Ćwiczenia z Metod Numerycznych II A
rok akademicki 2001/2002

seria 6

Jakub Narębski Tomasz Pawłowski

21 stycznia 2005 roku

1 Interpolacja funkcjami sklejanymi

1.1 Sformułowanie zadania

Przy interpolacji Lagrange'a lub wymiernej zakłada się, że dana funkcja f przybliżana jest **jednym** wielomianem (funkcją wymierną) na całym odcinku $[a, b]$. Przy tym założeniu jedyną możliwością uzyskania lepszego przybliżenia jest zwiększenie stopnia wielomianu. Nie zawsze musi to jednak działać (przykładem może być funkcja $1/(1 + x^2)$ interpolowana za pomocą wielomianu). W tej sytuacji naturalne jest inne sformułowanie zadania.

Niech (Δ) będzie układem $N+1$ punktów $x_0, x_1, \ldots, x_N$ dzielących przedział $[a, b]$ na N części, tzn.

$$(\Delta): \quad a = x_0 < x_1 < \ldots < x_N = b \tag{1}$$

W każdym z podprzedziałów $[x_i, x_{i+1}]$ przybliżamy daną funkcję f wielomianem ustalonego (raczej niskiego) stopnia. Zakładamy ponadto, że funkcja interpolująca była ciągła wraz z odpowiednimi pochodnymi na **całym odcinku** $[\mathbf{a}, \mathbf{b}]$. Funkcjami o tych własnościach są **funkcje sklejane** (ang. spline). Omówimy tu własności interpolacyjnych funkcji sklejanych i przedstawimy algorytm ich wyznaczania dla przypadku funkcji stopnia 3.

1.2 Ogólne fakty dotyczące funkcji sklejane

Definicja 1.1 *Funkcję rzeczywistą S nazywamy* ***funkcją sklejaną*** *stopnia m z węzłami $x_0, x_1, \ldots, x_N$, gdzie $x_0 < x_1 < \ldots < x_N$, jeżeli spełnia poniższe własności:*

- *w każdym przedziale (x_{i-1}, x_i) dla $i = -1, 0, 1, \ldots, N+1$ $(x_{-1} = -\inf, x_{N+1} = +\inf)$ S jest wielomianem stopnia nie wyższego niż m,*
- *funkcja S i jej pochodne rzędu $1, 2, \ldots, m-1$ są ciągłe na całej osi rzeczywistej, tzn. $S \in \mathcal{C}^{m-1}$.*

W najprostszym przypadku, dla $m = 1$, funkcja sklejana jest po prostu łamaną przechodzącą przez punkty x_i. Z definicji tej wynika również, że wielomiany są szczególnym przypadkiem funkcji sklejanych.

Definicja 1.2 *Funkcję sklejana stopnia $2m-1$ z węzłami $x_0, x_1, \ldots, x_N$ nazywamy* ***naturalną*** *funkcją sklejaną, jeśli w przedziałach $(-\inf, x_0)$ i $(x_N, +\inf)$ dana jest wielomianami $m-1$ stopnia (a nie $2m-1$ stopnia).*

Dalej będziemy zajmowali się funkcjami sklejanymi interpolującymi dane punkty (x_i, y_i), tzn. takimi, że

$$S(x_i) = y_i \quad \text{dla } i = 0, 1, \ldots, N. \tag{2}$$

W całej klasie funkcji sklejanych zadanie to nie jest wyznaczone jednoznacznie (co zobaczymy także w rozdziale 1.3). Wystarczy jednak ograniczyć się do naturalnych funkcji sklejanych, dla których jest prawdziwe poniższe twierdzenie.

Twierdzenie 1.1 *Jeżeli węzły x_i są różne dla $i = 0, 1, \ldots, N$, oraz $1 \leq m \leq N+1$, to dla dowolnych wartości y_i istnieje dokładnie jedna naturalna funkcja sklejana S interpolująca punkty (x_i, y_i), tzn. taka, że $S(x_i) = y_i$ dla $i = 0, 1, \ldots, N$.*

1.3 Algorytm wyznaczania interpolującej funkcji sklejanej

1.3.1 Algorytm „wprost”

Zajmijmy się algorytmem wyznaczania naturalnej interpolacyjnej funkcji sklejanej S stopnia 3. Najprostszą metodą jest przedstawienie szukanej funkcji w każdym z podprzedziałów jako wielomian stopnia 3. w postaci naturalnej:

$$S(x) = a_i + b_i t + c_i t^2 + d_i t^3 \tag{3}$$

gdzie $t = x - x_i$ dla $x \in [x_i, x_{i+1})$, $i = 0, 1, \ldots, N-1$. Oczywiście $a_i = S(x_i) = y_i$. Do wyznaczenia pozostaje zatem jeszcze $3N$ współczynników. Ponieważ S, S', S'' mają być ciągłe w węzłach $x_i (i = 1, 2, \ldots, N-1)$, otrzymamy stąd $3N-3$ równań (granica z lewej równa jest granicy prawej). Brakujące warunki uzyskamy z założenia, że S jest funkcja sklejana naturalną, czyli że drugie pochodne na zewnątrz przedziału $[a, b]$ się zerują.

Nie będziemy tutaj przedstawiać postaci tych równań. Pozostawiamy do sprawdzenia, że układ równań na współczynniki wielomianów składowych funkcji sklejanej jest układem trójdiagonalnym. Widać, że ta postać nie jest zbyt wygodna do wyznaczania wartości funkcji sklejanej w punkcie (musimy pamiętać dodatkowo trzy współczynniki). Przydaje się jednak ona do całkowania/różniczkowania numerycznego.

1.3.2 Wyznaczanie funkcji sklejanej przy znanych drugich pochodnych

Mamy zadaną funkcję w postaci stabularyzowanej $y_i = f(x_i), i = 0, 1, \ldots, N$. Przypomnijmy wzór na interpolację liniową na przedziale (x_j, x_{j+1}) w punkcie:

$$y = Ay_j + By_{j+1}, \tag{4}$$

gdzie

$$A \equiv \frac{x_{j+1} - x}{x_{j+1} - x_j}, \qquad B \equiv 1 - A = \frac{x - x_j}{x_{j+1} - x_j}. \tag{5}$$

Jest to szczególna postać wzoru interpolacyjnego Lagrange'a.

Załóżmy, że oprócz stabularyzowanych wartości funkcji $y_i = f(x_i)$ mamy także stabularyzowane wartości drugich pochodnych $y_i'' = f''(x_i)$. (Później podamy sposób wyznaczania tych wartości). Do równania (4) możemy dodać wielomian trzeciego stopnia, którego drugie pochodne w punktach x_j i x_{j+1} są równe odpowiednio y_j'' i y_{j+1}'', zaś wartości w tych punktach są równe zeru. Odpowiednikiem wzoru (4) jest

$$y = Ay_j + By_{j+1} + Cy_j'' + Dy_{j+1}'', \tag{6}$$

gdzie

$$\begin{aligned} A &\equiv \frac{x_{j+1} - x}{x_{j+1} - x_j}, & B &\equiv 1 - A = \frac{x - x_j}{x_{j+1} - x_j}, \\ C &\equiv \frac{1}{6}(A^3 - A)(x_{j+1} - x_j)^2, & D &\equiv \frac{1}{6}(B^3 - B)(x_{j+1} - x_j)^2. \end{aligned} \tag{7}$$

Łatwo zauważyć, że powyższy wzór daje dobre wartości w punkcie. Pozostawiamy do sprawdzenia, że daje dobre wartości drugich pochodnych.

1.3.3 Obliczanie drugich pochodnych z warunku ciągłości

Wyznaczona przez nas funkcja interpolująca jest ciągła i ma (z definicji) ciągłe drugie pochodne. Nie znamy jednak zazwyczaj drugich pochodnych funkcji. Możemy jednak wyznaczyć drugie pochodne interpolującej funkcji sklejanej przez narzucenie warunku ciągłości *pierwszych* pochodnych funkcji interpolującej.

Żądane równania uzyskujemy żądając by pochodna równania (6) obliczona w punkcie x_j na przedziale (x_j, x_{j+1}) była równa pochodnej w tym samym punkcie ale obliczona na przedziale (x_{j-1}, x_j). Po odpowiednich przekształceniach uzyskujemy równanie

$$\frac{x_j - x_{j-1}}{6} y_{j-1}'' + \frac{x_{j+1} - x_{j-1}}{3} y_j'' + \frac{x_{j+1} - x_j}{6} y_{j+1}'' = \frac{y_{j+1} - y_j}{x_{j+1} - x_j} - \frac{y_j - y_{j-1}}{x_j - x_{j-1}}. \tag{8}$$

Jest to $(N+1) - 2$ równań liniowych na $N+1$ zmiennych $y_i'', i = 0, 1, \ldots, N$. Brakujące równania (na y_0'' i y_N'') możemy uzyskać na jeden z poniższych sposobów:

- ustalić jedną lub obie y_0'' i y_N'' na zero, otrzymując tzw. *naturalne funkcje sklejane* (patrz definicja 1.2),
- ustalić jedną lub obie y_0'' i y_N'' na zadaną wartość,
- obliczyć jedną lub obie y_0'' i y_N'' na podstawie zadanej pierwszej pochodnej w odpowiednim punkcie
- obliczyć y_0'' i y_N'' narzucając cykliczne warunki brzegowe (czyli m. in. pierwsze pochodne w x_0 i x_N są sobie równe).

Wszystkie z powyższych warunków brzegowych z wyjątkiem ostatniego dają układ równań w postaci *trójdiagonalnej*, która jest szczególnie łatwa do rozwiązania. Ostatni warunek brzegowy daje układ równań w postaci prawie trójdiagonalnej.

1.3.4 Algorytm „dwustopniowy”

Wygodnie zatem zastosować następujący algorytm. Najpierw wyliczamy drugie pochodne funkcji sklejanych dla zadanych punktów, rozwiązując w tym celu układ równań (8). Następnie stosujemy wzory (6), (7) do obliczania wartości interpolującej funkcji sklejanej w punkcie.

W porównaniu z algorytmem „wprost” musimy zapamiętać mniej wartości (jedną na punkt a nie trzy na punkt danych). Algorytm ma też lepsze własności numeryczne (wyznaczanie współczynników wielomianu w postaci naturalnej jest często źle uwarunkowane). Nie daje on nam jednak równania (postaci analitycznej) funkcji sklejanej.

1.4 Zadanie

Napisać program dokonujący interpolację funkcjami sklejanymi trzeciego stopnia. Wymagane jest zastosowanie tylko *jednego* z opisanych algorytmów. Należy przetestować interpolację funkcjami sklejanymi wielomianów, funkcji będących kawałkami wielomianami (np. $|x|$), oraz jakiejś „trudnej” funkcji, np. $1/(1+x^2)$. Grafika wskazana.

Uwaga: funkcje sklejane *naturalne* nie nadają się do **ekstrapolacji**!

A Rozwiązywanie układu równań trójliniowych

Układ równań liniowych typu:

$$\begin{bmatrix} b_1 & c_1 & 0 & \dots & \dots & 0 \\ a_2 & b_2 & c_2 & 0 & \dots & \dots \\ 0 & a_3 & b_3 & c_3 & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 0 & a_{N-1} & b_{N-1} & c_{N-1} \\ 0 & \dots & \dots & 0 & a_N & c_N \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ \vdots \\ v_{N-1} \\ v_N \end{bmatrix} \tag{9}$$

nazywamy układem w postaci **trójliniowej**.

Taki układ można rozwiązać w bardzo prosty sposób poprzez kolejne odejmowanie wierszy w celu wyeliminowania kolejnych zmiennych. To znaczy: od drugiego wiersza należy odjąć pierwszy przemnożony przez $\frac{b_2}{a_1}$. Pozbywamy się w ten sposób zmiennej x_1 z drugiego równania. Powtarzając te czynność w ostatnim równaniu otrzymamy równanie na x_N. Następnie idąc do góry będziemy mogli obliczyć wszystkie x_i na podstawie znanej wartości x_{i+1}.

Wybór elementu głównego nie jest zazwyczaj potrzebny przy rozwiązywaniu układu równań trojliniowych. Zagadnienia które prowadzą do równań trójliniowych dają zazwyczaj układy *diagonalnie dominujące*, tzn. takie że $|b_j| > |a_j| + |c_j|$.

Poniżej podane są algorytmy rozwiązywania układu równań trójliniowych (być może niektóre z nich są równoważne):

Zmodyfikowany algorytm Gaussa

$$y_N = \beta_{N+1}, \qquad y_i = \alpha_{i+1} y_{i+1} + \beta_{i+1} \quad i = N-1, \dots, 1 \tag{10a}$$

gdzie

$$\alpha_2 = \frac{b_1}{c_1}, \qquad \alpha_{i+1} = \frac{b_i}{c_i - a_i\alpha_i} \quad i = 2, \ldots, N-1 \tag{10b}$$

$$\beta_2 = \frac{v_1}{c_1}, \qquad \beta_{i+1} = \frac{v_i - a_i\beta_i}{c_i - a_i\alpha_i} \quad i = 2, \ldots, N \tag{10c}$$

Algorytm Gaussa

$$\begin{aligned} & & u_1 &= b_1, & y_1 &= v_1, \\ l &= \frac{a_i}{u_{i-1}}, & u_i &= b_i - lc_i, & y_i &= v_i - ly_{i-1}, \end{aligned} \tag{11a}$$

dla $i = 2, 3, \ldots, N$, oraz

$$\begin{aligned} x_N &= \frac{y_N}{u_N}, \\ x_i &= \frac{y_i - c_i x_{i+1}}{u_i}, \end{aligned} \tag{11b}$$

dla $i = N-1, N-2, \ldots, 1$.

Algorytm Crouta

$$\begin{aligned} & & l &= b_1, & y_1 &= v_1/l, \\ u_i &= \frac{c_i}{l} & l &= b_{i+1} - u_i a_{i+1}, & y_i &= \frac{v_{i+1} - a_{i+1}y_i}{l}, \end{aligned} \tag{12a}$$

dla $i = 1, 2, \ldots, N-1$, oraz

$$\begin{aligned} x_N &= y_N, \\ x_i &= y_i - u_i x_{i+1}, \end{aligned} \tag{12b}$$

dla $i = N-1, N-2, \ldots, 1$.

Należy wybrać jeden z algorytmów, zaimplementować go (przekazując do procedury **wyłącznie** niezerowe elementy tablicy, ewentualnie wektory pomocnicze) i go przetestować (sprawdzić czy znalezione rozwiązanie spełnia podane równanie). Przygotował: Jakub Narębski