Subiectul 3

Problema 1 *Considerăm problema de interpolare Hermite: determinaţi* $H_{2n-1}f \in P_{2n-1}$ *astfel încât*

$$\left(H_{2n-1}f\right)\left(\tau_\nu\right) = f\left(\tau_\nu\right), \ \ \left(H_{2n-1}f\right)'\left(\tau_\nu\right) = f'\left(\tau_\nu\right), \ \ \nu = 1, 2, \ldots, n. \qquad ((*))$$

Prin analogie cu formula lui Lagrange, polinomul care rezolvă (*) se poate scrie cu ajutorul polinoamelor fundamentale de interpolare Hermite $h_{\nu,0}$, $h_{\nu,1}$ *sub forma*

$$\left(H_{2n-1}f\right)(t) = \sum_{\nu=1}^{n} \left[h_{\nu,0}(t) f_\nu + h_{\nu,1}(t) f'_\nu\right].$$

(a) Căutaţi $h_{\nu,0}$ *şi* $h_{\nu,1}(t)$ *sub forma*

$$h_{\nu,0}(t)(t) = \left(a_\nu + b_\nu t\right) \ell_\nu^2(t), \ \ h_{\nu,1}(t)(t) = \left(c_\nu + d_\nu t\right) \ell_\nu^2(t),$$

unde ℓ_ν *sunt polinoamele fundamentale de interpolare Lagrange. Determinaţi constantele* a_ν, b_ν, c_ν, d_ν.

(b) Obţineţi formula de cuadratură

$$\int_a^b f(t) w(t) \mathrm{d}\, t = \sum_{\nu=1}^{n} \left[\lambda_\nu f\left(\tau_\nu\right) + \mu_\nu f'\left(\tau_\nu\right)\right] + R_n(f)$$

cu proprietatea $R_n(f) = 0$ *pentru orice* $f \in \mathbb{P}_{2n-1}$.

(c) Ce condiţii trebuie impuse asupra polinomului nodurilor $\omega_n(t) = \prod_{\nu=1}^{n}(t - t_\nu)$ *(sau asupra nodurilor* τ_ν*) astfel ca* $\mu_\nu = 0$ *pentru* $\nu = 1, 2, \ldots, n$*?*

Soluţie.

(a) Polinoamele $h_{\nu,0}$ trebuie să verifice

$$h_{\nu,0}\left(\tau_\nu\right) = 1, \ \ h'_{\nu,0}\left(\tau_\nu\right) = 0,$$

iar condiţiile $h_{\nu,0}(\tau_\mu) = h'_{\nu,0}(\tau_\mu) = 0$, $\mu \neq \nu$, sunt automat verificate datorită formei lui $h_{\nu,0}$. Astfel,

$$a_\nu + b_\nu \tau_\nu = 1, \ \ b_\nu + \left(a_\nu + b_\nu \tau_\nu\right) \cdot 2\ell_\nu\left(\tau_\nu\right) \ell'_\nu\left(\tau_\nu\right) = 0,$$

adică,

$$\begin{aligned} a_\nu + b_\nu \tau_\nu &= 1, \\ b_\nu + 2\ell'_\nu\left(\tau_\nu\right) &= 0. \end{aligned}$$

Rezolvând sistemul cu necunoscutele a_ν şi b_ν şi înlocuind în $h_{\nu,0}$ se obţine

$$h_{\nu,0}(t) = \left[1 - 2(t - \tau_\nu) \ell'_\nu(\tau_\nu)\right] \ell_\nu^2(t), \quad \nu = 1, 2, \ldots, n.$$

Analog, $h_{\nu,1}$ satisface

$$h_{\nu,1}\left(\tau_\nu\right)=0,\ h'_{\nu,1}\left(\tau_\nu\right)=1$$

din care se obţine

$$c_\nu+d_\nu\tau_\nu=0,\ d_\nu+(c_\nu+d_\nu\tau_\nu)\cdot 2\ell_\nu(\tau_\nu)\ell'_\nu(\tau_\nu)=1,$$

adică,

$$c_\nu+d_\nu\tau_\nu=0,\quad d_\nu=1.$$

Astfel, $c_\nu=-\tau_\nu$ şi

$$h_{\nu,1}(t)=(t-\tau_\nu)\ell_\nu^2(t),\quad \nu=1,2,\ldots,n.$$

Derivata polinomului fundamental în τ_ν este

$$\ell'_\nu(\tau_\nu)=\sum_{\mu\neq\nu}\frac{1}{\tau_\nu-\tau_\mu}.$$

(b) Formula de cuadratură se obţine integrând termen cu termen

$$\int_a^b f(t)w(t)\mathrm{d}\,t=\int_a^b p(t)w(t)\mathrm{d}\,t+R_n(f),$$

Gradul de exactitate este $2n-1$. Utilizând punctul (a), se obţine

$$\begin{aligned}\int_a^b p(t)w(t)\mathrm{d}\,t&=\int_a^b\sum_{\nu=1}^n\left[h_{\nu,0}(t)f_\nu+h_{\nu,1}\left(t\right)f'_\nu\right]w(t)\mathrm{d}\,t\\&=\sum_{\nu=1}^n\left[f_\nu\int_a^b h_\nu(t)w(t)\mathrm{d}\,t+f'_\nu\int_a^b k_\nu\left(t\right)w(t)\mathrm{d}\,t\right].\end{aligned}$$

Deci

$$\lambda_\nu=\int_a^b h_{\nu,0}(t)w(t)\mathrm{d}\,t,\quad \mu_\nu=\int_a^b h_{\nu,1}\left(t\right)w(t)\mathrm{d}\,t,\quad \nu=1,2,\ldots,n.$$

(b) Pentru ca toţi coeficienţii μ să fie nuli, trebuie să avem

$$\int_a^b h_{\nu,1}\left(t\right)w(t)\mathrm{d}\,t=0,\quad \nu=1,2,\ldots,n.$$

sau, pe baza lui (a), observând că $\ell_\nu(t)=\frac{\omega_n(t)}{(t-\tau_\nu)\omega'_n(\tau_\nu)}$,

$$\frac{1}{\omega'_n(\tau_\nu)}\int_a^b\frac{\omega_n(t)}{(t-\tau_\nu)\omega'_n(\tau_\nu)}\omega_n(t)w(t)\mathrm{d}\,t=0,\quad \nu=1,2,\ldots,n.$$

adică,

$$\int_a^b \ell_\nu(t)\omega_n(t)w(t)\mathrm{d}\,t = 0, \quad \nu = 1, 2, \ldots, n.$$

Deoarece $\{\ell_\nu(t)\}_{\nu=1}^n$ formează o bază a lui $\mathbb{P}_{n-1}$ (ℓ_ν sunt liniar independente şi generează $\mathbb{P}_{n-1}$), ω_n trebuie să fie ortogonal pe $[a, b]$ în raport cu $w(t) = 1$ pe toate polinoamele de grad mai mic, adică, $\omega_n(t) = \pi_n(t; w)$. Se obţine o formulă de cuadratură gaussiană.

∎

Problema 2 *Implementaţi în MATLAB o metodă hibridă Newton+înjumătăţire pentru rezolvarea ecuaţiei $f(x) = 0$, $f \in C^1$. Testaţi pentru $f(x) = J_0(x)$, pe intervalul $[0, 4]$ şi comparaţi cu metoda lui Newton pentru $x_0 = 0.01$.*

```
function [xFinal,ni]=Newtonsafevb(f,fd,a,b,tol,nitmax,varargin)
% NEWTSAFEVB - root-finder using hybrid Newton-Bisection method to always maintain bracket
%
% [xFinal, xN, errorN, ni]=Newtonsafe(f,fd,a,b,tol,nitmax,varargin)
%
% f, fd - returns the function and its derivative
% a,b: initial bracket
% tol: stopping condition for error f(x) <= tol or |b-a|<tol*b
%
% xFinal: final value
% xN: vector of intermediate iterates
% errorN: vector of errors
if nargin<6, nitmax=50; end
% initialize the problem
h = b-a;
fa = f(a,varargin{:});
fb = f(b,varargin{:});
if ( sign(fa) == sign(fb) )
    error('function must be bracketed to begin with' );
end
c = a ; % start on the left side (could also choose the middle
fc = f(c,varargin{:}); df=fd(c,varargin{:}) ;
%xN(1) = c;
%errorN(1,:) = [ abs(fc) h ];
% begin iteration until convergence or Maximum Iterations
ni=1;
for i = 1:nitmax
    % try a Newton step
    c = c - fc/df;
    % if not in bracket choose bisection
    if ( ~(a <= c && b >= c) )
        c = a + h/2;
```

```
    end
    % Evaluate function and derivative at new c
    fc = f(c,varargin{:}); df=fd(c,varargin{:});
    % check and maintain bracket
    if ( sign(fc) ~= sign(fb) )
        a=c;
        fa=fc;
    else
        b = c;
        fb = fc;
    end
    h = b-a;
    % calculate errors and track solutions
    absError = abs(fc);
    relError = h;
    % check if converged
    if ( absError < tol || relError < tol*b ) %succes
        xFinal=c; ni=i; return;
    end
end
% clean up
error('Maximum iterations exceeded' );
```

Test

```
%testsub3
g = @(x) besselj(0,x);
gd = @ (x) -besselj(1, x);
[z2,ni2]=Newton(g,gd,0.01,tol)
[z5, ni5]=Newtonsafevb(g,gd,0,4,tol)
```

Rezultate:

```
z2 =
200.2772
ni2 =
5
z5 =
2.4048
ni5 =
4
```

Subiectul 4

Problema 3 *(a) Utilizaţi metoda coeficienţilor nedeterminaţi pentru a obţine formula de cuadratură (cu gradul de exactitate $d \geq 2$) de forma*

$$\int_0^1 y(s)\mathrm{d}\, s \approx ay(0) + by(1) - c\left[y'(1) - y'(0)\right] + R(f).$$

(b) Transformaţi formula de la (a) într-o formulă pentru a aproxima $\int_x^{h+x} f(t)\mathrm{d}\, t$.

(c) Obţineţi o formulă de integrare repetată bazată pe formula de la (b) pentru a aproxima $\int_a^b f(t)\mathrm{d}\, t$. Interpretaţi rezultatul.

Soluţie.

(a) Punând $y(s) = 1$, $y(s) = s$, $y(s) = s^2$, din condiţiile de exactitate se obţine

$$\begin{aligned} a + b + 0c &= 1 \\ 0a + b - 0c &= \frac{1}{2} \\ 0a + b - 2c &= \frac{1}{3} \end{aligned}$$

Soluţia este: $a = \frac{1}{2}, b = \frac{1}{2}, c = \frac{1}{12}$, adică,

$$\int_0^1 y(s)\mathrm{d}\, s = \frac{1}{2}\left[y(0) + y(1)\right] - \frac{1}{12}\left[y'(1) - y'(0)\right] + R(f)$$

$$R(f) = \frac{f^{(4)}(\xi)}{4!}\int_0^1 s^2(s-1)^2\mathrm{d}\, s = \frac{1}{720}f^{(4)}(\xi)$$

:

(b) Schimbarea de variabilă $t = x + hs$, $dt = hds$ ne conduce la

$$\begin{aligned} \int_x^{x+h} f(t)\mathrm{d}\, t = h\int_0^1 (x + hs)\,\mathrm{d}\, s &= \frac{h}{2}\left[f(x) + f(x+h)\right] \\ &- \frac{h^2}{12}\left[f'(x+h) - f'(h)\right] + \frac{h^4}{720}f^{(4)}(\zeta)\,. \end{aligned}$$

(c) Punând $h = (b-a)/n$, $x_k = a + kh$, $f_k = f(x_k)$, $f'_k = f'(x_k)$, $k = 0, 1, \ldots, n$, obţinem folosind (b), că

$$\begin{aligned} \int_a^b f(t)\mathrm{d}\, t = \sum_{k=0}^{n-1}\int_{x_k}^{x_k+h} f(t)\mathrm{d}\, t &\approx \frac{h}{2}\left[(f_0 + f_1) + (f_1 + f_2) + \cdots\right. \\ &+ \left.(f_{n-1} + f_n)\right] - \frac{h^2}{12}\left[(f'_1 - f'_0) + (f'_2 - f'_{10}) + \cdots + \left(f'_n - f'_{n-1}\right)\right] \\ &= h\left(\frac{1}{2}f_0 + f_1 + \cdots + f_{n-1} + \frac{1}{2}f_n\right) - \frac{h^2}{12}\left[f'(b) - f'(a)\right]. \end{aligned}$$

Restul

$$R_n(f) = \frac{(b-a)^4}{720n^3} f^{(4)}(\zeta)$$

Interpretare regula trapezelor cu o “corecţie la capete”. Corecţia aproximează eroarea în formula trapezelor:

$$-\frac{b-a}{12} h^2 f''(\xi) \approx -\frac{h^2}{12} \left[f'(b) - f'(a) \right].$$

■

Problema 4 *Implementaţi în MATLAB o metodă hibridă secantă+înjumătăţire pentru rezolvarea ecuaţiei $f(x) = 0$. Testaţi pentru $f(x) = 1 - \frac{2}{x^2-2x+2}$. Câte iteraţii sunt necesare? Comparaţi cu metoda secantei şi a înjumătăţirii.*

```
function [z,ni]=secantsafe(f,a,b,tol)
% SECANTSAFE - safe secant method = secant + bisection
% call z=secantsafe(f,a,b,tol)

% The method uses three points a, b, and c. The points a and b are the next
% points xk and xk?1 in the secant method approximation.
% The points b and c form a sign change interval (proper bracket) for the root.
% The idea behind the method is that if the secant method produces an
% undesirable approximation, we take the midpoint of the sign change interval
% (next bisection iterate) as our next approximation.

% Let fa = f(a), fb = f(b) and fc = f(c) which must satisfy
% Conditions:
% 1. fa, fb, fc ~= 0,
% 2. sign(fb) ~= sign(fc) (sign change interval)
% 3. |fb| <= |fc|.

fa=f(a); fb=f(b);
if fa==0
    z=a; return;
end
if fb==0
    z=b; return;
end
if sign(fa)==sign(fb)
    error('f(a) and f(b) must have different sign')
end
c=a; fc=fa;
ni=0;
while 1 %forever
```

```
    ni=ni+1;
    if abs(fc) < abs(fb) %swap b and c
        t = c; c = b; b = t;
        t = fc; fc = fb; fb = t;
        % In this case, a and b may no longer
        % be a pair of secant iterates, and we must set a = c.
        a = c; fa = fc;
    end
    if abs(b-c) <= tol, break; end %success
    dm = (c-b)/2;
    df = (fa-fb);
    if df == 0
        ds = dm;
    else
        ds = -fb*(a-b)/df;
    end
    if (sign(ds)~=sign(dm) || abs(ds) > abs(dm))%bisection or secant
        dd = dm;
    else
        dd = ds;
    end
    if abs(dd) < tol
        dd = 0.5*sign(dm)*tol;
    end
    % New iterate b+dd
    d = b + dd;
    fd = f(d);
    if fd == 0
        b = d; c = d; fb = fd; fc = fd;
        break;
    end
    a = b; b = d;
    fa = fb; fb = fd;
    if sign(fb) == sign(fc)
        c = a; fc = fa;
    end
end
z=(b+c)/2;
```

Test

```
f=@(x) 1-2./(x.^2-2*x+2);
%[-10,1]
[z1,ni1]=secantsafe(f,-10,1,1e-8)
[z2,ni2]=Bisection(f,-10,1,1e-8)
[z3,ni3]=secant(f,-10,1,1e-8)
```

Rezultate

```
z1 =
-1.9747e-09
ni1 =
11
z2 =
-1.6298e-09
ni2 =
31
Error using secant (line 28)
numarul maxim de iteratii depasit
Error in testsecantsafe2 (line 5)
[z3,ni3]=secant(f,-10,1,1e-8)
```