04 **Techniki Obliczeniowe w Nauce i Technice**
Interpolacja i ekstrapolacja
dr inż. Jarosław Bułat

2016.11.05

1. Interpolacja wielomianowa (2 pkt)

W poniższej tabeli przedstawiono wyniki pomiaru temperatury (*N*=4):

Godzina	(x_k, k=1,2,3,4)	x_1=5:00	x_2=6:00	x_3=8:00	x_4=11:00
Temperatura [°C]	(y_k, k=1,2,3,4)	y_1=1	y_2=2	y_3=7	y_4=15

Narysuj ciągły przebieg temperatury od godziny 5:00 do 11:00. Nieistniejące próbki przybliż wielomianem interpolującym. Wykorzystaj metodę Lagrange'a i Newtona.

Metoda Lagrange'a: Potraktuj godziny jako ułamek dziesiętny. Oblicz $y(x)$ dla $x=5, 5.1, 5.2, \ldots, 11$ według zależności:

$$y(x)=\sum_{k=1}^{N} y_k L_k(x) \quad \text{, gdzie } N=4, \quad L_k(x)=\frac{\prod_{j=1,\ldots,N,\, j\neq k}(x-x_j)}{\prod_{j=1,\ldots,N,\, j\neq k}(x_k-x_j)}$$

Wielomiany Lagrange'a $L_k(x)$ zaimplementuj jako funkcje w języku Matlab według poniższego wzorca:

```
function y = Lk( x, xk, k )
% x  - argument x (skalar)
% xk -wektor wszystkich znanych argumentów (wektor N-elementowy)
% k  - numer wielomianu
% y  - obliczona wartość wielomianu Lk (wynik)
```

Metoda Newtona: Oblicz $y(x)$ dla parametrów oraz zakresu jw. według zależności:

$$y(x)=a_0+\sum_{k=1}^{N-1}\left(a_k \prod_{j=1,\ldots,k}(x-x_j)\right) \quad \text{, gdzie} \quad a_{i-1}=d_{i,i}$$

Parametry $d_{k,i}=\dfrac{d_{k,j-1}-d_{k-1,j-1}}{x_k-x_{k-j+1}}, d_{k,1}=y_k$ wyznaczane są następującym kodem:

```
d(:,1) = yk;
for j=2:N
    for k=j:N
        d(k,j)=( d(k,j-1)-d(k-1,j-1) )/( xk(k)-xk(k-j+1) );
    end
end
```

Porównaj przebieg interpolacji wielomianami Lagrange'a i Newtona. Na jedynym wykresie wyrysuj znane punkty oraz obie krzywe. Oblicz temperaturę o godzinie 7:15. W jaki sposób wyznaczyć o której godzinie temperatura przekroczyła 3 [°]?

2. Interpolacja funkcjami sklejanymi (1 pkt)

Porównaj trzy metody interpolacji dla procesu opisanego zależnością:

$$f(x)=\frac{1}{1+x^2}$$

zdefiniowanego dla argumentów: x=-5, -4, …, 5. Wykonaj interpolację:

1. Wielomianową algorytmem Lagrange'a lub Newtona zaimplementowanym w ćwiczeniu 1.
2. Funkcjami sklejanymi I stopnia (interpolacja liniowa). Wykorzystaj algorytm bezpośredni:
$$y(x_i\leq x\leq x_{i+1})=S_i(x)=y_i+\frac{(y_{i+1}-y_i)}{(x_{i+1}-x_i)}(x-x_i)$$

3. Funkcjami sklejanymi III stopnia (cubic spline). Wykorzystaj funkcje spline Matlaba według poniższego fragmentu kodu.

```
xk = (-5:1:5)';
yk = 1./(1+xk.^2);

% funkcje sklejane - III stopień
cs = spline( xk', [0 yk' 0] );
y3 = ppval( cs, x );
```

Wyświetl na jednym wykresie funkcję oryginalną, oraz odtworzoną algorytmami opisanymi w pkt. 1, 2 oraz 3. Oblicz maksymalny błąd popełniany przez funkcje interpolujące w przedziale <-5,5>.

3. Interpolacja 2D (2 pkt)

Poniższy fragment kodu wczytuje z pliku obrazek w celu jego obrotu o dowolny kąt i przesunięcia. Relacja pomiędzy współrzędnymi obrazka wynikowego `out(i,j)` oraz wejściowego `in(i,j)` jest opisana przez tablice `Xi(i,j)` oraz `Yi(i,j)`. W następujący sposób:

`out(i,j) = in( Xi(i,j), Yi(i,j) )`

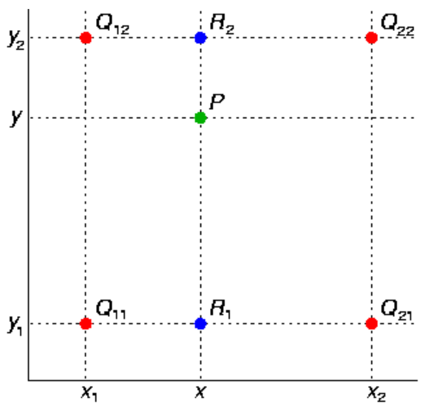
Zmienne `i`, `j` to współrzędne pikseli (i,j=1, 2, 3, ..., N). Wartości w tablicach `Xi` i `Yi` są rzeczywiste i zazwyczaj wskazują na współrzędne ,,pomiędzy'' pikselami oryginalnego obrazka. Dlatego wymagana jest interpolacja.

```
clear all;
close all;

t=pi/6;
rot_mtx=[cos(t), -sin(t); sin(t), cos(t)];
T=[50;-20];

in = imread( 'checkerBoard_20_200.png' );
in = double( rgb2gray( in ) );

N = size( in, 1 );
[X,Y]=meshgrid( 1:N );

XY=[ reshape(X,1,N*N); reshape(Y,1,N*N) ];
XYr=rot_mtx*(XY+repmat(T,1,N*N));

Xi=reshape(XYr(1,:),N,N);
Yi=reshape(XYr(2,:),N,N);
```

Utwórz obrazek wyjściowy przy wykorzystaniu interpolacji zerowego stopnia (najbliższy sąsiad) oraz interpolacji funkcjami sklejanymi I stopnia – interpolacji liniowej. W przypadku dwuwymiarowym interpolację liniową nazywamy interpolacją biliniową.

Interpolację biliniową można przedstawić następująco[1]: (x1,x1), (x1,y2), (x2,y1) i (x2,y2) to współrzędne sąsiadujących pikseli oryginalnego obrazka o jasnościach odpowiednio: Q11, Q12, Q21 i Q22 co zostało przedstawione na poniższym obrazku. P to poszukiwany punkt, którego jasność wyraża się zależnością:

$$P=\frac{Q_{11}}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y_2-y)+\frac{Q_{21}}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y_2-y)+\frac{Q_{12}}{(x_2-x_1)(y_2-y_1)}(x_2-x)(y-y_1)+\frac{Q_{22}}{(x_2-x_1)(y_2-y_1)}(x-x_1)(y-y_1)$$



1 http://en.wikipedia.org/wiki/Bilinear_interpolation

4. Interpolacja 3D (opcjonalnie 3 pkt)

Fragment danych, zebranych kamerą rejestrującą głębię (Kinect), został zapisany w pliku `X.mat` w wektorze `X` o 3 kolumnach, reprezentujących kolejno współrzędne x, y i z. Liczba wierszy wektora odpowiada liczbie zarejestrowanych punktów.

Wyświetl dane w przestrzeni 3D za pomocą ,,kropek'' używając funkcji `plot3(...)`. Zauważ, że Matlab pozwala na manipulację zbiorem danych – obrót, przybliżenie i przesunięcie.

Na podstawie wektora `X`, wykorzystując interpolację ,,linear'' i ,,cubic'', utwórz powierzchnie o regularnej siatce w płaszczyźnie x-y w zakresie -10<y<10 (poziom), -5<x<15 (pion), a następnie pokaż ją za pomocą funkcji `sufrf(...)`.

Punkty, które w oryginalnym wektorze nie są reprezentowane, a leżą poza obrysem obiektu, powinny zostać zastąpione punktami oddalonymi od kamery tak daleko jak najdalszy punkt reprezentowany w `X`.

3 punkty można otrzymać za wykonanie zadania bez pomocy funkcji `griddata(...)`.