

CALCULERCV

```
function [polyz,polyt,varfixe,cv2]=calculercv(tps,haut,data)
%
% syntaxe [polyz,polyt,varfixe,cv2]=calculercv(tps,haut,data)
%
% input: tps : vecteur colonne des temps t
%      haut: vecteur colonne des hauteurs z
%      data: matrice des donnees avec z pour les lignes et tps pour les colonnes
%
% output: cv2: matrice  $cv=f'(t)/f''(z)$ ;
%      varfixe : temps correspondant a chaque cv
[nd,pd]=size(data);
% determination des polynomes de degre 3,  $u=f(z)$ 
varfixe=haut;
degre=3;
polyz=zeros(degre+1,pd);
for i=1:pd
    [pp,ss]=polyfit(varfixe,data(:,i),degre);
    polyz(:,i)=pp';
end;

% calcul des derivees secondes  $f''(z) = 6az+2b$ 
f2z=6*kron(polyz(1,:),haut)+2*kron(polyz(2,:),ones(length(haut),1));
f2z2=[];
indt=[];
for i=1:pd
    id=f2z(:,i)<0;
    if (sum(id)== nd | sum(id)== 0)
        f2z2=[f2z2,f2z(:,i)];
    end
end
```

```

        indt=[indt;i];
    end;
end;
% determination des polynomes de degre 2, u=f(t)
varfixe=tps(indt);
degre=2;
polyt=zeros(degre+1,nd);
for i=1:nd
    [pp,ss]=polyfit(tps,data(i,:)',degre);
    polyt(:,i)=pp';
end
% calcul des derivees premieres  $f'(t) = 2at+b$ 
f1t=2*kron(polyt(1,:),varfixe)+kron(polyt(2,:),ones(length(varfixe),1));

% calcul de cv
cv=f1t',/f2z2;
% élimination des cv incorrects
[nd,pd]=size(cv)
cv2=[];
indt=[];
for i=1:pd
    id=cv(:,i)<0;
    if (sum(id)== 0)
        cv2=[cv2,cv(:,i)];
        indt=[indt;i];
    end;
end;
varfixe=tps(indt);

```

