

# Metody numeryczne

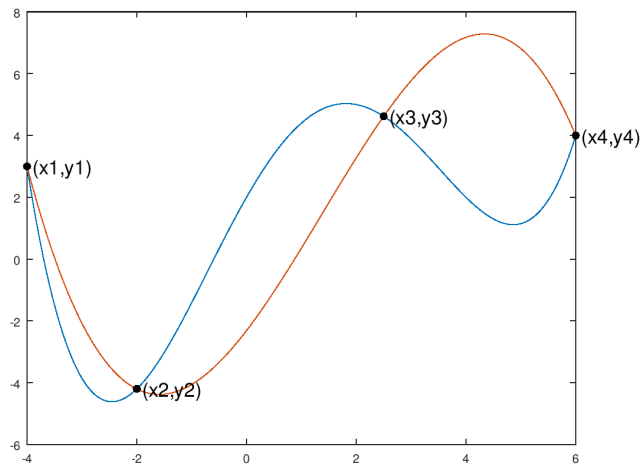
## Laboratorium 4

### Interpolacja i aproksymacja

## 1 Aproksymacja, interpolacja

Celem **aproksymacji** jest znalezienie zależności funkcyjnej  $F(x)$ , w przybliżeniu pokrywającej się z pewną funkcją  $f(x)$ , określoną w postaci ciągu punktów. Punkty te mogą pochodzić z pomiarów, albo mogą być wynikami innych obliczeń.

**Interpolacja** - metoda numeryczna polegająca na wyznaczeniu w danym przedziale tzw. funkcji intepolacyjnej, która przyjmuje w nim z góry zadane wartości w ustalonych punktach, nazywanych węzłami. Stosowana jest ona często w naukach doświadczalnych, gdzie dysponuje się zazwyczaj skończoną liczbą danych do określenia zależności między wielkościami oraz w celu uproszczenia skomplikowanych funkcji, np. podczas całkowania numerycznego. **Interpolacja** jest szczególnym przypadkiem metod numerycznych typu **aproksymacja**.



Rysunek 1: Przykładowa interpolacja funkcji

## 1.1 Wzór interpolacyjny Lagrange'a

Wzór Lagrange'a został opracowany na podstawie ogólnego wzoru na wielomian interpolujący. Dla  $n$  węzłów zapisuje się go w następujący sposób:

$$f(x) = \sum_{i=0}^{n-1} a_i \varphi_i(x) \quad (1)$$

gdzie:  $a_i$  - współczynnik interpolacji,  $\varphi_i$  - funkcja bazowa.

Liczba potrzebnych funkcji bazowych oraz współczynników interpolacji jest dokładnie taka sama jak liczba węzłów. Gdy jako nieznanne współczynniki  $a_0, a_1, \dots, a_{n-1}$  wstawimy znane wartości funkcji w węzłach  $y_0, y_1, \dots, y_{n-1}$  to do wyznaczenia pozostaną postaci funkcji bazowych  $\varphi_i$ . Jak wiadomo funkcja interpolująca powinna dla podanych węzłów  $x_i$  przyjmować ustalone wartości  $y_i$ . Jeżeli wartość funkcji bazowej  $\varphi_i$  dla podanego węzła  $x_i$  będzie wynosiła 1 a dla każdego innego węzła różnego niż  $x_i$  wynosić będzie 0 to ze wzoru (1) uzyskamy sumę:

$$f(x_i) = \dots + y_{i-1} \cdot 0 + y_i \cdot 1 + y_{i+1} \cdot 0 + \dots \quad (2)$$

czyli zostanie spełniony warunek  $f(x_i) = y_i$ . Wartości funkcji  $\varphi$  możemy więc rozpisać jako:

$$\varphi_i(x_j) = \begin{cases} 0 & \text{gd } j \neq i \\ 1 & \text{gd } j = i \end{cases} \quad (3)$$

Wzór na obliczenie tej funkcji został podany poniżej.

$$\varphi_i(x) = \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_{n-1})}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_{n-1})} \quad (4)$$

co zapisujemy jako:

$$\varphi_i(x) = \prod_{j=0, j \neq i}^{n-1} \frac{x - x_j}{x_i - x_j} \quad (0 \leq i < n). \quad (5)$$

Wzór pomimo skomplikowanego wyglądu jest bardzo prosty. Jeśli jako  $x$  podamy wartość  $x_i$  to licznik i mianownik ułamka będzie taki sam. Czyli iloraz będzie wynosił 1. Gdy podana zostanie wartość węzła innego niż  $x_i$  to któryś z iloczynów w liczniku da 0, czyli cały ułamek wynosić będzie 0.

### Przykład 1.

Mając dane węzły 0,1,3,8 wraz z wartościami 2,6,-1,8 obliczamy wielomian interpolacyjny:

$$\begin{aligned} W_n(x) &= 2 \frac{(x-1)(x-3)(x-8)}{(0-1)(0-3)(0-8)} + 6 \frac{(x-0)(x-3)(x-8)}{(1-0)(1-3)(1-8)} \\ &\quad - 1 \frac{(x-1)(x-0)(x-8)}{(3-1)(3-0)(3-8)} + 8 \frac{(x-1)(x-3)(x-0)}{(8-0)(8-3)(8-0)} \\ &= \frac{57}{140} x^3 - \frac{289}{70} x^2 + \frac{1081}{140} x + 2 \end{aligned}$$

```

1 function f = Lagrange(x, y, xi)
2 n = length(x);
3 f = 0;
4
5 for i=1:n
6     ilocz = 1;
7     for j=1:n
8         if(j!=i)
9             ilocz = ilocz * (xi-x(j))/(x(i)-x(j));
10        endif
11    endfor
12    f = f + #TODO!!#
13 endfor

```

Listing 1: Niekompletna funkcja interpolacji Lagrange'a

Powyższy skrypt przedstawia funkcję która jako argumenty przyjmuje:

- $x$  - wektor zawierający węzły interpolacji o wielkości  $n$  (indeksy w Octave zaczynają się od 1),
- $y$  - wektor wartości w węzłach interpolacji o wielkości  $n$ ,
- $xi$  - argument dla którego szukamy wartości

oraz powinna zwracać szukaną wartość w zmiennej  $f$ . Obecnie funkcja zwróci zawsze 0. W głównej pętli for z linii 5 następuje przejście przez wszystkie podane węzły. W pętli zagnieżdżonej liczona jest wartość funkcji bazowej dla każdego węzła. Linie 9 łatwo zrozumieć patrząc na wzór (5).

### Ćwiczenie 1.

Uzupełnij linie 12 z listingu 1. Funkcja powinna zwracać wartość funkcji interpolującej obliczonej zgodnie ze wzorem (1). Pamiętaj, że w zmiennej  $ilocz$  znajdują się odpowiednia wartość funkcji bazowej  $\varphi_i(x_i)$  oraz, że współczynnikami  $a_i$  są wartości podane w wektorze  $y$ .

Funkcję przetestuj na prostym przykładzie funkcji  $f(x) = x^2$ . Jako węzły przyjmij  $x = [1, 2, 3]$ . Sprawdź czy funkcja zwróci przewidywane wyniki dla innych wybranych argumentów podanych w  $xi$ .

## 1.2 Interpolacja w Octave

Do interpolacji wielomianowej w Octave można wykorzystać funkcję *polyfit* opisaną w rozdziale 1.5 dotyczącym aproksymacji. Gdy podany stopień wielomianu aproksymującego będzie  $n - 1$ , gdzie  $n$  oznacza liczbę węzłów to w efekcie otrzymamy wielomian interpolujący taki sam jaki uzyskalibyśmy metodą Lagrange'a. Octave udostępnia też inną, często używaną w praktyce interpolację funkcjami sklejanymi. Realizowana jest ona przez funkcję *interp1* a jej efekty zobaczyć można uruchamiając poniższy skrypt.

```

1 xf = -1:0.01:1; % Punkty do obliczania wartosci funkcji
2 yf = 1./(1 + 25*xf.^2); % Dokladne wartosci funkcji
3 xp = linspace(-1,1,10) ; % Wzly interpolacji

```

```

4 yp = 1./(1 + 25*xp.^2); % Wartości funkcji w węzłach
5 lin = interp1(xp,yp,xf);
6 cub = interp1(xp,yp,xf,"cubic");
7 spl = interp1(xp,yp,xf,"spline");
8
9 plot(xf, yf, "linewidth", 2);
10 hold on
11 plot(xf,lin, xp, yp, "r*");
12 legend("funkcja","interpolacja liniowa", "wezly");
13 figure(2)
14
15 plot(xf, yf, "linewidth", 2);
16 hold on
17 plot(xf,cub, xp, yp, "r*");
18
19 legend("funkcja","sześcienna hermita", "wezly");
20 figure(3)
21 plot(xf, yf, "linewidth", 2);
22 hold on
23 plot(xf,spl, xp, yp, "r*");
24 legend("funkcja","sześcienna spline", "wezly");

```

Listing 2: Interpolacja w octave

Funkcja *interp1* udostępnia trzy typy interpolacji:

- liniowa (domyślna) - pomiędzy każdymi kolejnymi węzłami funkcja przyjmuje inny wzór liniowy ( $ax + b$ )
- sześcienna Hermite'a - pomiędzy każdymi kolejnymi węzłami funkcja przyjmuje inny wzór sześcienny ( $ax^3 + bx^2 + cx + d$ ) z warunkiem ciągłości pierwszej pochodnej.
- sześcienna typu Spline - pomiędzy każdymi kolejnymi węzłami funkcja przyjmuje inny wzór sześcienny ( $ax^3 + bx^2 + cx + d$ ) z warunkiem ciągłości pierwszej i drugiej pochodnej.

### Ćwiczenie 2.

Sprawdź efekty zmiany ilości węzłów w powyższym listingu (linia nr 3). Przy ilu węzłach różnica między funkcją interpolującą, a interpolowaną będzie niezauważalna na wykresach dla opisanych typów interpolacji?

Uwaga! Przy kopiowaniu kodu z listingu przy argumentach podawanych jako łańcuch znaków dodawane są niepotrzebne znaki spacji które należy usunąć.

## 1.3 Efekt Rungego

Efekt Rungego jest to zjawisko pogarszania się dokładności przybliżenia funkcji interpolującej dla dużej ilości węzłów. Przebieg funkcji oscyluje na krawędziach przedziału. Zjawisko to występuje dla interpolacji wielomianem o dużym stopniu, gdzie węzły rozmieszczone są w równych odstępach. Aby zbadać to zjawisko spróbujemy dokonać interpolacji funkcji  $f(x) = \frac{1}{1+2x^2}$  dla dużej ilości równo rozmieszczonych węzłów. Wykorzystana zostanie funkcja napisana w ćwiczeniu

nr.1. Skrypt z listingu 3 generuje wykres zawierający 4 przebiegi. Pierwszy to oryginalna funkcja  $f(x) = \frac{1}{1+2x^2}$  której współrzędne generowane są w liniach 1, 2 oraz wyświetlana jest ona w linii 3. Następnie w liniach 6 – 14 generowane są trzy kolejne wykresy o liczbie węzłów 7, 9, oraz 11.

```

1 x2 = linspace(-5,5,200);
2 y2 = 1./(1+ 2*x2.^2);
3 plot(x2,y2); #Narysowanie funkcji oryginalnej
4 hold on;
5
6 for i=7:2:11
7     x = linspace(-5,5,i); #Ustalenie rowno odleglych wezlow
8     y = 1./(1+ 2*x.^2); #Ustalenie wartosci w wezlach
9     f=ones(length(x2),1);
10    for j=1:length(x2)
11        f(j) = Lagrange(x,y,x2(j));
12    endfor
13    plot(x2,f);
14 endfor
15 h = legend ("1/(1 + 2x^2)", "n = 7", "n = 9" ,"n = 11", ...
16 "location", "northeastoutside");
17 axis ([-5 5 -1 1])

```

Listing 3: Wizualizacja efektu Rungego

Uwaga! Do uruchomienia tego skryptu wymagane jest posiadanie dokończonego skryptu z ćw.1 zapisanego w pliku o nazwie “Lagrange.m”. Przy kopiowaniu, w linii 11 dodawany jest znak spacji który należy usunąć.

### Ćwiczenie 3.

Korzystając z funkcji *polyval* oraz *polyfit* oblicz i narysuj na wykresie funkcję interpolującą wzór  $f(x) = \frac{1}{1+2x^2}$  w przedziale  $\langle -5, 5 \rangle$  o 13 oraz 17 równo oddalonych węzłach. Dla ilu węzłów wystąpiły oscylacje o większej amplitudzie?

## 1.4 Wielomiany Czebyszewa

Aby wyeliminować efekt Rungego należy rozmieścić większą ilość węzłów na krańcach interpolowanego przedziału. Węzły te najlepiej jest przyjąć jako miejsca zerowe wielomianów Czebyszewa. Zagwarantowane jest wtedy zmniejszanie się maksymalnego błędu interpolacji wraz ze zwiększaniem ilości węzłów.

Wielomian  $T_k$  (gdzie  $k$  to stopień wielomianu Czebyszewa) posiada  $k$  różnych pierwiastków w punktach

$$x_m = \cos\left(\frac{2m+1}{2k}\pi\right), \quad m = 0, 1, \dots, k-1 \quad (6)$$

Węzły te będą spełniały oczekiwane rezultaty tylko dla interpolacji w przedziale  $\langle -1, 1 \rangle$ . Dla innych przedziałów wymagane będzie przeprowadzenie transformacji liniowej z przedziału  $\langle -1, 1 \rangle$  do innego  $\langle a, b \rangle$ , gdzie  $a$  będzie najmniejszą wartością węzła a  $b$  największą dla przeprowadzanej interpolacji. Dzięki transformacji każde miejsce zerowe wielomianu Czebyszewa zostaje sprowadzone do przedziału  $\langle a, b \rangle$  z zachowaniem proporcji odpowiednich wartości

funkcji. Wzór obliczający położenie tych miejsc zerowych przekształconych do przedziału  $\langle a, b \rangle$  został podany poniżej.

$$x_m = \frac{1}{2} \left[ \cos \left( \frac{2m+1}{2k} \pi \right) (b-a) + (b+a) \right] \quad m = 0, 1, \dots, k-1 \quad (7)$$

#### Ćwiczenie 4.

Napisz funkcję która dla podanego zakresu  $\langle a, b \rangle$  oraz stopnia wielomianu zwróci jego miejsca zerowe przekształcone do tego zakresu. Możesz skorzystać z poniższego listingu odpowiednio wypełniając linie 4 zgodnie z równaniem (7). Korzystając z obliczonych za pomocą tej funkcji węzłów przeprowadź interpolację tej samej funkcji, o takim samym przedziale jak w ćwiczeniu 3. Sprawdź jak wyglądać będzie wykres funkcji dla 20, 30 węzłów obliczonych za pomocą tej funkcji w porównaniu z równo oddalonymi węzłami.

```

1 function x = Czebyszew(a,b,n)
2 x=ones(n,1);
3 for i=0:n-1
4 #TODO!#
5 endfor

```

Listing 4: Przekształcenie miejsc zerowych wielomianu Czebyszewa

## 1.5 Aproksymacja w Octave

Do aproksymacji wielomianowej w Octave stosuje się funkcję *polyfit* która jako argumenty przyjmuje współrzędne  $[x, y]$  węzłów oraz stopień wielomianu aproksymującego.. Funkcja *polyfit* zwraca współczynniki wielomianu aproksymującego. Aby wyliczyć jego wartości można skorzystać z funkcji *polyval* która na podstawie wektora współczynników wielomianu, oraz wektora argumentów zwraca wartości tego wielomianu. Przykładowe wykorzystanie tych dwóch funkcji dla  $f(x) = -2x^3 - 2x^2 + 42x + 90$  zostało podane poniżej.

```

1 x = -5:1:6; %wezly
2 y = -2*x.^3 - 2*x.^2 + 42*x + 90; %wartosci w wezlach
3 p1 = polyfit(x,y,1); %wielomiany aproksymujace
4 p2 = polyfit(x,y,2);
5 p3 = polyfit(x,y,3);
6 x2 = linspace(-5,6); %punkty dla ktorzych szukana bedzie
7 %wartosc
8 plot(x2,polyval(p1,x2)); %obliczenie wartosci w punktach i
9 %wyswietlenie ich na wykresie
10 hold on;
11 plot(x2,polyval(p2,x2));
12 plot(x2,polyval(p3,x2), x, y, "r*");
13 h = legend("n=1","n=2","n=3","wezly","location", ...
14 "northeastoutside");

```

Listing 5: Aproksymacja w Octave

Po wykonaniu powyższego listingu wyświetlone zostają na wykresie wyniki aproksymacji funkcji  $f(x) = -2x^3 - 2x^2 + 42x + 90$  wielomianami stopnia 2,3

oraz 4. Jak widać wielomian 4 stopnia przechodzi przez wszystkie zaznaczone węzły więc jest to już interpolacja.

### Ćwiczenie 5\*.

Przekształć funkcję realizującą algorytm interpolacji Lagrange'a z listingu 1 tak aby przyjmowana wartość  $x_i$  oraz zwracana  $f$  mogły być wektorami.

## 1.6 Interpolacja Newtona

Wzór interpolacyjny Newtona wywodzi się z podejścia rekurencyjnego do interpolacji. Jeśli przez  $W_n(x)$  oznaczymy wielomian interpolujący  $n$ -tego stopnia o węzłach  $x_0, x_1, \dots, x_n$  to wielomian interpolujący posiadający dodatkowy węzeł  $x_{n+1}$ , z resztą węzłów równą wielomianowi  $W_n(x)$  obliczymy ze wzoru:

$$W_{n+1}(x) = W_n(x) + c(x - x_0)(x - x_1) \dots (x - x_n); \quad (8)$$

Który możemy rozpisać na:

$$W_n(x) = c_0 + c_1(x - x_0) + \dots + c_n(x - x_0)(x - x_1) \dots (x - x_{n-1}) \quad (9)$$

lub w stosując zapis skrótowy:

$$W_n(x) = \sum_{i=0}^n c_i \prod_{j=0}^{i-1} (x - x_j) \quad (10)$$

Niewiadomymi w tym wzorze są współczynniki  $c_0, c_1, \dots, c_n$ . Można obliczać je rekurencyjnie z pomocą równań:

$$c_0 = y_0 \quad (11)$$

$$y_1 = c_0 + c_1(x_1 - x_0) \quad (12)$$

$$y_2 = c_0 + c_1(x_2 - x_0) + c_2(x_2 - x_0)(x_2 - x_1) \quad (13)$$

⋮

Z równania (11) oblicza się  $c_0$ , z równania (12)  $c_1$  itp. Jednak dużo wydajniejszym sposobem jest stworzenie tablicy ilorazów różnicowych.

Jako, że dany współczynnik  $c_i$  zależy wyłącznie od węzłów  $x_0, x_1, \dots, x_i$  i wartości w tych węzłach to zależność tą możemy zapisać następująco:

$$c_i = f[x_0, x_1, \dots, x_i] \quad (14)$$

gdzie  $f[x_0, x_1, \dots, x_i]$  jest ilorazem różnicowym węzłów  $x_0, x_1, \dots, x_k$ .

Ilorazy różnicowe zerowego oraz pierwszego rzędu zapisujemy w następujący sposób:

$$f[x_0] = y_0 \quad (15)$$

$$f[x_0, x_1] = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (16)$$

Ilorazy wyższych rzędów oblicza się rekurencyjnie stosując wzór podobny do obliczenia ilorazu rzędu pierwszego czyli równania (16). Dokładny wzór został pokazany poniżej.

$$f[x_i, x_{i+1}, \dots, x_{i+j}] = \frac{f[x_{i+1}, x_{i+2}, \dots, x_{i+j}] - f[x_i, x_{i+1}, \dots, x_{i+j-1}]}{x_{i+j} - x_i} \quad (17)$$

Obliczone w ten sposób ilorazy zapisujemy za pomocą tablicy. Jej przykładowa postać dla węzłów  $x_0, x_1, x_2, x_3$  oraz wartościach w tych węzłach oznaczonych jako  $y_0, y_1, y_2, y_3$  wygląda następująco:

Tabela 1: Sposób wypełniania tablicy ilorazów różnicowych

$x_0$	$f[x_0]$	$f[x_0, x_1]$	$f[x_0, x_1, x_2]$	$f[x_0, x_1, x_2, x_3]$
$x_1$	$f(x_1)$	$f[x_1, x_2]$	$f[x_1, x_2, x_3]$	
$x_2$	$f(x_2)$	$f[x_2, x_3]$		
$x_3$	$f(x_3)$			

Zaznaczone kolorem elementy tablicy są szukanymi współczynnikami  $c$ . Przykładowe wypełnienie tablicy ilorazów różnicowych oraz sposób budowy wielomianu interpolacyjnego został pokazany poniżej. Jako węzły przyjęto  $x = [1, 3, 5]$  oraz  $y = [2, 6, 18]$ .

Tabela 2: Tablica ilorazów dla przykładowych danych

x	y		
1	2	$\frac{6-2}{3-1} = 2$	$\frac{6-2}{5-1} = 1$
3	6	$\frac{18-6}{5-3} = 6$	
5	18		

Obliczone współczynniki to  $c_0 = 2, c_1 = 2, c_2 = 1$ . Podstawiając je to równania (10) otrzymujemy funkcję interpolacyjną.

$$f(x) = 2 + 2(x - 1) + (x - 1)(x - 3) \quad (18)$$

Aby sprawdzić ten wzór podstawiamy do niego węzły.

$$\begin{aligned} f(1) &= 2 + 2(1 - 1) + (1 - 1)(1 - 3) = 2 \\ f(3) &= 2 + 2(3 - 1) + (3 - 1)(3 - 3) = 6 \\ f(5) &= 2 + 2(5 - 1) + (5 - 1)(5 - 3) = 18 \end{aligned}$$

Jak widać spełnione są warunki interpolacji.



**Ćwiczenie 6\*.**

Napisz funkcję *function*  $f = \text{Newton}(x, y, xi)$  obliczającą wartość  $xi$  funkcji interpolacyjnej o węzłach  $x, y$  zbudowanej na podstawie algorytmu Newtona. Pierwszym krokiem powinno być zbudowanie tablicy ilorazów różnicowych w macierzy. Następnie należy wyciągnąć z niej elementy będące współczynnikami oraz obliczyć wartość funkcji interpolacyjnej zgodnie ze wzorem (9)