Zadanie domowe 1

Należy znaleźć funkcję $u\colon [0,1]^2 \to \mathbb{R}$ spełniającą
równanie różniczkowe Poissona

$$-\frac{\partial^2 u(x,y)}{\partial x^2} - \frac{\partial^2 u(x,y)}{\partial y^2} = f(x,y),$$

w którym występuje dana funkcja ciągła $f\colon [0,1]^2 \to \mathbb{R}$ i taką, która na brzegu kwadratu $[0,1]^2$ ma wartość 0.

To zadanie jest dobrze postawione, tj. ma jednoznaczne rozwiązanie, które zależy od funkcji f w sposób ciągły. Nie za się go w ogólności rozwiązać metodami algebry, ale metody numeryczne umożliwiają znalezienie przybliżenia rozwiązania z dużą dokładnością.

Dyskretyzacja równania polega na założeniu, że funkcja u_h, tj. poszukiwane przybliżenie rozwiązania, ma szczególną postać, zależną od skończenie wielu zmiennych, i na ułożeniu układu równań, którego rozwiązaniem jest wektor tych zmiennych.

W metodzie różnic w kwadracie $[0,1]$ wprowadzimy regularną siatkę punktów. Wybieramy liczbę naturalną $N > 1$ i określamy punkty (x_i, y_j), gdzie $x_i = i/N$, $y_j = j/N$ dla $i, j = 0, \ldots, N$. Jeśli zatem i lub j jest równe 0 lub N, to punkt (x_i, y_j) leży na brzegu kwadratu.

Oznaczamy $h = 1/N$, zatem $x_i = ih$, $y_j = jh$. Poszukiwana funkcja u_h jest określona tylko w punktach (x_i, y_j): jej wartości w tych punktach oznaczamy u_{ij}. Dla punktu wewnątrz kwadratu $[0,1]$ sumę pochodnych drugiego rzędu funkcji u zastąpimy wyrażeniem

$$\frac{1}{h^2}(-u_{i-1,j} - u_{i,j-1} + 4u_{ij} - u_{i+1,j} - u_{i,j+1}).$$

W ten sposób otrzymamy układ $(N-1)^2$ równań liniowych

$$\frac{1}{h^2}(-u_{i-1,j} - u_{i,j-1} + 4u_{ij} - u_{i+1,j} - u_{i,j+1}) = f_{ij}.$$

w których $f_{ij} = f(u_i, y_j)$.

Macierz tego układu składa się z $(N-1)^2$ bloków o wymiarach $(N-1) \times (N-1)$, przy czym ma ona strukturę <u>blokowo-trójdiagonalną</u>:

$$A = \frac{1}{h^2} \begin{bmatrix} T & -I & 0 & \cdots & 0 \\ -I & T & -I & \ddots & 0 \\ 0 & -I & T & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -I \\ 0 & \cdots & 0 & -I & T \end{bmatrix}.$$

Każdy blok I jest macierzą jednostkową $(N-1) \times (N-1)$. Blok T jest macierzą trójdiagonalną:

$$T = \begin{bmatrix} 4 & -1 & 0 & \cdots & 0 \\ -1 & 4 & -1 & \ddots & 0 \\ 0 & -1 & 4 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 4 \end{bmatrix}.$$

Macierz A jest symetryczna i dodatnio określona, a więc jest nieosobliwa, choć jej wskaźnik uwarunkowania jest proporcjonalny do $1/h^2$, czyli dla dużych N jest bardzo duży.

Macierz A jest zatem <u>rzadka</u>: w żadnym wierszu nie ma ona więcej niż 5 współczynników różnych od zera. Dla N rzędu kilkadziesiąt lub więcej nie należy jej reprezentować za pomocą pełnej tablicy współczynników. Są dwie sensowne alternatywy: można użyć reprezentacji w postaci wykazu niezerowych współczynników, jaką w pakiecie Matlab albo Octave tworzy funkcja sparse, albo można napisać funkcję (skrypt Octave'a), która dla dowolnego wektora $\boldsymbol{x} \in \mathbb{R}^{(N-1)^2}$ obliczy iloczyn $A\boldsymbol{x}$, generując potrzebne współczynniki „w locie".

Pewne metody iteracyjne rozwiązywania układów równań liniowych wymagają mnożenia macierzy układu przez kolejne wektory, aby wytworzyć ciąg zbieżny do rozwiązania układu. Takie są metody Richardsona i sprzężonych gradientów. Inne metody, np. Jacobiego lub Gaussa–Seidela, wymagają mnożenia wektora przez macierze otrzymane z rozłożenia macierzy A na składniki i rozwiązywania równań z macierzami będącymi tymi składnikami — według opisu na wykładzie.

Rozwiązanie zadania polega na napisaniu i uruchomieniu skryptu (lub zestawu skryptów) pakietu Octave, który znajduje opisanym wyżej sposobem tablicę wartości funkcji u_h, przy użyciu jednej z metod iteracyjnych rozwiązywania układów równań liniowych (polecam metodę sprzężonych gradientów). Liczba N ma być parametrem do wyboru.
Za funkcję f można przyjąć funkcję stałą lub wielomian niskiego stopnia zmiennych x, y, np. $f(x, y) = x$ itp. W kolejnych iteracjach trzeba wypisywać normę wektora residuum, $\mathbf{b} - A\mathbf{x}_k$, gdzie $\mathbf{b}$ jest wektorem prawej strony (jego współrzędne są wartościami funkcji f), a $\mathbf{x}_k$ jest kolejnym przybliżeniem rozwiązania (wektora wartości funkcji u_h).
Wisienką na torcie może być wykonanie wykresu rozwiązania, przy użyciu funkcji `mesh` lub `surf`.

Zadanie domowe 2

Dany jest niemalejący ciąg liczb (węzłów) $u_0, \ldots, u_N$, gdzie $N > 2n$. Unormowane unkcje B-sklejane stopnia n z tymi węzłami, oznaczane symbolem N_i^n, mają kilka równoważnych definicji, w tym definicję opartą na wzorze rekurencyjnym Mansfielda–de Boora–Coxa:

$$N_i^0(x) = \begin{cases} 1 & \text{dla } x \in [u_i, u_{i+1}), \\ 0 & \text{w przeciwnym razie,} \end{cases}$$

$$N_i^n(x) = \frac{x - u_i}{u_{i+n} - u_i} N_i^{n-1}(x) + \frac{u_{i+n+1} - x}{u_{i+n+1} - u_{i+1}} N_{i+1}^{n-1}(x) \quad \text{dla } n > 0,$$

przy czym jeśli $u_i = \cdots = u_{i+n+1}$, to funkcja N_i^n jest funkcją zerową. Funkcja N_i^n jest równa 0 poza przedziałem $[u_i, u_{i+n+1})$. W węźle u_k, takim że $u_{k-1} < u_k = \cdots = u_{k+r-1} < u_{k+r}$ funkcje te mają ciągłe pochodne rzędu $n - r$. Niezerowe funkcje B-sklejane stopnia n są liniowo niezależne, a ich suma w każdym punkcie przedziału $[u_n, u_{N-n})$ jest równa 1.

Na podanym wyżej wzorze opierają się dwa algorytmy de Boora, z których pierwszy służy do obliczania wartości funkcji B-sklejanych w danym punkcie:

```
/* x ∈ [u_k, u_{k+1}) ⊂ [u_n, u_{N−n})  */
b[k] = 1;                                  /* N_k^0 = 1  */
for ( j = 1; j ⩽ n; j++ ) {
  β = (u_{k+1} − x)/(u_{k+1} − u_{k−j+1});       /* β = β_{k−j+1}^{(j)}  */
  b[k − j] = β * b[k − j + 1];               /* N_{k−j}^j = βN_{k−j+1}^{j−1}  */
  for ( i = k − j + 1; i < k; i++ ) {
    α = 1 − β;                             /* α = α_i^{(j)}  */
    β = (u_{i+j+1} − x)/(u_{i+j+1} − u_{i+1});  /* β = β_{i+1}^{(j)}  */
    b[i] = α * b[i] + β * b[i + 1];        /* N_i^j = αN_i^{j−1} + βN_{i+1}^{j−1}  */
  }
  b[k] *= (1 − β);                         /* N_k^j = αN_k^{j−1}  */
}
/* b[i] = N_i^n(x) dla i = k − n, ..., k,  */
```

a drugi umożliwia obliczenie wartości funkcji

$$s(x) = \sum_{i=0}^{N-n-1} d_i N_i^n(x)$$

dla $x \in [u_n, u_{N-n})$:

```
/* d_i^(0) = d_i  dla  i = k − n, ..., k,  x ∈ [u_k, u_{k+1}) ⊂ [u_n, u_{N−n})  */
for ( j = 1; j ⩽ n; j++ )
  for ( i = k − n + j; i ⩽ k; i++ ) {
    α = (x − u_i)/(u_{i+n+1−j} − u_i);      /* α = α_i^(n+1−j)  */
    d_i^(j) = (1 − α) * d_{i−1}^(j−1) + α * d_i^(j−1);
  }
/* d_k^(n) = s(x)  */
```

Powyższe wzory i algorytmy są znacznie prostsze w szczególnym przypadku, gdy węzły są równoodległe, tj. $u_i = a + (i - n)h$, gdzie $h = (b - a)/(N - 2n)$; ten wzór jest dobrany tak, aby podzielić przedział $[a, b]$, w którym suma funkcji B-sklejanych jest równa 1, na podprzedziały o jednakowej długości h. Zadaniem pomocniczym jest więc napisanie (na papierze) tych uproszczonych wzorów i algorytmów.

Teraz zadanie do zaprogramowania: niech $n = 3$ i niech $\hat{h} = (b - a)/M$ dla $M = c(N - 2n)$; N ma być rzędu kilkanaście, a c będzie niewielką liczbą naturalną (od 2 do 10). Dane są wartości y_j pewnej funkcji f w punktach $x_j = a + j\hat{h}$ dla $j = 0, \ldots, M$. Należy znaleźć funkcję sklejaną s, dla której wyrażenie

$$E = \sum_{j=0}^{M} \big(s(x_j) - y_j\big)^2$$

ma najmniejszą wartość. Dokładniej, trzeba znaleźć współczynniki d_i tej funkcji $s(x) = \sum_{i=0}^{N-4} d_i N_i^3(x)$. Można to zrobić, rozwiązując liniowe zadanie najmniejszych kwadratów dla układu równań

$$\sum_{i=0}^{N-4} a_{ji} d_i = y_j, \quad j = 0, \ldots, M,$$

w którym $a_{ji} = N_i^3(x_j)$; w tym celu trzeba obliczyć te współczynniki i utworzyć z nich macierz.

Uwaga 1: To jest macierz rzadka, a dokładniej wstęgowa: w każdym jej wierszu są tylko 3 lub 4 niezerowe współczynniki. Ponadto każde kolejne c wierszy można otrzymać, przesuwając poprzednie c wierszy o jedną pozycję w prawo (i przenosząc zera z końca na początek). Dopuszczalne jest reprezentowanie tej macierzy (o wymiarach $(M+1) \times (N-3)$) jako macierzy pełnej, ale preferowane jest użycie oszczędniejszej reprezentacji.

Uwaga 2: Jeśli krotność węzła nie jest większa niż n, to w otoczeniu tego węzła funkcje B-sklejane stopnia n są ciągłe. W szczególności ciągłe są funkcje trzeciego stopnia z węzłami równoodległymi (bo każdy węzeł w takim ciągu ma krotność 1). Dlatego w punkcie $x_M = u_{N-3} = b$ suma wartości funkcji $N^3_{N-6}, N^3_{N-5}, N^3_{N-4}$ jest równa 1 i można obliczać wartość funkcji s, biorąc $k = N-4$.

Liczby a, b, c i N mają być parametrami w implementacji algorytmu rozwiązującego zadanie. Liczby y_j można otrzymać przez stablicowanie dowolnej funkcji danej jakimś wzorem, przy czym należy przetestować algorytm na funkcjach niegładkich, na przykład $f(x) = |x - (a+b)/2|$. Należy wykonać wykres przedstawiający otrzymaną funkcję s, z zaznaczeniem punktów (x_j, y_j).

Zadanie domowe 3

Krzywa parametryczna jest to zbiór wektorów (albo punktów) $S = \{ \mathbf{s}(t) \colon t \in [a, b] \}$ określony za pomocą pewnej funkcji wektorowej $\mathbf{s}$, zwanej parametryzacją. Jeśli ta funkcja ma (prawie wszędzie) ciągłą pochodną, to długość krzywej można obliczyć na podstawie wzoru

$$\ell(S) = \int_a^b \|\mathbf{s}'(t)\|_2 \, dt.$$

Krzywa B-sklejana stopnia n jest określona za pomocą parametryzacji

$$\mathbf{s}(t) = \sum_{i=0}^{N-n-1} \mathbf{d}_i N_i^n(t), \quad t \in [a, b),$$

przy użyciu funkcji B-sklejanych stopnia n określonych przez niemalejący ciąg węzłów $u_0, \dots, u_N$, przy czym $u_n = a$, $u_{N-n} = b$. Wektorowe współczynniki $\mathbf{d}_i$ są nazywane punktami kontrolnymi krzywej.

Pochodna krzywej sklejanej stopnia 3 jest sklejaną funkcją wektorową stopnia 2. Należy zatem obliczyć całkę z funkcji, która jest pierwiastkiem kwadratowym z funkcji sklejanej stopnia 4. Takich całek nie można obliczać za pomocą wzorów analitycznych (które nie istnieją), pozostaje jedynie obliczenie numeryczne przy użyciu pewnej kwadratury.

Zadanie polega na napisaniu skryptu, który na podstawie podanego w pliku ciągu węzłów i punktów kontrolnych (mających dwie lub trzy współrzędne) obliczy długość krzywej trzeciego stopnia. Można też przyjąć, że węzły są równoodległe, tj. $u_i = i$ dla każdego i.

Użyta kwadratura powinna być złożona, tj. powinna być sumą kwadratur przybliżających całki w przedziałach $[u_n, u_{n+1}), \dots, [u_{N-n-1}, u_{N-n})$, w których funkcja podcałkowa jest gładka. Można założyć, że parametryzacja ma ciągłą pochodną i użyć kwadratur złożonych Simpsona lub Gaussa–Legendre'a czwartego rzędu. Do obliczania pochodnej parametryzacji można użyć wzoru

$$\mathbf{s}'(t) = \sum_{i=0}^{N-n-1} \mathbf{d}_i \big(N_i^n(t)\big)', \quad t \in [a, b),$$

przy czym

$$\frac{d}{dt}N_i^n(t) = \frac{n}{u_{i+n} - u_i}N_i^{n-1}(t) - \frac{n}{u_{i+n+1} - u_{i+1}}N_{i+1}^{n-1}(t).$$

Alternatywnie,

$$\mathbf{s}'(t) = \sum_{i=0}^{N-n-2} \frac{n}{u_{i+n+1} - u_{i+1}}(\mathbf{d}_{i+1} - \mathbf{d}_i)N_{i+1}^{n-1}(t).$$

Funkcje B-sklejane N_i^{n-1} w powyższych wzorach są określone przez ten sam ciąg węzłów $u_0, \ldots, u_N$, co funkcje N_i^n.

Dodatkowym wymaganiem w zadaniu jest dążenie do otrzymania wyniku możliwie małym kosztem, tzn. do ograniczenia liczby węzłów kwadratury. Dla zadanej tolerancji ε (np. rzędu 10^{-6}) można porównać wyniki otrzymane za pomocą kwadratur złożonych z przedziałami o długościach h i $h/2$ i wybrać takie h, aby różnica kwadratur była mniejsza niż ε (w pewnych przypadkach wartość bezwzględna tej różnicy jest oszacowaniem błędu „dokładniejszej” kwadratury).