KSN — III FK — teoria 4
Sklejki kubiczne

Pozostajemy w tematyce związanej z zagadnieniem interpolacji.

W praktyce, choćby przy opracowaniu wyników pomiarów, spotykamy się często z koniecznością przeprowadzenia gładkiej krzywej najlepiej pasującej do otrzymanych w wyniku pomiarów punków $\{(x_i, y_i)\}$. Ekonomiści załatwiają ten problem łącząc kolejno te punkty ze sobą — a więc przybliżając zależność $y(x)$ funkcją liniową $f_i(x) = a_i x + b_i$ dla $x \in [x_i, x_{i+1}]$. Funkcje f_i, a więc i współczynniki a_i i b_i są oczywiście różne w różnych przedziałach, mamy cały ich zbiór $\{a_i, b_i\}$. Można pokusić się o wygładzenie tej „giełdowej" krzywej poprzez określenie w każdym z przedziałów funkcji kwadratowej spełniającej dodatkowo warunek ciągłości pierwszej pochodnej przy przejściu z przedziału $[x_{i-1}, x_i]$ do przedziału $[x_i, x_{i+1}]$ — a więc w x_i.

W przypadku, gdy w każdym z przedziałów wyznaczonym przez odcięte węzłów interpolacyjnych przybliżymy zależność $y(x)$ wielomianem trzeciego stopnia $f_i(x) = a_i x^3 + b_i x^2 + c_i x + d_i$, żądając *zszycia* w węzłach nie tylko wartości funkcji $f_i(x_i) = f_{i+1}(x_i)$ ale również jej pierwszej $f'_i(x_i) = f'_{i+1}(x_i)$ i drugiej $f''_i(x_i) = f''_{i+1}(x_i)$ pochodnej, to mówimy, że przez punkty $\{(x_i, y_i)\}$ poprowadziliśmy *sklejkę kubiczną* $s(x)$. Do jednoznacznego rozwiązania problemu w pierwszym $[x_1, x_2]$ i ostatnim $[x_{N-1}, x_N]$ przedziale interpolacji potrzeba wyspecyfikować *warunki brzegowe*: np. żądając zerowania się drugiej pochodnej sklejki dla $x = x_1$ i $x = x_N$: $s''(x_1) = s''(x_N) = 0$. Mówimy wtedy o *normalnej* sklejce kubicznej.

W bibliotece *Numerical Recipes* konstukcję sklejek kubicznych umożliwiają procedury `spline` i `splint`.

Procedura `spline` wyznacza drugie pochodne (`y2`) interpolowanej funkcji w węzłach i dla zadanych węzłów (`x, y`) i warunków brzegowych (`yp1, ypn`) powinna być wywoływana *tylko* raz.

```
SUBROUTINE spline(x,y,n,yp1,ypn,y2)
INTEGER n,NMAX
REAL yp1,ypn,x(n),y(n),y2(n)
PARAMETER (NMAX=500)
...
```

Wówczas procedurę `splint` wywołujemy żądaną ilość razy otrzymując wartość sklejki `y` dla konkretnej wartości `x`.

```
SUBROUTINE splint(xa,ya,y2a,n,x,y)
INTEGER n
REAL x,y,xa(n),y2a(n),ya(n)
...
```

Biblioteka *GNU Scientific Library (GSL)* realizuje interpolację sklejkami kubicznymi za pomocą funkcji zadeklarowanych w pliku nagłówkowym `gsl_spline.h`.
Do wyliczenia wartości sklejki w żądanym punkcie `x` używa się funkcji

```
double gsl_spline_eval (const gsl_spline * spline, double x, gsl_interp_accel * acc)
```

Podobnie jak w przypadku interpolacji wielomianowej należy wcześniej przygotować obiekty pomocnicze:

- `spline` – wskaźnik do obiektu interpolacyjnego zaalokowanego przez funkcję
  `gsl_spline * gsl_spline_alloc (const gsl_interp_type * T, size_t size)`
  i zainicjowanego wywołaniem
  `int gsl_spline_init (gsl_spline * spline, const double xa[], const double ya[], size_t size)`.
  Jako typ interpolacji `T` można podać m.in. `gsl_interp_cspline` (sklejka normalna) i `gsl_interp_cspline_periodic` (periodyczne warunki brzegowe).

- `acc` – wskaźnik do obiektu zaalokowanego funkcją `gsl_interp_accel * gsl_interp_accel_alloc (void)`.

Funkcji alokujących pomocnicze obiekty używamy tylko raz przed późniejszym (być może wielokrotnym) użyciem fukcji `gsl_spline_eval`.

Po zakończeniu obliczeń zwalniamy zarezerwowaną wcześniej pamięć – do tego celu przeznaczone są funkcje `gsl_spline_free` oraz `gsl_interp_accel_free`.

Krzysztof Malarz & Maciej Wołoszyn, Kraków, 5 listopada 2003