

```

1: % UNM, prva grupa zadataka, prvi zadatak
2: % Resenje zapisano za GNU Octave, www.octave.org
3: %
4: % Goran Rakic, <gox AT devbase DOT net>
5: % ponedjeljak, 27. mart 2006. godine
6:
7: % Ucitavamo tablica1.m i proveravamo da li postoje varijable X i F
8: if( exist("tablica1", "file")==0 )
9:     error("Ne postoji datoteka tablica1.m");
10: else
11:     global X F;
12:     source("tablica1.m");
13:     f();
14:
15:     % >> tablica1.m
16:     %
17:     %     function f()
18:     %         global X F;
19:     %         X = linspace(-1, 1, 10);
20:     %         F = 1./(1+25*X.^2);
21:     %     %     Nule Cebisevljevog polinoma
22:     %     %     X = cos( (2*(9:-1:0)+1)/20 * pi );
23:     %     %     F = 1./(1+25*X.^2);
24:     %     endfunction
25:
26:     if( ~ isglobal("X") && isglobal("F") )
27:         error("Nisu definisani globalni nizovi X i F");
28:     elseif( length(X) ~= length(F) )
29:         error("Nizovi X i F nemaju isti broj tacaka");
30:         % Ne kontrolisemo da li je X strogo rastuci
31:     endif
32: endif
33:
34: % Za uneti argument u nalazimo pribliznu vrednost tablicno zadate
35: % f-je f(X) = F pomocu Lagranzevog interp. polinoma koriscenjem
36: % svih vrednosti iz tablice
37: function val = Lagr(u, LX, LF)
38:     n = length(LX);
39:     L = zeros(1,n);
40:     for i=1:n
41:         p = poly( LX(1:n ~= i) );
42:         L = L+p*LF(i)/polyval(p, LX(i));
43:     endfor
44:     %     tacke = linspace(LX(1), LX(n), 100);
45:     %     plot(tacke, 1./(1+25*tacke.^2), '@-b', tacke, polyval(L, tacke), 'r')
46:     val = polyval(L, u);
47: endfunction
48:
49: % Pravi novu tabelu vrednosti funkcije dodavanjem cvorova na
50: % sredini izmedju svaka dva cvora sa vrednoscu aritmeticke
51: % vrednosti f-je u okolnim cvorovima
52: function [X1, F1] = Prosiri(LX, LF)
53:     global X F;
54:     n = length(X);
55:     for i=1:n-1,
56:         X1(2*i-1) = X(i);
57:         F1(2*i-1) = F(i);
58:         X1(2*i) = (X(i)+X(i+1))/2;
59:         F1(2*i) = (F(i)+F(i+1))/2;
60:     endfor
61:     X1(2*n-1) = X(n);
62:     F1(2*n-1) = F(n);
63: endfunction
64:
65: % Omotaci za gornje f-je kako bi se dobilo ono sto
66: % se trazi u zadatku...
67: function val = Lagr1(u)
68:     global X F;
69:     val = Lagr(u, X, F);
70: endfunction
71:
72: function val = Lagr2(u)
73:     global X F;
74:     [X1, F1] = Prosiri(X, F);
75:     val = Lagr(u, X1, F1);
76: endfunction

```