

چکیده

در نظام آموزشی کشور ما، کلاس‌های چند پایه واقعی اجتناب‌ناپذیر است. شرایط سخت محیطی، کمبود جمعیت دانش‌آموزی، کمبود نیروی انسانی، ناکافی بودن فضای آموزشی، مهاجرت طبیعی یا اجباری، وجود مشکلات و مسائل متعدد مالی و اقتصادی، دور بودن آبادی و روستاها از یکدیگر، عدم وجود منابع کافی و لازم همگی از جمله عواملی هستند که امکان تشکیل واحدهای آموزشی را با کلاس‌های تک پایه محدود می‌سازند. لذا شناخت مسئولان نسبت به کلاسهای چند پایه، سرپیدایش کلاسهای چند پایه، اصول آموزشی در این کلاسها و ویژگی‌ها و محدودیت‌های حرفه‌ای در کلاسهای چند پایه لازم و ضروری است. از طرفی گسترش کلاسهای چند پایه از دوره ابتدایی به دوره‌ی راهنمایی تحصیلی مستلزم تحقیقاتی وسیع‌تر در راستای شناسایی و بررسی مشکلات روشهای موجود تدریس در این کلاسها می‌باشد. ارائه روشهای مناسب‌تر آموزش، شناسایی و رفع مشکلات آموزشی دبیران و لزوم کلاسهای ضمن خدمت، بررسی تطبیقی □ تحلیلی کلاسهای چند پایه در کشورهای مختلف دنیا از دیگر مؤلفه‌های سازنده در این زمینه خواهد بود.

```

function c=pseout(Q,w,epsilons,xrange,yrange,hx,hy)

% PSEOUT(Q,w,epsilon,xrange,yrange,hx,hy) estimates the spectrum and,
% for several epsilons, the boundaries of the epsilon-pseudospectra of
% a square matrix polynomial of the form
%
%
%      m      m-1
%      P(z) = A z  + A z  + ... + A z  + A
%              m      m-1      1      0
%
% with a nonsingular leading coefficient A .
%              m
%
% The algorithm is based on the construction of exclusion discs
% centered only at exterior points.
%
% (Note that the command tic-toc prints the execution time.)
%
% INPUT:  Q = {A0,A1,...,Am} array of the matrix-coefficients,
%          w = [w0,w1,...,wm] array of the corresponding weights,
%          epsilons = [eps1,eps2,...,epsN] array of epsilons,
%          xrange = [left bound on x-axis, right bound on x-axis],
%          yrange = [lower bound on y-axis, upper bound on y-axis],
%          hx = length of the grid for the x-axis,
%          hy = length of the grid for the y-axis.
%
% OUTPUT: c = contour plot points
%          (boundaries of the pseudospectra and eigenvalues of P(z)).
%
% Authors: Stavros Fatouros and Panayiotis Psarrakos,
%          Department of Mathematics,
%          National Technical University of Athens.
%
% Research supported by a grant of the EPEAEK project PYTHAGORAS II.
% The project is co-funded by the European Social Fund (75%) and
% Greek National Resources (25%).

tic

% detaching the epsilons (e1>=e2>=...>=eN)
epsilons=detach(epsilons);
N=length(epsilons);

% Setting sizes and the degree of the matrix polynomial P(z)
[t,n]=size(Q);      % here n=m+1=degree+1
degree=n-1;         % the degree of the matrix polynomial
[t,d]=size(Q{1});   % here d is the order of matrix-coefficients

% Reversing the array of weights
rw=[];

for j=1:n
    rw(1,j)=w(1,n-j+1);
end % for

% Constructing the grid
n1=round((xrange(2)-xrange(1))/hx);
n2=round((yrange(2)-yrange(1))/hy);

% Setting initial values on the grid
Z=zeros(n1+1,n2+1);
Iterations=0;
expoints=0;

% Computing factorials

```

```

par=1;

for j=1:degree
    par=par*j;
    p(j)=par;
end % for

% Main part of the program
for x=xrange(1):hx:xrange(2)
    I=round((x-xrange(1))/hx)+1;

    for y=yrange(1):hy:yrange(2)
        J=round((y-yrange(1))/hy)+1;
        mu=x+(y*i);
        absmu=abs(mu);

        if Z(I,J)==0
            r1=0;
            B0=Q{1};
            wm0=w(1);

            for j=1:degree
                B0=B0+(mu^j)*Q{j+1};
                wm0=wm0+(absmu^j)*w(j+1);
            end % for

            s=min(svd(B0));
            Z(I,J)=s/wm0;
            Iterations=Iterations+1;

            if Z(I,J)>epsilons(1)
                expoints=expoints+1;

                for k=1:n
                    Lamda0(k)=absmu^(k-1);
                end % for

                a=deriv(Q,absmu,p,degree);
                r1=wderiv1(rw,epsilons(1),s,wm0,degree,Lamda0);
                r1=r1-absmu;
                L=0;
                R=1;

                for k=1:11
                    g=(L+R)/2;
                    rg=r1*g;
                    lambda=absmu+rg;
                    a(n)=w(1);

                    for j=1:degree
                        a(n)=a(n)+w(j+1)*(lambda^j);
                    end % for

                    a(n)=-s+epsilons(1)*a(n);
                    star=a(n);

                    for j=1:degree
                        star=star+a(j)*(rg^(n-j));
                    end % for

                    if star>0
                        R=g;
                    else
                        L=g;
                    end % if
                end
            end
        end
    end
end

```

[illegible]