Rezolvare Set 1

Mursa Ovidiu-Vlad

2 Iunie 2025

Problema 1

(a) Ortogonalitatea polinoamelor $\pi_k^+(t^2)$

Polinoamele Legendre monice $(\pi_k)_{k\in\mathbb{N}}$ sunt ortogonale pe intervalul [-1,1] în raport cu ponderea $w(t) = 1$. Adică:

$$\int_{-1}^{1} \pi_m(t)\pi_l(t)dt = c_m\delta_{ml}, \quad c_m > 0$$

Definim $\pi_k^+(x) = \pi_{2k}(\sqrt{x})$ pentru $x \geq 0$. Enunțul folosește notația $\pi_k^+(t^2) = \pi_{2k}(t)$, ceea ce înseamnă că argumentul lui π_k^+ este t^2. Vom arăta că polinoamele $\pi_k^+(x)$ sunt ortogonale monice pe [0, 1] în raport cu ponderea $w(x) = \frac{1}{\sqrt{x}}$. Considerăm integrala:

$$I = \int_0^1 \frac{1}{\sqrt{x}}\pi_k^+(x)\pi_j^+(x)dx$$

Folosind definiția din enunț, $x = t^2$, deci $\pi_k^+(t^2) = \pi_{2k}(t)$. Cu substituția $x = t^2$, $dx = 2tdt$. Pentru $x \in [0,1]$, alegem $t \in [0,1]$.

$$I = \int_0^1 \frac{1}{\sqrt{t^2}}\pi_{2k}(t)\pi_{2j}(t)(2tdt) = \int_0^1 \frac{1}{t}\pi_{2k}(t)\pi_{2j}(t)(2tdt) = 2\int_0^1 \pi_{2k}(t)\pi_{2j}(t)dt$$

Deoarece $\pi_{2k}(t)$ și $\pi_{2j}(t)$ sunt funcții pare (polinoamele Legendre de grad par conțin doar puteri pare ale lui t), produsul lor este tot o funcție pară. Astfel:

$$2\int_0^1 \pi_{2k}(t)\pi_{2j}(t)dt = \int_{-1}^1 \pi_{2k}(t)\pi_{2j}(t)dt$$

Din ortogonalitatea polinoamelor Legendre pe [-1,1] cu ponderea $w(t) = 1$:

$$I = c_{2k}\delta_{kj}$$

unde $c_{2k} = \int_{-1}^1 (\pi_{2k}(t))^2 dt > 0$. Polinoamele $\pi_{2k}(t)$ sunt monice. Fie $\pi_{2k}(t) = t^{2k} + \dots$. Atunci $\pi_k^+(x) = \pi_{2k}(\sqrt{x}) = (\sqrt{x})^{2k} + \dots = x^k + \dots$. Deci, $\pi_k^+(x)$ sunt polinoame monice de grad k în x. Prin urmare, polinoamele $\pi_k^+(t^2)$ (interpretate ca $\pi_k^+(x)$ cu $x = t^2$) sunt ortogonale monice pe [0, 1] în raport cu ponderea $w(t) = \frac{1}{\sqrt{t}}$ (sau $w(x) = \frac{1}{\sqrt{x}}$).

(b) Stabilirea formulei de cuadratură

Dorim să stabilim formula:

$$\int_0^1 \frac{1}{\sqrt{x}}f(x)dx = 2\sum_{k=1}^{n} A_k f(t_k^2) + R_n(f)$$

unde A_k și t_k sunt coeficienții și, respectiv, nodurile formulei de cuadratură Gauss-Legendre cu 2n noduri pe [-1,1]. Pornim de la integrala din stânga și facem substituția $x = u^2$, $dx = 2udu$. Pentru $x \in [0,1]$ luăm $u \in [0,1]$.

$$\int_0^1 \frac{1}{\sqrt{x}}f(x)dx = \int_0^1 \frac{1}{\sqrt{u^2}}f(u^2)(2udu) = \int_0^1 \frac{1}{u}f(u^2)(2udu) = 2\int_0^1 f(u^2)du$$

Deoarece $f(u^2)$ este o funcție pară de u:

$$2\int_0^1 f(u^2)du = \int_{-1}^1 f(u^2)du$$

Aplicăm formula de cuadratură Gauss-Legendre cu 2n noduri $(t_i^*, A_i^*)_{i=1}^{2n}$ pe intervalul $[-1,1]$ pentru funcția $g(u) = f(u^2)$:

$$\int_{-1}^1 f(u^2)du = \sum_{i=1}^{2n} A_i^* f((t_i^*)^2) + \tilde{R}_{2n}(f(u^2))$$

Nodurile t_i^* sunt rădăcinile polinomului Legendre $P_{2n}(t)$ și sunt simetrice față de origine. Adică, dacă t^* este nod, atunci și $-t^*$ este nod, și au același coeficient A^*. Fie $t_1, \ldots, t_n$ nodurile pozitive din setul $\{t_i^*\}_{i=1}^{2n}$, și $A_1, \ldots, A_n$ coeficienții corespunzători. Suma devine:

$$\sum_{i=1}^{2n} A_i^* f((t_i^*)^2) = \sum_{k=1}^{n} A_k f(t_k^2) + \sum_{k=1}^{n} A_k f((-t_k)^2)$$

Deoarece $(-t_k)^2 = t_k^2$ avem:

$$\sum_{i=1}^{2n} A_i^* f((t_i^*)^2) = \sum_{k=1}^{n} A_k f(t_k^2) + \sum_{k=1}^{n} A_k f(t_k^2) = 2\sum_{k=1}^{n} A_k f(t_k^2)$$

Aici, $(t_k, A_k)_{k=1}^n$ sunt perechile nod (pozitiv) coeficient din formula Gauss-Legendre cu 2n noduri. Enunțul folosește $(t_k, A_k)_{k=1}^{2n}$ pentru toate nodurile, iar apoi suma $\sum_{k=1}^n A_k f(t_k^2)$ implică că t_k din sumă sunt cele n perechi corespunzând nodurilor pozitive. Astfel, obținem formula cerută:

$$\int_0^1 \frac{1}{\sqrt{x}} f(x)dx = 2\sum_{k=1}^{n} A_k f(t_k^2) + R_n(f)$$

(c) Implementarea formulei de cuadratură în MATLAB

Funcția MATLAB de mai jos implementează această formulă de cuadratură. Se folosește o funcție ajutătoare 'get_gauss_legendre_coeffs(N)' care returnează cele N noduri și ponderi Gauss-Legendre pe [-1,1].

```matlab
function I = gauss_sqrt_inv_f(func, n_gauss_formula)
% Calculeaza integrala int_0^1 (1/sqrt(x)) f(x) dx
% folosind formula derivata.
% func: handle-ul functiei f(x)
% n_gauss_formula: numarul de noduri pozitive 'n' din formula (2n noduri GL)

num_gl_nodes = 2*n_gauss_formula;
[nodes_gl_full, weights_gl_full] = get_gauss_legendre_coeffs(num_gl_nodes);

% nodes_gl_full sunt pe [-1,1], sortate crescator
% weights_gl_full sunt ponderile corespunzatoare

% Selectam nodurile pozitive si ponderile corespunzatoare
% Nodurile pozitive sunt ultimele n_gauss_formula noduri
tk_positive = nodes_gl_full(n_gauss_formula+1:num_gl_nodes);
Ak_corresponding = weights_gl_full(n_gauss_formula+1:num_gl_nodes);

sum_val = 0;
for k = 1:n_gauss_formula
    sum_val = sum_val + Ak_corresponding(k) * func(tk_positive(k)^2);
end

I = 2 * sum_val;
end
```

Listing 1: Funcția principală de cuadratură

```
function [nodes, weights] = get_gauss_legendre_coeffs(N)
% Implementare a algoritmului Golub-Welsch pentru
% noduri si ponderi Gauss-Legendre pe [-1,1].
if N == 0
    nodes = [];
    weights = [];
    return;
end
beta = (1:N-1)./sqrt(4*(1:N-1).^2-1);
J = diag(beta,1) + diag(beta,-1);
[V,D] = eig(J);
nodes = diag(D);
[nodes,i] = sort(nodes); % Sorteaza nodurile
weights = 2*V(1,i)'.^2; % Ponderile corespunzatoare
end
```

Listing 2: Funcția ajutătoare pentru coeficienți Gauss-Legendre

(d) Calculul integralei $\int_0^1 \frac{\sin(x)}{\sqrt{x}}dx$ cu 8 zecimale exacte

Folosim funcția implementată pentru $f(x) = \sin(x)$. Căutăm o valoare 'n_gauss_formula' suficient de mare.

```
f_sin = @(x) sin(x);
tol = 1e-9; % toleranta pentru 8 zecimale exacte

integral_val_prev = 0;

fprintf('Calculul integralei int_0^1 sin(x)/sqrt(x) dx:\n');
for n_iter = 1:10 % 'n' din formula, deci 2*n_iter noduri GL
    integral_val_curr = gauss_sqrt_inv_f(f_sin, n_iter);
    if n_iter > 1
        diff_val = abs(integral_val_curr - integral_val_prev);
        fprintf('n = %d (2n=%d noduri GL), Integral = %.10f, Dif = %.2e\n', ...
            n_iter, 2*n_iter, integral_val_curr, diff_val);
        if diff_val < tol
            fprintf('Convergenta atinsa pentru n = %d.\n', n_iter);
            final_integral_value = integral_val_curr;
            break;
        end
    else
        fprintf('n = %d (2n=%d noduri GL), Integral = %.10f\n', ...
            n_iter, 2*n_iter, integral_val_curr);
    end
    final_integral_value = integral_val_curr; % in caz ca bucla e scurta sau nu se
        ↪ intrerupe
    integral_val_prev = integral_val_curr;
    if n_iter == 10
        fprintf('Convergenta nu a fost atinsa pana la n = 10.\n');
    end
end
fprintf('Valoarea finala calculata a integralei este: %.8f\n', final_integral_value);
```

Listing 3: Script pentru calculul integralei specificate

Rezultatul rulării codului MATLAB:

```
Calculul integralei int_0^1 sin(x)/sqrt(x) dx:
n=1 (2n=2 noduri GL), Integral = 0.6438538995
n=2 (2n=4 noduri GL), Integral = 0.6216153861, Dif = 2.22e-02
n=3 (2n=6 noduri GL), Integral = 0.6205663613, Dif = 1.05e-03
n=4 (2n=8 noduri GL), Integral = 0.6205378287, Dif = 2.85e-05
n=5 (2n=10 noduri GL), Integral = 0.6205366198, Dif = 1.21e-06
n=6 (2n=12 noduri GL), Integral = 0.6205366021, Dif = 1.77e-08
Convergenta atinsa pentru n = 6.
Valoarea finala calculata a integralei este: 0.62053660
```

Problema 2

(a) Deducerea metodei Newton pentru $\frac{1}{\sqrt{a}}$

Dorim să calculăm $x = \frac{1}{\sqrt{a}}$. Aceasta este echivalent cu $x^2 = \frac{1}{a}$ sau $\frac{1}{x^2} = a$. Considerăm funcția $f(x) = \frac{1}{x^2} - a$. Rădăcina acestei funcții este $x = \frac{1}{\sqrt{a}}$. Derivata funcției este $f'(x) = -2x^{-3} = -\frac{2}{x^3}$. Iterația Newton este dată de formula $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$.

$$x_{k+1} = x_k - \frac{\frac{1}{x_k^2} - a}{-\frac{2}{x_k^3}} = x_k + \left(\frac{1}{x_k^2} - a\right)\frac{x_k^3}{2}$$

$$x_{k+1} = x_k + \frac{x_k}{2} - \frac{ax_k^3}{2} = \frac{3x_k}{2} - \frac{ax_k^3}{2}$$

Deci, formula de iterație Newton este:

$$x_{k+1} = x_k\left(\frac{3}{2} - \frac{ax_k^2}{2}\right) = \frac{x_k}{2}(3 - ax_k^2)$$

(b) Condiții de convergență pentru x_0

Pentru a studia convergența, analizăm funcția de iterație $g(x) = \frac{x}{2}(3 - ax^2)$. Derivata sa este $g'(x) = \frac{1}{2}(3 - ax^2) + \frac{x}{2}(-2ax) = \frac{3}{2} - \frac{ax^2}{2} - ax^2 = \frac{3}{2} - \frac{3ax^2}{2}$. La rădăcina $\alpha = 1/\sqrt{a}$, avem $a\alpha^2 = a(1/a) = 1$. Astfel, $g'(\alpha) = \frac{3}{2} - \frac{3\cdot 1}{2} = 0$. Deoarece $g'(\alpha) = 0$, metoda converge cel puțin pătratic (de fapt, este cubică pentru această formulare specifică) dacă x_0 este suficient de aproape de α. Pentru ca iteratul x_{k+1} să rămână pozitiv (presupunând $x_k > 0$), trebuie ca $3 - ax_k^2 > 0$. Aceasta implică $ax_k^2 < 3$, deci $x_k^2 < \frac{3}{a}$. Deoarece x_k trebuie să fie pozitiv pentru a aproxima $1/\sqrt{a}$ (cu $a > 0$), condiția este $0 < x_k < \sqrt{\frac{3}{a}}$. Dacă $x_0 \in (0, \sqrt{\frac{3}{a}})$ atunci șirul (x_k) converge la $1/\sqrt{a}$. Dacă $x_0 = \sqrt{3/a}$, atunci $x_1 = \frac{\sqrt{3/a}}{2}(3 - a(3/a)) = 0$, iar x_2 este nedefinit deoarece $f'(0)$ ar fi implicat în pasul următor. Dacă $x_0 \searrow 0$ atunci $x_1 \approx \frac{3}{2}x_0$, deci șirul crește inițial. Dacă $x_0 \nearrow \sqrt{3/a}$, atunci $3 - ax_0^2 \searrow 0$ deci $x_1 \searrow 0$. Intervalul de convergență este $x_0 \in (0, \sqrt{3/a})$.

(c) Metodă de calcul al radicalului fără împărțiri

Iterația pentru $y = 1/\sqrt{a}$ este $y_{k+1} = \frac{y_k}{2}(3 - ay_k^2)$. Această formulă nu conține operații de împărțire (considerăm că împărțirea cu 2 este un shift binar sau o înmulțire cu 0.5). Pentru a calcula $\sqrt{a}$, putem folosi relația $\sqrt{a} = a \cdot \frac{1}{\sqrt{a}}$. Algoritmul este:

1. Se alege o aproximație inițială $y_0 \in (0, \sqrt{3/a})$ pentru $1/\sqrt{a}$.
2. Se iterează $y_{k+1} = \frac{y_k}{2}(3 - ay_k^2)$ până la convergență (fie y_N valoarea finală).
3. Se calculează $\sqrt{a} \approx a \cdot y_N$.

Această metodă necesită doar înmulțiri, adunări/scăderi și o înmulțire finală.