Ćwiczenia z Metod Numerycznych II A
rok akademicki 2001/2002

seria 3

Jakub Narębski Tomasz Pawłowski

22 listopada 2004 roku

1 Interpolacja wielomianowa

Zadanie interpolacyjne Lagrange'a polega na znalezieniu dla danej funkcji f (dla zadanych liczb f_i) wielomianu L_n stopnia nie wyższego niż n, którego wartości w $n+1$ punktach x_i są takie same jak wartości funkcji f: $L_n(x_i) = f(x_i)$ (są równe zadanym liczbom f_i: $L_n(x_i) = f_i$. Wielomian n–tego stopnia jest jednoznacznie wyznaczony przez wartości w $n+1$ punktach.

1.1 Interpolacja liniowa

Przez dwa punkty (x_1, y_1) i (x_2, y_2) możemy przeprowadzić prostą. Aby obliczyć wartość y przy zadanym x dla punktu leżącego na tej prostej najlepiej (numerycznie) posłużyć się wzorem:

$$y = Ay_1 + By_2, \text{ gdzie } A = \frac{x_2 - x}{x_2 - x_1}, B = \frac{x - x_1}{x_2 - x_1}. \tag{1}$$

Łatwo sprawdzić, że $A(x = x_1) = 1$, $B(x = x_1) = 0$, natomiast $A(x = x_2) = 0$, $B(x = x_2) = 1$.

1.2 Wzór Lagrange'a

Naturalnym rozszerzeniem wzoru (1) z podrozdziału 1.1 na interpolację liniową do interpolacji wielomianem dowolnego stopnia jest klasyczny **wzór Lagrange'a**:

$$\begin{aligned} P(x) = {} & \frac{(x - x_2)(x - x_3)\dots(x - x_N)}{(x_1 - x_2)(x_1 - x_3)\dots(x_1 - x_N)} y_1 + \frac{(x - x_1)(x - x_3)\dots(x - x_N)}{(x_2 - x_1)(x_2 - x_3)\dots(x_2 - x_N)} y_2 \\ & + \dots + \frac{(x - x_1)(x - x_2)\dots(x - x_{N-1})}{(x_N - x_1)(x_N - x_2)\dots(x_N - x_{N-1})} y_N \end{aligned} \tag{2}$$

Powyższy wzór określa wielomian $N-1$ stopnia, taki że $y_1 = P(x_1)$, $y_2 = P(y_2)$, $\dots$, $y_N = P(x_N)$. Postać ta służy do udowodnienia twierdzenia o jednoznaczności zagadnienia interpolacji wielomianowej (zagadnienia interpolacyjnego Lagrange'a):

Twierdzenie 1.1 *Zadanie interpolacyjne Lagrange'a ma jednoznaczne rozwiązanie, które można przedstawić np. w postaci:*

$$L_n(x) = \sum_{i=0}^{n} f(x_i) l_i(x), \tag{3}$$

gdzie $l_i(x)$ to funkcje pomocnicze zdefiniowane jako

$$l_i(x) = \prod_{\substack{j=0 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}, \quad dla\ i = 0, 1, \ldots, n. \tag{4}$$

Wielomian L_n nazywa się wielomianem interpolacyjnym Lagrange'a $n+1$–go stopnia funkcji f opartym na węzłach $x_0, x_1, \ldots, x_n$. (*Uwaga:* oznaczenia węzłów są inne niż w równaniu (2)!).

Wzór Lagrange'a (2) na wielomian interpolacyjny nie jest zbyt wygodny w zastosowaniu. Nie daje żadnego oszacowania błędu, ponadto jest trudny do zaprogramowania. Dlatego do interpolacji wielomianowej w punkcie używa się raczej przedstawionego w następnym podrozdziale **algorytmu Neville'a** (czasem zwanego *metodą Aitkena*).

1.3 Algorytm Neville'a

Zadanie interpolacyjne można początkowo rozpatrzyć dla mniejszej ilości punktów węzłowych, a następnie adoptować otrzymane rozwiązania dla otrzymania właściwego wielomianu interpolacyjnego.

Zdefiniujmy kolejne wielomiany interpolacyjne i ich wartości w zadanym punkcie x. p_i jest wielomianem zerowego stopnia (czyli stałą) przechodzącą przez punkt (x_i, y_i), a $p_i = y_i$ jego wartością w punkcie x. $p_{i(i+1)}$ będzie wartością w punkcie x wielomianu drugiego stopnia przechodzącego przez punkty (x_i, y_i) i (x_{i+1}, y_{i+1}). Analogicznie $p_{i(i+1)(i+2)}$ itd. Dla czterech punktów x_0, x_1, x_2, x_3 wyglądało by to następująco:

$$\begin{array}{llccc}
x_0: & y_0 = p_0 & & & \\
 & & p_{01} & & \\
x_1: & y_1 = p_1 & & p_{012} & \\
 & & p_{12} & & p_{0123} \\
x_2: & y_2 = p_2 & & p_{123} & \\
 & & p_{23} & & \\
x_3: & y_3 = p_3 & & &
\end{array}$$

Ogólnie $p_{k\ldots l}(x)$ — wartość, w punkcie x, wielomianu n-tego stopnia przechodzącego przez $l - k + 1$ punktów (x_i, y_i), $(i = k, \ldots, l)$.

Wartości wielomianów tego typu spełniają następujący związek rekurencyjny:

$$p_{i(i+1)\ldots(i+k)} = \frac{(x - x_{i+k}) p_{i(i+1)\ldots(i+k-1)} + (x_i - x) p_{(i+1)(i+2)\ldots(i+k)}}{x_i - x_{i+k}}, \tag{5}$$

gdzie k odpowiada stopniu wielomianu, $k = 1, \ldots, n$.

Algorytm ten możemy zapisać w innej notacji: $t_{i+k,k} \equiv p_{i(i+1)\ldots(i+k)}$. Pierwszy indeks w $t_{i,j}$ oznacza ostatni z węzłów na którym oparty jest wielomian, zaś drugi oznacza krok iteracji (stopień wielomianu). W tej notacji związek rekurencyjny przyjmuje postać:

$$
\begin{aligned}
t_{kl} &= \frac{(x_{k-l}-x)t_{k,l-1}-(x_k-x)t_{k-1,l-1}}{x_{k-l}-x_k} = \\
&= t_{k,l-1}+(t_{k,l-1}-t_{k-1,l-1})\frac{x-x_k}{x_k-x_{k-l}} .
\end{aligned}
\tag{6}
$$

Implementując metodę Neville'a warto zauważyć, że można ją zaimplementować korzystając wyłącznie z jednego wektora pomocniczego o rozmiarze równym ilości punktów na których rozpinamy interpolację, $\mathcal{O}(m)$. Nie należy stosować rekurencji, ponieważ skutkuje ona wielokrotnym wyliczaniem tych samych wartości pomocniczych i wymaga więcej pamięci. Zakładając, że chcemy aby końcowy wynik interpolacji znalazł się w tablicy pod indeksem 0, możemy pożądany rezultat uzyskać na (przynajmniej) dwa sposoby, opisane poniżej. Załóżmy, że na początku mamy tablicę w[0..3] wypełnioną wartościami $p_i = y_i$.

w[0]	$y_0 = p_0$	p_{01}	p_{012}	p_{0123}
w[1]	$y_1 = p_1$	p_{12}	p_{123}	
w[2]	$y_2 = p_2$	p_{23}		
w[3]	$y_3 = p_3$			

Pierwszą z nich jest wyliczanie wartości kolumna po kolumnie, tzn w kolejności $p_{01}, p_{12}, p_{23}; p_{012}, p_{123}; p_{0123}$, korzystając z faktu że np. po wyliczeniu p_{01} wartość p_0 nie będzie już potrzebna.

Drugą jest wyliczanie wartości skosem (po przekątnej), wyliczając w wewnętrznej iteracji wszystkie $p_{\ldots i}$ kończące się w danym punkcie, tzn. w kolejności $p_{01}; p_{12}, p_{012}; p_{23}, p_{123}, p_{0123}$.

Aby przybliżenie interpolacyjne miało tendencję do optymalności i było szybko zbieżne należy uporządkować punkty w kolejności rosnącej odległości od punktu intepolacji, tzn. $|x_i - x| \leq |x_j - x|$ dla $i < j$. Ulepszona metoda (patrz następny rozdział) nie ma tego ograniczenia.

1.4 Ulepszenie metody Neville'a (z Numerical Recipies)

Równanie rekurencyjne (5) metody Neville'a można ulepszyć obliczając małe *różnice* między przodkami (wynikami cząstkowymi dla k-tego stopnia) i potomkami (wynikami dla $k+1$-go stopnia). Zdefiniujmy

$$D_{i,m} \equiv p_{i\ldots(i+m)} - p_{i\ldots(i+m-1)} \tag{7a}$$

$$U_{i,m} \equiv p_{i\ldots(i+m)} - p_{(i+1)\ldots(i+m)} \tag{7b}$$

Z wzoru (5) łatwo można wyprowadzić wzory rekurencyjne na $D_{i,m}$ (ang. ***down***) i $U_{i,m}$ (ang. ***up***):

$$D_{i,m+1} = \frac{D_{i+1,m} - U_{i,m}}{x_i - x_{i+m+1}}(x_i - x) \tag{8a}$$

$$U_{i,m+1} = \frac{D_{i+1,m} - U_{i,m}}{x_i - x_{i+m+1}}(x_{i+m+1} - x) \tag{8b}$$

Na każdym poziomie m, wielkości $D_{i,m}$ i $U_{i,m}$ są poprawkami potrzebnymi do interpolacji jeden rząd wyżej. $p_{0\ldots n}$ będące końcowym wynikiem otrzymujemy biorąc sumę *dowolnego* y_i oraz zestawu D i/lub U które tworzą ścieżkę przez drzewo od wybranego y_i aż do $p_{0\ldots n}$.

W celu minimalizacji błędów najlepiej zacząć od y_i należącego do pary (x_i, y_i) dla x_i najbliższego punktowi w którym interpolujemy. W implementacji z „Numerical Recipies" prowadzimy ścieżkę jak najbardziej środkiem drzewa, wybierając ścieżkę w dół gdy jestesmy powyżej środka i ścieżkę w górę gdy jesteśmy poniżej środka. Ostatnia poprawka może służyć jako oszacowanie błędu interpolacji.

Uwaga: notacja jest nieco zmieniona w porównaniu z „Numerical Recipies in C".

2 Znajdowanie punktów, na których chcemy oprzeć interpolację

Często jest tak, że funkcję mamy zadaną za pomocą wielu punktów, a interpolację w zadanym punkcie x chcemy oprzeć tylko o kilka najbliższych x punktów. Przykładem może być interpolacja wielomianowa: wielomiany wyższego stopnia zazwyczaj silnie oscyluja i moga dawać błędne wyniki interpolacji.

Zakładamy, że punkty w których mamy stablicowaną funkcję są ułożone rosnąco, t.j. $-\inf < x_0 \leq x_1 \leq \cdots \leq x_{n-1} < +\int$. Mamy zadaną wartość x, chcemy znaleźć taki indeks i, że $x_i \leq x \leq x_{i+1}$. Wartość $i = -1$ oznacza $x_{-1} = -\infty$, wartość $i = n$ oznacza $x_n = +\infty$.

- Metoda **bisekcji**. W każdym kroku połowimy przedział w którym musi znajdować się x.
- Metoda **polowania**. Zaczynamy z jakiegoś przybliżenia na indeks i^*, następnie poruszamy się w krokach będących kolejnymi potęgami dwójki (1, 2, 4, itp.) aż „osaczymy" x. Następnie wykonujemy bisekcję zaczynając od znalezionego przedziału.

Aby przekazać funkcji interpolującej (niech nazywa się ona polint) odpowiedni fragment tablicy o długości m (chcemy oprzeć interpolację w punkcie x na m najbliższych mu punktach), należy znaleźć początek (początkowy indeks) fragmentu oraz jego długość. Zakładamy, że znaleźliśmy takie i, że $x_i \leq x \leq x_{i+1}$, tablica jest numerowana od 0 i ma n elementów. Początkowy indeks znajdujemy za pomocą wzoru:

$$k = \min(\max(i - \frac{m}{2}, 1), n - m). \tag{9}$$

W ten sposób zabezpieczamy się przed przypadkiem $k < 0$ oraz $k + m > n$ (wyjściem poza tablicę). Długość fragmentu to oczywiście $\min(m, n - k)$. Naszą funkcję wywołujemy zatem z parametrami

```
polint(tab + k, MIN(m, n−k), \dots),
```

gdzie MIN(a, b) możemy zdefiniować jako **#define** MIN(a,b) ((a)>(b)?(a):(b))

Zadanie 1. Napisać procedurę znajdującą dla zadanej wartości x taki indeks i w uporządkowanej rosnąco tablicy `xx` (`xx[0]` $<$ `xx[1]` $< \ldots <$ `xx[N-1]`, gdzie `N` jest rozmiarem tablicy), że `xx[i-1]` $< x \leq$ `xx[i]` metodą **bisekcji**. Jeśli x jest mniejsze od najmniejszego elementu tablicy (czyli od `xx[0]`) należy zwrócić wartość `0`, jeżeli natomiast x jest większe od największego elementu tablicy (czyli od `xx[N-1]`) należy zwrócić `N`.

Zadanie 2. Napisać procedurę, znajdującą dla zadanego przybliżenia i^* i wartości x taki indeks i w uporządkowanej rosnąco tablicy `xx`, że `xx[i-1]` $<$ $x \leq$ `xx[i]`. Należy zastosować algorytm **polowania** zaczynając od indeksu i^*. Można skorzystać z napisanej poprzednio procedury (funkcji) stosującej algorytm bisekcji.

3 Zadanie domowe

Rok akademicki 2004/05.

Napisać program dokonujący interpolacji wielomianem zadanego stopnia za pomocą metodą Neville'a. Przy funkcji zadanej na $N >= m+1$ punktach (gdzie m jest stopniem wielomianu interpolacyjnego), wyznaczać wartość wielomianu interpolacyjnego w punkcie na podstawie interpolacji opartej o najbliższe (w sensie odległości albo w sensie indeksów) $m+1$ punktów.

Przykładowe funkcje do testowania:

$$
\begin{aligned}
f(x) &= |x|\,, \\
f(x) &= \sin(\pi x)\,, \\
f(x) &= \frac{1}{1+x^2}\,, \\
f(x) &= \exp\bigl(-x^2\bigr)\,, \\
f(x) &= \exp\left(\frac{-1}{x^2}\right)\,, \\
f(x) &= a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n \quad \text{wielomian}\,.
\end{aligned}
$$

Wymagania:

1. Napisanie funkcji (procedury, metody) znajdującej wartość interpolacji wielomianowej wielomianem określonego stopnia m, rozpiętej na $m+1$ punktach, w zadanym punkcie x (70%). Za użycie rekurencji bądź macierzy pomocniczej -10%. Za użycie algorytmu poprawek z Numerical Recipies +15%.

2. Napisanie kodu generującego wykres interpolacji wielomianowej, zadanych punktów i porównanie interpolacji z interpolowaną (prawdziwą) funkcją (10%). Program może zapisywać wyniki do pliku (punkty interpolacji oraz wynik interpolacji), a wykres może być robiony zewnętrznym programem np. `gnuplot`.

3. Napisanie funkcji znajdującej punkty na których należy rozpiąć interpolację, przy zadanym punkcie interpolacji (i w przypadku metody polowania zgadniętego indeksu) oraz stopniu wielomianu interpolującego. Użyć funkcji pomocniczej do znalezienia indeksu "środkowego", tzn takiego że xx[i] $<=$ x $<$ xx[i+1] (20%).

Przygotował: Jakub Narębski