

```

function [GF,dGF,GRID]=GridAlgo(Gm,dG,dt,~,t)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Function to calculate the Grid algorithm
%% input
%[Vector] Gm Measurement (Length: k)
%[int] dG (deltaG) Maximum allowable ROC
%[int] dt sampling period
%[int] ~ tauF Isn't used since we use specific value for tauF
%[Vector] t timevector (Length: k)
%% Output
%[Vector] GRID with length k either 0 or 1 at the i'th entry.
%% Best values for parameters
tauF=6;%minutes
% den var 6 før
Gmin=130;%mg/dL
% Var før 130

Gmin3=1.5 ;%mg/dL/min
% den var 1.5 før

Gmin2=1.6;%mg/dL/min
% uændret

%% Inisializing
k=length(Gm);

dGF=zeros(1,k);
%% preprocessing section
GFNS=NoiseSpikeFilter(Gm,dG);

%% Low pass filter

GF=lowpassfilter(GFNS,dt,tauF,k);

%% Lagrangian interpolation

dGF(1)=GF(1);
dGF(2)=GF(2);
% dGF(1)=0;
% dGF(2)=0;
for i=3:k
    T1=(t(i)-t(i-1))/...
        ((t(i-2)-t(i-1))*(t(i-2)-t(i)));
    T2=(t(i)-t(i-2))/...
        ((t(i-1)-t(i-2))*(t(i-1)-t(i)));
    T3=(2*t(i)-t(i-2)-t(i-1))/...
        ((t(i)-t(i-1))*(t(i)-t(i-2)));
    dGF(i)=T1*GF(i-2)...
        +T2*GF(i-1)...
        +T3*GF(i);
end

%% GRID
GRID=zeros(1,k);
for i=3:k
    %if GF(i)>Gmin && (max((dGF(end-2:end)>Gmin3)) || max((dGF(end-1:end)>Gmin2)))
    LastThreeBool=dGF(i-2)>Gmin3&& dGF(i-1)>Gmin3&& dGF(i)>Gmin3;
    LastTwoBool=dGF(i-1)>Gmin2&& dGF(i)>Gmin2;
    % if GF(i)>Gmin && (max((dGF(i-2:i)>Gmin3))...
    % || max((dGF(i-1:i)>Gmin2)))
    % GRID(i)=1;
    % else

```

```
if GF(i)>Gmin && (LastThreeBool||LastTwoBool)
    GRID(i)=1;
else
    %Not necessary since GRID is a vector of zeros.
    GRID(i)=0;
end
end
end
```