

```

f = @(x)(1./(1+x.^2));
alfa=-5; beta=5;

figure(1)
fplot(f,[alfa,beta],'r--');
hold on
title('polinomio interpolante')
xlabel('variabile x')
ylabel('valore della funzione ')
pause

fprintf('Approssimazione mediante interpolazione\n')

itype=input('tipo di nodi. 0=equidist, 1=cheb : ');

for n=[ 8 12 16 18]

% Calcolo dei nodi
    switch itype
        case 0
            x=linspace(alfa,beta,n+1)'; % nodi di interpolazione
        case 1
            xcap=cos( (2*(0:n)+1)./(2*n+2)*pi)';
            x=(alfa+beta)/2+(beta-alfa)/2*xcap;
        otherwise
            fprintf('Errore\n')
            break
    end
% Metodo classico con if
% if (itype==0)
%     x=linspace(alfa,beta,n+1)'; % nodi di interpolazione
% else
%     xcap=cos( (2*(0:n)+1)./(2*n+2)*pi)';
%     x=(alfa+beta)/2+(beta-alfa)/2*xcap;
% end
    plot(x,f(x),'*');

% Calcolo dei coeff. del polinomio interpolante
    y = f(x);
    a = get_polyn(x,y);

% Grafico del polinomio interpolante
    t = linspace(alfa,beta,1000); % punti di valutazione per il grafico
    yp= polyval(a,t);
    figure(1)
    plot(t,yp,'k');

% Stima dell'errore
    t = linspace(alfa,beta,10000); % punti di valutazione per l'errore
    figure(2)
    pp=polyval(a,t);
    plot(t,f(t)-pp);
    title('Funzione Errore di interpolazione')
    Errore = max( abs(f(t)-pp));
    disp([n,Errore])

    pause

end

hold off

% Es 4 (facoltativo)
alfa=-1; beta=1;
n=14; n1=n+1;
N=100;

```

```

x=linspace(alfa,beta,n1)'; % nodi equispaziati
xcap = cos( (2*(0:n)+1)./(2*n+2)*pi)'; % nodi di cheb su [-1,1]
w=@(t,x,N,n1)( prod( t*ones(1,n1)-ones(N,1)*x',2));
t=linspace(alfa,beta,N)';

figure(3)
% plot con nodi equispaziati
plot(t, w(t,x,N,n1))
hold on
% plot con nodi di cheb
plot(t, w(t,xcap,N,n1),'r')
title('Polinomio nodale')
legend('nodi equispaz.','nodi Cheb')
hold off

% Es 5 (facoltativo)
alfa=-5; beta=5;
n=10; n1=n+1;
N=100;
t=linspace(alfa,beta,N)';
x=linspace(alfa,beta,n1)'; % nodi equispaziati

figure(10)
for i=1:n1,
    xm=x; xm(i)=[];
    % i-esimo polin caratteristico di Lagrange
    L=@(t)( prod( t*ones(1,n)-ones(N,1)*xm',2)/prod(x(i)-xm));
    plot(t, L(t))
    hold on
end
title('Polinomi di Lagrange - nodi equispaziati')
legend('L0','L1','L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10')
hold off

figure(11)
xcap=cos( (2*(0:n)+1)./(2*n+2)*pi)';
x=(alfa+beta)/2+(beta-alfa)/2*xcap; % nodi di Cheb
for i=1:n1,
    xm=x; xm(i)=[];
    % i-esimo polin caratteristico di Lagrange
    L=@(t)( prod( t*ones(1,n)-ones(N,1)*xm',2)/prod(x(i)-xm));
    plot(t, L(t))
    hold on
end
title('Polinomi di Lagrange - nodi di Chebyshev')
legend('L0','L1','L2', 'L3', 'L4', 'L5', 'L6', 'L7', 'L8', 'L9', 'L10')
hold off

%%%%%%%%%%%%%%
%% Studio di approssimazione con polinomi composti
%%%%%%%%%%%%%%

f=@(x)( 1./((x-3/10).^2 + 1/100) + 1./((x-9/10).^2 + 4/100) - 6);
alfa=0; beta=2;

figure(1)
fplot(f,[alfa,beta], '*-');
hold on
k=10; % # intervalli
x = linspace(alfa,beta,k+1);
s = ['k','r','b'];

for l=1:3,
    for j=1:k

```

```

        x_in=linspace(x(j),x(j+1),l+1)'; % nodi di interpolazione
% Calcolo dei coeff. del polinomio interpolante
    y = f(x_in);
    a = get_polyn(x_in,y); % coeff polinomio composito in [x_j, x_{j+1}]
    Coeff(1:l+1,j)=a;
% Grafico del polinomio interpolante
    t = linspace(x(j),x(j+1),100); % punti di valutazione per il grafico
    yp= polyval(a,t);
    plot(t,yp,s(l));
%
    pause

    end

end

pause
% Studio dell'errore

figure(2)
f=@(x)( 1./((x-3/10).^2 + 1/100) + 1./((x-9/10).^2 + 4/100) - 6);
alfa=0; beta=2;
% Es 3 (Facoltativo)
% f=@(x)(1./(1+x.^2) );
% alfa=-5; beta=5;
fplot(f,[alfa,beta], '*-');
hold on

l=1;
Err_tot_old=1;

fprintf('      h          Err          p\n')
for k=1:10

    h = (beta-alfa)/2^k;
    x = (alfa:h:beta)';

    for j=1:length(x)-1 % length(x)-1 corrisponde al # intervallini

        x_in=linspace(x(j),x(j+1),l+1)'; % nodi di interpolazione
% Calcolo dei coeff. del polinomio interpolante
            y = f(x_in);
            a = get_polyn(x_in,y); % coeff polinomio composito in [x_j, x_{j+1}]
% Grafico del polinomio interpolante
            t = linspace(x(j),x(j+1),100); % punti di valutazione per il grafico
            yp= polyval(a,t);
            plot(t,yp);
% Approssimazione dell'errore nell'intervallo j-esimo
            t = linspace(x(j),x(j+1),10000);
            yp= polyval(a,t);
            Err_max(j)=max(abs( f(t)-yp ));
        end
        Err_tot=max(Err_max);
        p = 1/log(1/2)*log( Err_tot/Err_tot_old);
        Err_tot_old=Err_tot;

        disp([h,Err_tot,p])
        pause
    end
end

```