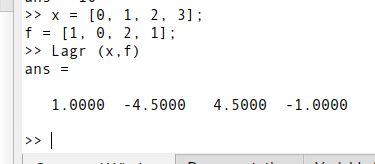
endfunction

**SKRYPT**

x = [0, 1, 2, 3];

f = [1, 0, 2, 1];

Lagr (x,f)



**WNIOSEK**

* W pierwszej linijce kodu definiowana jest funkcja "Lagr", która przyjmuje dwa argumenty wejściowe: "x" i "f". "x" jest wektorem zawierającym węzły interpolacji, a "f" jest wektorem wartości funkcji w tych węzłach.
* Następnie w drugiej linijce kodu obliczana jest długość wektora "x" za pomocą funkcji "length(x)", co pozwala na określenie liczby węzłów interpolacji.
* W pierwszej pętli "for" iterującej po wartościach od 1 do "n", w każdej iteracji wykonuje się druga pętla "for", która iteruje po wartościach od "n" do "k". W każdej iteracji drugiej pętli obliczana jest różnica wartości funkcji "f" dla dwóch kolejnych węzłów interpolacji i dzielona przez różnicę odpowiadających im wartości wektorów "x". Wynik jest zapisywany z powrotem do wektora "f".
* Następnie, w drugiej części funkcji, obliczany jest wielomian interpolacyjny Lagrange'a. W pierwszej linijce kodu inicjowany jest wektor "a", który jest wektorem współczynników wielomianu interpolacyjnego. W drugiej pętli "for" iterującej po wartościach od "n-1" do 0, w każdej iteracji tworzona jest pomocnicza zmienna "a", która przyjmuje początkowo wartości funkcji w węzłach interpolacji "f(i+1)". Następnie, w pętli "for" wewnętrznej iterującej po wartościach od "i" do "n-1", w każdej iteracji aktualizowany jest wektor "a" zgodnie z wzorem Lagrange'a. Wynik jest zapisywany z powrotem do wektora "a".
* Na koniec, w ostatniej linijce kodu, zwracany jest wektor współczynników wielomianu interpolacyjnego "a".
* Wielomian interpolacyjny Lagrange'a pozwala na interpolację wartości funkcji w punktach "x" na podstawie wartości funkcji w tych punktach "f".