Problema 1

Să se determine o formulă de cuadratură de forma

$$\int_{-1}^{1} f(t)dt = A_1 f(-1) + A_2 f(-1) + A_3 f(t_3) + A_4 f(t_4) + R(f)$$

care să aibă grad maxim de exactitate. Dacă neglijăm restul (eroarea), integrala se poate aproxima prin:

$$\int_{-1}^{1} f(t)dt \approx A_1 f(-1) + A_2 f(-1) + A_3 f(t_3) + A_4 f(t_4)$$

Fiindca dorim ca formula de cuadratură să aibă grad maxim de exactitate, adică pentru toate polinoamele cu grad <= n, integrala:

$\int_{-1}^{1} f(t)dt = A_1 f(-1) + A_2 f(-1) + A_3 f(t_3) + A_4 f(t_4)$, pentru toate polinoamele $\mathbb{P} \in \mathbb{P}_n$.

Căutăm o formulă de cuadratură $F(f) = \int_0^\infty w(t) f(t) dt$ care este exactă, anume aplicând formula asupra funcției, obținem chiar valorile integralei. Integrala de aproximat este apropiată de o cuadratură de tip **Gauss-Legendre**, însă ne încurcă apariția derivatei $f'(-1)$. Alegem funcțiile de test $f(t) = 1, t, t^2, t^3, \ldots$, deoarece acestea se pretează cuadraturilor de tip Gauss-Laguerre, putând face o paralelă cu aceste tipuri de cuadraturi.

Stim ca $\int_{-1}^{1} t^k dt = \frac{t^{k+1}}{k+1} |_{-1}^{1} = \frac{1-(-1)^{k+1}}{k+1}$. Avem 6 necunoscute, asa ca alegem functiile de test: $1, t, t^2, t^3, t^4, t^5$ pentru a nu limita cuadratura prin alegerea punctelor fixe t_3, t_4. Rezulta urmatorul tabel:

$$\begin{bmatrix} f(t) & f(-1) & f'(-1) & f(t_3) & f(t_4) & \int_{-1}^{1} f(t)\mathrm{dt} \\ 1 & 1 & 0 & 1 & 1 & 2 \\ t & -1 & 1 & t_3 & t_4 & 0 \\ t^2 & 1 & -2 & t_3^2 & t_4^2 & \frac{1}{2} \\ t^3 & -1 & 3 & t_3^3 & t_4^3 & 0 \\ t^4 & 1 & -4 & t_3^4 & t_4^4 & \frac{2}{5} \\ t^5 & -1 & 5 & t_3^5 & t_4^5 & 0 \end{bmatrix}$$

Folosind valorile din tabel, ajungem la urmatorul sistem:

$$
\begin{array}{ll}
f(t) = 1 : A_1 + A_3 + A_4 = 2 & A_1 + A_3 + A_4 = 2 \\
f(t) = t : -A_1 + A_2 + t_3A_3 + t_4A_4 = 0 & -A_1 + A_2 + t_3A_3 + t_4A_4 = 0 \\
f(t) = t^2 : A_1 - 2A_2 + t_3^2A_3 + t_4^2A_4 = \frac{1}{2} & A_1 - 2A_2 + t_3^2A_3 + t_4^2A_4 = \frac{1}{2} \\
f(t) = t^3 : -A_1 + 3A_2 + t_3^3A_3 + t_4^3A_4 = 0 \quad \Rightarrow & -A_1 + 3A_2 + t_3^3A_3 + t_4^3A_4 = 0 \\
f(t) = t^4 : A_1 - 4A_2 + t_3^4A_3 + t_4^4A_4 = \frac{2}{5} & A_1 - 4A_2 + t_3^4A_3 + t_4^4A_4 = \frac{2}{5} \\
f(t) = t^5 : -A_1 + 5A_2 + t_3^5A_3 + t_4^5 = 0 & -A_1 + 5A_2 + t_3^5A_3 + t_4^5 = 0
\end{array}
$$

Rezolvam sistemul:

$$A_1 = 2 - A_3 - A_4$$

$$A_2 = 2 - A_3(1 + t_3) - A_4(1 + t_4)$$

$$-2 + A_3 + A_4 - 4 + 2A_3 + 2t_3A_3 + 2A_4 + 2t_4A_4 + t_3^2A_3 + t_4^2A_4 = \frac{1}{2} \Leftrightarrow -2 + A_3\left(t_3^2 + 2t_3 + 1\right) = \frac{1 - 2A_4\left(t_4^2 + 3t_4 + 1\right)}{2} \Leftrightarrow A_3 = \frac{5 - 2A}{t_3^2}$$

Sistem neliniar daca lasam t_3, t_4 nefixate. Am putea lua radacinile polinomului Legendre de gradul 2, insa putem folosi si MATLAB Symbolic pentru a rezolva.

```
syms A1 A2 A3 A4 t3 t4

eq1 = A1 + A3 + A4 == 2;
eq2 = -A1 + A2 + t3 .* A3 + t4 .* A4 == 0;
eq3 = A1 - 2 * A2 + t3 .^ 2 .* A3 + t4 .^ 2 .* A4 == 1/2;
eq4 = -A1 + 3 * A2 + t3 .^ 3 .* A3 + t4 .^ 3 .* A4 == 0;
eq5 = A1 - 4 * A2 + t3 .^ 4 .* A3 + t4 .^ 4 .* A4 == 2/5;
eq6 = -A1 + 5 * A2 + t3 .^ 5 .* A3 + t4 .^ 5 .* A4 == 0;

solution = solve([eq1, eq2, eq3, eq4, eq5, eq6], [A1, A2, A3, A4, t3, t4]);
fields = fieldnames(solution);
for i = 1:length(fields)
    fprintf('%s = ', fields{i});
    disp(vpa(solution.(fields{i}), 6))
    fprintf('\n');
end
```

A1 =

$$\begin{pmatrix} 0.294707 \\ 0.294707 \end{pmatrix}$$

A2 =

$$\begin{pmatrix} 0.0235043 \\ 0.0235043 \end{pmatrix}$$

A3 =

$$\begin{pmatrix} 0.319098 \\ 1.3862 \end{pmatrix}$$

A4 =

$$\begin{pmatrix} 1.3862 \\ 0.319098 \end{pmatrix}$$

```
t3 =
```

$$\begin{pmatrix} 0.888999 \\ -0.00899889 \end{pmatrix}$$

```
t4 =
```

$$\begin{pmatrix} -0.00899889 \\ 0.888999 \end{pmatrix}$$

Solutia finala:

$$A1 = \begin{pmatrix} 0.294707 \\ 0.294707 \end{pmatrix} A2 = \begin{pmatrix} 0.0235043 \\ 0.0235043 \end{pmatrix} A3 = \begin{pmatrix} 0.319098 \\ 1.3862 \end{pmatrix} A4 = \begin{pmatrix} 1.3862 \\ 0.319098 \end{pmatrix} t3 = \begin{pmatrix} 0.888999 \\ -0.00899889 \end{pmatrix} t4 = \begin{pmatrix} -0.00899889 \\ 0.888999 \end{pmatrix}$$

Eroarea:

$$R(f) = \frac{f^{(6)}(\xi)}{6!} \cdot \int_{-1}^{1} \omega(t)dt, \int_{-1}^{1} \omega(t)dt = \int_{-1}^{1} (t+1)^2(t-t_3)(t-t_4)dt = \frac{8t_3t_4}{3} - \frac{4t_4}{3} - \frac{4t_3}{3} + \frac{16}{15} = -0.128$$

```
syms t t3 t4

omega = (t+1) .^ 2 .* (t - t3) .* (t - t4);
int_omega = int(omega, t, -1, 1);
fprintf("I_omega = ");
```

```
I_omega =
```

```
disp(int_omega);
```

$$\frac{8\,t_3\,t_4}{3} - \frac{4\,t_4}{3} - \frac{4\,t_3}{3} + \frac{16}{15}$$

```
fprintf("\n");

omega_known = (t+1) .^ 2 .* (t - 0.888999) .* (t + 0.00899889);
int_omega_known = int(omega_known, t, -1, 1);
fprintf("I_omega = ");
```

```
I_omega =
```

```
disp(vpa(int_omega_known, 6));
```

$$-0.128$$

```
fprintf("\n");
```

$$R_1(f) = R_2(f) = R(f) = -\frac{f^{(6)}(\xi)}{6!} \cdot 0.128, \xi \in (-1, 1)$$

Rezulta 2 formule de cuadratura:

$$\int_{-1}^{1} f(t)\mathrm{dt} = 0.294707 \cdot f(-1) + 0.0235043 \cdot f'(-1) + 0.319098 \cdot f(0.888999) + 1.3862 \cdot f(-0.00899889) - \frac{f^{(6)}(\xi)}{6!} \cdot 0.128$$

$$\int_{-1}^{1} f(t)\mathrm{dt} = 0.294707 \cdot f(-1) + 0.0235043 \cdot f'(-1) + 1.3862 \cdot f(-0.00899889) + 0.319098 \cdot f(0.888999) - \frac{f^{(6)}(\xi)}{6!} \cdot 0.12$$

```
f = @(t) exp(t);
df = @(t) exp(t);
I_exact = integral(@(t) f(t), -1, 1);

A1 = 0.294707;
A2 = 0.235043;
A3 = 0.319098;
A4 = 1.3862;
t3 = 0.888999;
t4 = -0.00899889;

I_approx = A1 * f(-1) + A2 * df(-1) + A3 * f(t3) + A4 * f(t4);
R = I_exact - I_approx;

fprintf("### I_EXACT: %.16e\n", I_exact);
```

```
### I_EXACT: 2.3504023872876028e+00
```

```
fprintf("### I_APPROX: %.16e\n", I_approx);
```

```
### I_APPROX: 2.3449334005032796e+00
```

```
fprintf("### REST: %.16e\n", R);
```

```
### REST: 5.4689867843231710e-03
```

Problema 2

Fie ecuatia $f(x) = 0, f : [a, b] \to \mathbb{R}, f \in C^3[a, b]$ si α o radacina simpla a ei.

a) Sa se arate ca

$$x_{k+1} = x_k - 2\frac{f(x_k)}{f'(x_k)\left(1 + \sqrt{1 - \frac{2f(x_k)f''(x_k)}{f'(x_k)^2}}\right)}$$

genereaza un sir care converge cubic.

Pentru a arata ca metoda iterativa genereaza un sir care converge cubic, putem folosi eroarea $R_k = x_k - \alpha$ si sa demonstram faptul ca $R_{k+1} = C \cdot R_k^3 + O(R_k^4), C \text{ constanta}$. Asta inseamna sa demonstram ca $R_{k+1} \in O(R_k^3)$.

Stim faptul ca $f \in C^3[a, b]$, deci ne putem folosi de **dezvoltarea Taylor** in jurul lui α pentru a rezolva problema.

$$f(x_k) = f(\alpha + R_k) = f(\alpha) + R_k \cdot f'(\alpha) + \frac{R_k^2}{2!} \cdot f''(\alpha) + \frac{R_k^3}{3!} \cdot f^{(3)}(\xi_k)$$

$$f'(x_k) = f'(\alpha + R_k) = f'(\alpha) + R_k \cdot f''(\alpha) + \frac{R_k^2}{2!} \cdot f^{(3)}(\eta_k)$$

$$f''(x_k) = f''(\alpha + R_k) = f''(\alpha) + R_k \cdot f^{(3)}(\gamma_k),$$

unde $R_k = x_k - \alpha$ si $\xi_k, \eta_k, \gamma_k \in (\alpha, x_k)$.

Cunoastem faptul ca α este o radacina simpla a aplicatiei f, asta inseamna ca $f(\alpha) = 0$ si $f'(\alpha) \neq 0$. Sistemul devine:

$$f(x_k) = f(\alpha + R_k) = f'(\alpha) + \frac{R_k^2}{2!} \cdot f''(\alpha) + \frac{R_k^3}{3!} \cdot f^{(3)}(\xi_k)$$

$$f'(x_k) = f'(\alpha + R_k) = f'(\alpha) + R_k \cdot f''(\alpha) + \frac{R_k^2}{2!} \cdot f^{(3)}(\eta_k)$$

$$f''(x_k) = f''(\alpha + R_k) = f''(\alpha) + R_k \cdot f^{(3)}(\gamma_k),$$

unde $R_k = x_k - \alpha$ si $\xi_k, \eta_k, \gamma_k \in (\alpha, x_k)$.

Rezolvarea iteratiei

```
syms Rk f1 f2 f3 real
fk = f1*Rk + f2*Rk^2/2 + f3*Rk^3/6;
fpk = f1 + f2*Rk + f3*Rk^2/2;
fppk = f2 + f3*Rk;

expr_Rkp1 = Rk - 2 * fk / (fpk * (1 + sqrt(1 - 2*fk*fppk / fpk^2)));
simplify(expr_Rkp1)
```

```
ans =
```

$$\mathrm{Rk} - \frac{2\,\mathrm{Rk}\,|\sigma_1|\,\left(f_3\,\mathrm{Rk}^2 + 3\,f_2\,\mathrm{Rk} + 6\,f_1\right)}{\left(3\,|\sigma_1| + \sqrt{3}\,\sqrt{-\mathrm{Rk}^4 {f_3}^2 - 4\,f_2\,\mathrm{Rk}^3 f_3 - 12\,\mathrm{Rk}^2 f_1 f_3 + 12\,{f_1}^2}\right)\sigma_1}$$

where

$$\sigma_1 = f_3\,\mathrm{Rk}^2 + 2\,f_2\,\mathrm{Rk} + 2\,f_1$$

Rezulta iteratia:

$$R_{k+1} = \frac{2R_k\left|f^{(3)}(\alpha)R_k^2 + 2f''(\alpha)R_k + 2f'(\alpha)\right|\left(f^{(3)}(\alpha)R_k^2 + 3f''(\alpha)R_k + 6f'(\alpha)\right)}{\left(3\left|f^{(3)}(\alpha)R_k^2 + 2f''(\alpha)R_k + 2f'(\alpha)\right| + \sqrt{3}\cdot\sqrt{-R_k^4 f^{(3)}(\alpha)^2 - 4f''(\alpha)R_k^3 f^{(3)}(\alpha) - 12R_k^2 f'(\alpha)f^{(3)}(\alpha) + 12f'(\alpha)^2}\right)\cdot\left(f^{(3)}(\alpha\right.}$$

$f'(\alpha)$ este dominant la numarator, astfel numarator $\approx 2R_k \cdot 2f'(\alpha) \cdot 6f'(\alpha) = 24R_k f'(\alpha)$.

Numitorul, la simplificare, va ajunge la ceva proportional cu R_k^4, pentru primul termen nenul R_k. Astfel, dupa simplificari rezulta : $R_{k+1} = O\left(R_k^3\right)$, deci sirul converge cubic.

```
f = @(x) exp(x) - x .^ 2;
df = @(x) exp(x) - 2 .* x;
d2f = @(x) exp(x) - 2;
```

```
[root, n_iter] = iter_method_cubic(f, df, d2f, 0.3);
fprintf('Root: %.16e | No. iterations: %d\n', root, n_iter);
```

```
Root: -7.0346742249839167e-01 | No. iterations: 4
```

```
function [root, n_iter] = iter_method_cubic(f, df, d2f, x0, tol, max_iter)
    %% ITER_METHOD_CUBIC = implementeaza metoda iterativa descrisa mai sus:
    %  x_{k+1} = x_k - 2 * f(x_k) / (f'(x_k) * (1 + sqrt(1 - 2 * f(x_k) * f''(x_k) / f'(x_k)^2
    %
    % Inputs:
    %
    % f         - functia de aproximat;
    % df        - prima derivata a functiei;
    % d2f       - a doua derivata a functiei;
    % x0        - nodul de pornire;
    % tol       - eroarea de aproximare admisa;
    % max_iter  - numarul maxim de iteratii;
    %
    % Outputs:
    %
    % root      - radacina aproximata;
    % n_iter    - numarul de iteratii;
    % Eroare: impartire la 0/radical imaginar sau daca nu converge in numarul maxim de iterati

    if nargin < 4
        x0 = 0;
    end
    if nargin < 5
        tol = 1e-6;
    end
    if nargin < 6
        max_iter = 100;
    end

    root = x0;
    for n_iter = 1:max_iter
        fxk = f(root);
        dfxk = df(root);
        d2fxk = d2f(root);

        b = dfxk .* (1 + sqrt(1 - (2 .* fxk .* d2fxk) ./ dfxk .^ 2));
        if abs(b) < eps || ~isreal(b)
            error('Zero division or non-real radical')
        end

        xknext = root - 2 .* fxk/b;
        if abs(xknext - root) < tol
            root = xknext;
            return;
        end

        root = xknext;
```

```
    end

    warning('No convergence in %d iterations!', max_iter);
end
```