Rezolvare - Set 2

Mursa Ovidiu-Vlad

2 Iunie 2025

Problema 3

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

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

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

Ni se dau polinoamele $\pi_k^-(t^2) = \frac{\pi_{2k+1}(t)}{t}$. Trebuie să arătăm că acestea, interpretate ca polinoame în $x = t^2$, sunt ortogonale monice pe [0, 1] în raport cu ponderea $w(x) = \sqrt{x}$. Fie $\pi_k^-(x)$ aceste polinoame ca funcție de x. Considerăm integrala:

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

Facem substituția $x = t^2$, unde $t \geq 0$. Atunci $dx = 2tdt$. Pentru $x \in [0, 1]$, avem $t \in [0, 1]$. Folosind definiția dată, $\pi_k^-(t^2) = \frac{\pi_{2k+1}(t)}{t}$.

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

Pentru $t \in [0, 1]$, $\sqrt{t^2} = t$.

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

Polinoamele Legendre $\pi_m(t)$ au paritatea lui m. Deci $\pi_{2k+1}(t)$ și $\pi_{2j+1}(t)$ sunt funcții impare. Produsul a două funcții impare este o funcție pară. Astfel, $\pi_{2k+1}(t)\pi_{2j+1}(t)$ este o funcție pară. Prin urmare,

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

Din proprietatea de ortogonalitate a polinoamelor Legendre monice pe [-1,1] cu ponderea $w(t) = 1$:

$$I = c_{2k+1}\delta_{(2k+1)(2j+1)}$$

Deoarece indicii polinoamelor originale sunt $2k + 1$ și $2j + 1$, ortogonalitatea este valabilă dacă $2k + 1 \neq 2j + 1$, ceea ce este echivalent cu $k \neq j$. Deci $I = \tilde{c}_k\delta_{kj}$ pentru o constantă $\tilde{c}_k > 0$.

Pentru a arăta că sunt monice: $\pi_m(t)$ este un polinom monic de grad m. $\pi_{2k+1}(t) = t^{2k+1} + a_{2k}t^{2k} + \cdots + a_1t$ (polinoamele Legendre de grad impar sunt funcții impare, deci conțin doar puteri impare ale lui t, iar termenul constant este zero). Atunci $\frac{\pi_{2k+1}(t)}{t} = t^{2k} + a_{2k}t^{2k-1} + \cdots + a_2t^2 + a_1$. (Nota: $\pi_{2k+1}(t)$ este monic, deci a_{2k} nu este neapărat coeficientul principal al $\pi_{2k}(t)$, ci coeficientul puterii t^{2k} în $\pi_{2k+1}(t)$.) De fapt, $\pi_{2k+1}(t) = t^{2k+1} + c_{2k-1}^{(2k+1)}t^{2k-1} + \cdots + c_1^{(2k+1)}t$. (Polinoamele Legendre conțin doar puteri de aceeași paritate cu gradul lor). Deci, $\frac{\pi_{2k+1}(t)}{t} = t^{2k} + c_{2k-1}^{(2k+1)}t^{2k-2} + \cdots + c_1^{(2k+1)}$. Acesta este un polinom în t^2. Fie $x = t^2$. Atunci $\pi_k^-(x) = x^k + c_{2k-1}^{(2k+1)}x^{k-1} + \cdots + c_1^{(2k+1)}$. Acesta este un polinom monic 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 $w(t) = \sqrt{t}$ (sau $w(x) = \sqrt{x}$).

(b) Stabilirea formulei de cuadratură

Dorim să stabilim formula:

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

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

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

Considerăm funcția $g(u) = u^2 f(u^2)$. Aceasta este o funcție pară de u, deoarece $(-u)^2 f((-u)^2) = u^2 f(u^2) = g(u)$. Prin urmare,

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

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

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

Nodurile t_i^* ale polinoamelor Legendre $P_{2n+1}(t)$ sunt simetrice față de origine. Un nod este $t_0^* = 0$. Celelalte $2n$ noduri sunt $\pm t_k$ pentru $k = 1, \ldots, n$, unde t_k sunt nodurile pozitive. Fie A_0^* ponderea pentru $t_0^* = 0$ și A_k ponderea pentru $\pm t_k$. Suma devine:

$$A_0^*(0)^2 f((0)^2) + \sum_{k=1}^{n} A_k (t_k)^2 f((t_k)^2) + \sum_{k=1}^{n} A_k (-t_k)^2 f((-t_k)^2)$$

Deoarece $(-t_k)^2 = t_k^2$, iar termenul corespunzător lui $t_0^* = 0$ este $A_0^* \cdot 0 \cdot f(0) = 0$.

$$\sum_{i=1}^{2n+1} A_i^* (t_i^*)^2 f((t_i^*)^2) = 0 + \sum_{k=1}^{n} A_k t_k^2 f(t_k^2) + \sum_{k=1}^{n} A_k t_k^2 f(t_k^2) = 2 \sum_{k=1}^{n} A_k t_k^2 f(t_k^2)$$

Aici, $(t_k, A_k)_{k=1}^n$ sunt perechile nod (pozitiv)-coeficient din formula Gauss-Legendre cu $2n+1$ noduri (există n noduri pozitive distincte). Astfel, obținem formula cerută:

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

(c) Implementarea formulei de cuadratură în MATLAB

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

```matlab
function I = integrare_gauss_radical_f(functie, n_formula_gauss)
% calculeaza integrala int_0^1 sqrt(x) f(x) dx
% folosind formula derivata.
% functie: handle-ul functiei f(x)
% n_formula_gauss: numarul 'n' de noduri pozitive din formula
% (formula gauss-legendre foloseste 2n+1 noduri in total)

numar_noduri_gl = 2*n_formula_gauss + 1;
[noduri_gl_complete, ponderi_gl_complete] =
    ↪ get_coeficienti_gauss_legendre(numar_noduri_gl);

% noduri_gl_complete sunt pe [-1,1], sortate crescator.
% ponderi_gl_complete sunt ponderile corespunzatoare.
```

```matlab

% nodurile sunt: t_(-n), ..., t_(-1), t_0, t_1, ..., t_n
% in vectorul sortat noduri_gl_complete:
% noduri_gl_complete(1:n_formula_gauss) sunt negative
% noduri_gl_complete(n_formula_gauss+1) este 0
% noduri_gl_complete(n_formula_gauss+2:2*n_formula_gauss+1) sunt pozitive

% selectam nodurile pozitive si ponderile corespunzatoare
tk_pozitive = noduri_gl_complete(n_formula_gauss+2:numar_noduri_gl);
Ak_corespunzatoare = ponderi_gl_complete(n_formula_gauss+2:numar_noduri_gl);

suma_val = 0;
for k = 1:n_formula_gauss % suma are n_formula_gauss termeni
    suma_val = suma_val + Ak_corespunzatoare(k) * (tk_pozitive(k)^2) * functie(tk_pozitive(k)^2);
end

I = 2 * suma_val;
end
```

Listing 1: Funcția principală de cuadratură pentru $\int_0^1 \sqrt{x} f(x) dx$

```matlab
function [noduri, ponderi] = get_coeficienti_gauss_legendre(N_noduri)
% implementare a algoritmului Golub-Welsch pentru
% noduri si ponderi Gauss-Legendre pe [-1,1].
if N_noduri == 0
    noduri = [];
    ponderi = [];
    return;
end
beta = (1:N_noduri-1)./sqrt(4*(1:N_noduri-1).^2-1);
J = diag(beta,1) + diag(beta,-1); % matricea jacobi
[V, D] = eig(J); % v = vectori proprii, d = valori proprii (nodurile)
noduri = diag(D);
[noduri, i] = sort(noduri); % sorteaza nodurile
ponderi = 2*V(1,i)'.^2; % ponderile corespunzatoare
end
```

Listing 2: Funcția ajutătoare pentru coeficienți Gauss-Legendre

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

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

```matlab
functie_sin_radical = @(x) sin(x); % functia f(x) din integrala
toleranta = 1e-9; % toleranta pentru 8 zecimale exacte (a noua zecimala sa nu influenteze a opta)

valoare_integrala_anterioara = 0;
valoare_integrala_finala = 0;

fprintf('Calculul integralei int_0^1 sqrt(x) * sin(x) dx:\n');
% 'n_iteratie' este 'n' din formula, deci 2*n_iteratie+1 noduri GL
for n_iteratie = 1:10
    valoare_integrala_curenta = integrare_gauss_radical_f(functie_sin_radical, n_iteratie);
    if n_iteratie > 1
        diferenta_valoare = abs(valoare_integrala_curenta - valoare_integrala_anterioara);
        fprintf('n = %d (2n+1=%d noduri GL), Integral = %.10f, Dif = %.2e\n', ...
            n_iteratie, 2*n_iteratie+1, valoare_integrala_curenta, diferenta_valoare);
        if diferenta_valoare < toleranta
            fprintf('Convergenta atinsa pentru n = %d.\n', n_iteratie);
            valoare_integrala_finala = valoare_integrala_curenta;
            break;
        end
    else
        fprintf('n = %d (2n+1=%d noduri GL), Integral = %.10f\n', ...
            n_iteratie, 2*n_iteratie+1, valoare_integrala_curenta);
    end
```

```
    valoare_integrala_finala = valoare_integrala_curenta; % actualizam in caz ca bucla
        ↪ se termina
    valoare_integrala_anterioara = valoare_integrala_curenta;
    if n_iteratie == 10
        fprintf('Convergenta nu a fost atinsa pana la n = 10.\n');
    end
end
fprintf('Valoarea finala calculata a integralei este: %.8f\n',
    ↪ valoare_integrala_finala);
```

Listing 3: Script pentru calculul integralei $\int_0^1 \sqrt{x}\sin(x)dx$

Rezultatul rulării codului MATLAB:

```
Calculul integralei int_0^1 sqrt(x)*sin(x) dx:
n = 1 (2n+1=3 noduri GL), Integral = 0.3764283156
n = 2 (2n+1=5 noduri GL), Integral = 0.3641591349, Dif = 1.23e-02
n = 3 (2n+1=7 noduri GL), Integral = 0.3642220621, Dif = 6.29e-05
n = 4 (2n+1=9 noduri GL), Integral = 0.3642219319, Dif = 1.30e-07
n = 5 (2n+1=11 noduri GL), Integral = 0.3642219320, Dif = 1.45e-10
Convergenta atinsa pentru n = 5.
Valoarea finala calculata a integralei este: 0.36422193
```

Problema 4

(a) Metoda Newton pentru $\alpha = \sqrt{A}$

Se consideră $A > 0$ și $\alpha = \sqrt{A}$.

Ecuația 1: $x^2 - A = 0$ Funcția este $f(x) = x^2 - A$. Derivata este $f'(x) = 2x$. Iterația Newton este $x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - A}{2x_k} = \frac{2x_k^2 - x_k^2 + A}{2x_k} = \frac{x_k^2 + A}{2x_k} = \frac{1}{2}\left(x_k + \frac{A}{x_k}\right)$. Fie $g(x) = \frac{1}{2}(x + \frac{A}{x})$. Derivata funcției de iterație este $g'(x) = \frac{1}{2}(1 - \frac{A}{x^2})$. La rădăcina $\alpha = \sqrt{A}$, $g'(\alpha) = \frac{1}{2}(1 - \frac{A}{(\sqrt{A})^2}) = \frac{1}{2}(1 - 1) = 0$. Deoarece $g'(\alpha) = 0$, convergența este cel puțin pătratică. Pentru $x_0 > 0$, toți $x_k > 0$. Conform inegalității mediilor aritmetică și geometrică, pentru $k \geq 0$, $x_{k+1} = \frac{1}{2}(x_k + \frac{A}{x_k}) \geq \sqrt{x_k \cdot \frac{A}{x_k}} = \sqrt{A}$. Deci $x_k \geq \sqrt{A}$ pentru $k \geq 1$. Dacă $x_k = \sqrt{A}$, atunci $x_{k+1} = \frac{1}{2}(\sqrt{A} + \frac{A}{\sqrt{A}}) = \sqrt{A}$. Dacă $x_k > \sqrt{A}$, atunci $x_k^2 > A$. $x_{k+1} - x_k = \frac{1}{2}(x_k + \frac{A}{x_k}) - x_k = \frac{1}{2}(\frac{A}{x_k} - x_k) = \frac{A - x_k^2}{2x_k}$. Dacă $x_k > \sqrt{A}$, atunci $x_k^2 > A$, deci $A - x_k^2 < 0$. Cum $x_k > 0$, $x_{k+1} - x_k < 0$. Deci, pentru $k \geq 1$, șirul (x_k) este descrescător și mărginit inferior de $\sqrt{A}$. Prin urmare, șirul converge la $\sqrt{A}$ pentru orice $x_0 > 0$.

Ecuația 2: $\frac{A}{x^2} - 1 = 0$ Funcția este $f(x) = \frac{A}{x^2} - 1$. Derivata este $f'(x) = -\frac{2A}{x^3}$. Iterația Newton este $x_{k+1} = x_k - \frac{A/x_k^2 - 1}{-2A/x_k^3} = x_k + \frac{x_k^3(A/x_k^2 - 1)}{2A} = x_k + \frac{x_k(A - x_k^2)}{2A}$. $x_{k+1} = \frac{2Ax_k + Ax_k - x_k^3}{2A} = \frac{3Ax_k - x_k^3}{2A} = x_k\left(\frac{3A - x_k^2}{2A}\right)$. Fie $g(x) = x\left(\frac{3A - x^2}{2A}\right)$. Derivata $g'(x) = \frac{3A - x^2}{2A} + x\left(\frac{-2x}{2A}\right) = \frac{3A - x^2 - 2x^2}{2A} = \frac{3A - 3x^2}{2A}$. La $\alpha = \sqrt{A}$, $g'(\alpha) = \frac{3A - 3(\sqrt{A})^2}{2A} = 0$. Convergență cel puțin pătratică. Pentru ca x_{k+1} să fie pozitiv (presupunând $x_k > 0$), trebuie ca $\frac{3A - x_k^2}{2A} > 0$. Deoarece $A > 0$, este necesar $3A - x_k^2 > 0$, adică $x_k^2 < 3A$. Deci $0 < x_k < \sqrt{3A}$. Dacă $x_0 = \sqrt{3A}$, $x_1 = \sqrt{3A}\left(\frac{3A - 3A}{2A}\right) = 0$. Atunci x_2 este nedefinit (numitorul $f'(0)$ este zero). Dacă $x_0 > \sqrt{3A}$, atunci $3A - x_0^2 < 0$, deci $x_1 < 0$. Șirul nu converge la $\sqrt{A}$ (care este pozitiv). Studiem convergența pentru $x_0 \in (0, \sqrt{3A})$. Fie $e_k = x_k - \sqrt{A}$. $e_{k+1} = x_{k+1} - \sqrt{A} = g(x_k) - g(\sqrt{A})$. $g''(x) = \frac{-6x}{2A} = -\frac{3x}{A}$. $g''(\sqrt{A}) = -\frac{3\sqrt{A}}{A} = -\frac{3}{\sqrt{A}}$. $e_{k+1} \approx \frac{g''(\sqrt{A})}{2}e_k^2 = -\frac{3}{2\sqrt{A}}e_k^2$. Dacă x_k este aproape de $\sqrt{A}$ și $x_k \neq \sqrt{A}$, atunci $e_k \neq 0$, deci $e_k^2 > 0$. Atunci $e_{k+1} < 0$, ceea ce înseamnă $x_{k+1} < \sqrt{A}$ pentru k suficient de mare (sau $k \geq 1$ dacă $x_0 \neq \sqrt{A}$). Dacă $0 < x_k < \sqrt{A}$: atunci $x_k^2 < A < 3A$. $3A - x_k^2 > 2A$. $x_{k+1} = x_k\left(\frac{3A - x_k^2}{2A}\right) > x_k\left(\frac{2A}{2A}\right) = x_k$. Șirul este crescător. Deoarece $x_{k+1} < \sqrt{A}$ (pentru $k \geq 1$) și x_k crește, șirul converge la $\sqrt{A}$. Dacă $\sqrt{A} < x_k < \sqrt{3A}$: atunci $A < x_k^2 < 3A$. $0 < 3A - x_k^2 < 2A$. $0 < \frac{3A - x_k^2}{2A} < 1$. Deci $0 < x_{k+1} < x_k$. Șirul este descrescător. Deoarece $x_{k+1} < \sqrt{A}$ (din $e_{k+1} < 0$), șirul converge la $\sqrt{A}$. Intervalul de convergență pentru $x_n > 0$ este $0 < x_0 < \sqrt{3A}$. Deci $b = \sqrt{3A}$.

(b) Metoda Newton pentru $\alpha = \sqrt[3]{A}$

Se consideră $A > 0$ și $\alpha = \sqrt[3]{A}$.

Ecuația 1: $x^3 - A = 0$ Funcția este $f(x) = x^3 - A$. Derivata $f'(x) = 3x^2$. Iterația Newton $x_{k+1} = x_k - \frac{x_k^3 - A}{3x_k^2} = \frac{3x_k^3 - x_k^3 + A}{3x_k^2} = \frac{2x_k^3 + A}{3x_k^2} = \frac{1}{3}\left(2x_k + \frac{A}{x_k^2}\right)$. Fie $g(x) = \frac{1}{3}(2x + \frac{A}{x^2})$. Derivata $g'(x) = \frac{1}{3}(2 - \frac{2A}{x^3})$. La $\alpha = \sqrt[3]{A}$, $g'(\alpha) = \frac{1}{3}(2 - \frac{2A}{(\sqrt[3]{A})^3}) = 0$. Convergență cel puțin pătratică. Pentru $x_0 > 0$, toți $x_k > 0$. $x_{k+1} - \alpha = \frac{1}{3}\left(2x_k + \frac{A}{x_k^2}\right) - \alpha = \frac{2x_k^3 + A - 3\alpha x_k^2}{3x_k^2}$. Cum $A = \alpha^3$: $x_{k+1} - \alpha = \frac{2x_k^3 + \alpha^3 - 3\alpha x_k^2}{3x_k^2} = \frac{(x_k - \alpha)^2(2x_k + \alpha)}{3x_k^2}$. Dacă $x_k > 0$ și $\alpha > 0$, atunci $2x_k + \alpha > 0$ și $3x_k^2 > 0$. Deci $x_{k+1} - \alpha \geq 0$. Așadar, $x_k \geq \alpha$ pentru $k \geq 1$. $x_{k+1} - x_k = \frac{1}{3}(2x_k + \frac{A}{x_k^2}) - x_k = \frac{1}{3}(-x_k + \frac{A}{x_k^2}) = \frac{A - x_k^3}{3x_k^2}$. Dacă $x_k > \alpha$ (pentru $k \geq 1$), atunci $x_k^3 > A$, deci $A - x_k^3 < 0$. Astfel $x_{k+1} - x_k < 0$. Șirul $(x_k)_{k \geq 1}$ este descrescător și mărginit inferior de α. Prin urmare, converge la α pentru orice $x_0 > 0$.

Ecuația 2: $\frac{A}{x^3} - 1 = 0$ Funcția $f(x) = \frac{A}{x^3} - 1$. Derivata $f'(x) = -\frac{3A}{x^4}$. Iterația Newton $x_{k+1} = x_k - \frac{A/x_k^3 - 1}{-3A/x_k^4} = x_k + \frac{x_k^4(A/x_k^3 - 1)}{3A} = x_k + \frac{x_k(A - x_k^3)}{3A}$. $x_{k+1} = \frac{3Ax_k + Ax_k - x_k^4}{3A} = \frac{4Ax_k - x_k^4}{3A} = x_k\left(\frac{4A - x_k^3}{3A}\right)$. Fie $g(x) = x\left(\frac{4A - x^3}{3A}\right)$. Derivata $g'(x) = \frac{4A - x^3}{3A} + x\left(\frac{-3x^2}{3A}\right) = \frac{4A - x^3 - 3x^3}{3A} = \frac{4A - 4x^3}{3A}$. La $\alpha = \sqrt[3]{A}$, $g'(\alpha) = \frac{4A - 4(\sqrt[3]{A})^3}{3A} = 0$. Convergență cel puțin pătratică. Pentru $x_{k+1} > 0$ (presupunând $x_k > 0$), $\frac{4A - x_k^3}{3A} > 0 \implies 4A - x_k^3 > 0 \implies x_k^3 < 4A$. Deci $0 < x_k < \sqrt[3]{4A}$. Dacă $x_0 = \sqrt[3]{4A}$, $x_1 = \sqrt[3]{4A}\left(\frac{4A - 4A}{3A}\right) = 0$. x_2 nedefinit. Dacă $x_0 > \sqrt[3]{4A}$, $x_1 < 0$. Nu converge la $\alpha > 0$. Studiem $x_0 \in (0, \sqrt[3]{4A})$. $x_{k+1} - \alpha = x_k\left(\frac{4A - x_k^3}{3A}\right) - \alpha = \frac{4Ax_k - x_k^4 - 3A\alpha}{3A}$. Cu $A = \alpha^3$: $x_{k+1} - \alpha = \frac{4\alpha^3 x_k - x_k^4 - 3\alpha^4}{3\alpha^3} = \frac{-(x_k^4 - 4\alpha^3 x_k + 3\alpha^4)}{3\alpha^3}$. Factorizăm $P(x_k) = x_k^4 - 4\alpha^3 x_k + 3\alpha^4$. $P(\alpha) = \alpha^4 - 4\alpha^4 + 3\alpha^4 = 0$. $P'(x_k) = 4x_k^3 - 4\alpha^3$. $P'(\alpha) = 0$. Deci α este rădăcină dublă. $P(x_k) = (x_k - \alpha)^2(x_k^2 + 2\alpha x_k + 3\alpha^2)$. $(x_k^2 + 2\alpha x_k + 3\alpha^2) = (x_k + \alpha)^2 + 2\alpha^2 > 0$ pentru x_k real și $\alpha \neq 0$. Deci $x_{k+1} - \alpha = -\frac{(x_k - \alpha)^2(x_k^2 + 2\alpha x_k + 3\alpha^2)}{3\alpha^3}$. Dacă $x_k \neq \alpha$, $(x_k - \alpha)^2 > 0$. Termenul $(x_k^2 + 2\alpha x_k + 3\alpha^2) > 0$. Deci $x_{k+1} - \alpha < 0$ pentru $x_k \neq \alpha$. Asta înseamnă că $x_{k+1} < \alpha$ pentru $k \geq 0$ (dacă $x_0 \neq \alpha$). Dacă $0 < x_k < \alpha$: șirul (x_k) trebuie să fie crescător și mărginit superior de α pentru a converge. $x_{k+1} - x_k = x_k\left(\frac{4A - x_k^3}{3A}\right) - x_k = x_k\left(\frac{4A - x_k^3 - 3A}{3A}\right) = x_k\left(\frac{A - x_k^3}{3A}\right)$. Dacă $0 < x_k < \alpha$, atunci $x_k^3 < A$, deci $A - x_k^3 > 0$. $x_{k+1} - x_k > 0$. Șirul este crescător și mărginit superior de α (deoarece $x_{k+1} < \alpha$). Converge la α. Dacă $\alpha < x_k < \sqrt[3]{4A}$: Am $x_{k+1} < \alpha$. La pasul următor, x_{k+1} este în cazul $0 < x < \alpha$ și va converge. Deci, pentru $0 < x_0 < \sqrt[3]{4A}$, șirul converge la α. Valoarea lui b este $\sqrt[3]{4A}$.