```
function  [ Gfm_vec , filt_prev , flag, zero_one ] = GRID_func( ...
         delta_G , G_vec , tau, tspan , filt_prev , Gmin, Gfm_vec , t_vec, flag)
%
% GRID_func()
%
% DESCRIPTION:
% The function is a part of the GRID algortihm. This is part of the
% dectection logic, the last part of the algortihm.
% The function takes in the glucose measurements, calls the two filter
% functions and finds the derivatives. After this it is able to detect
% weather or not there a meal has been detected. Lastly, it counts down
% such that a meal will not be detected twice within two hours
%
% INPUT:
% delta_G               - The maximum ROC (rate of change)
%
% G_vec                 - Vector consisting of the glucose value and the
%                         two previous glucose measurements.
%                         As follows: [Gm-2, Gm-1, Gm].
%
% tspan                 - The interval step given as a number
%
% filt_prev             - Vector of previous filteret glucose measurements
%                         As follows: [G_{F,NS}(k-1), G_{F}(k-2)].
%                         For equation (1)&(3).
%
% Gmin                  - Vector of minumum glucose measurements
%                         As follows: [G_{min,1},G_{min,2},G_{min,3}].
%                         For equation (4).
%
% Gfm_vec               - Vector of previous derivatives
%                         As follows: [G'_{F}(k-2),G'{F}(k-2)].
%                         For equation (4).
%
% t_vec                 - Vector of sampling time respectively for G
%
% flag                  - For counting the time from last detected meal
%
% OUTPUT:
% Gfm_vec               - The stored new vector of the previous filtered
%                         glucose measurements.
%                         As follows: [G'_{F}(k-1),G'{F}(k)].
%
% G_prev                - The stored new vector of previous glucose
%                         measurements.
%                         As follows: [G_{F,NS}(k-2),G_{F}(k-2)].
%
% zero_one              - 1 or 0 for detected meal.
%
% flag                  - For counting the time from last detected meal
%
% PROJECT:
% Fagprojekt 2022
% A diabetes case study - Meal detection
%
% GENEREL:
% BSc                       : Mathematics and technology
% University                : The Technical University of Denmark (DTU)
% Department                : Applied Mathematics and Computer Science
%
% AUTHORS:
% Emma Victoria Lind
% Mariana de Sá Madsen
% Mona Saleem
%
% CONTACT INFORMATION
% s201205@student.dtu.dk
% s191159@student.dtu.dk
% s204226@student.dtu.dk
%
%

% Inisializing all values

Gfm_m2 = Gfm_vec(1);     % The second previous derivative used in euqation 4
Gfm_m1 = Gfm_vec(2);     % The previous derivative used in equation 4

Gfnsm2_prev = filt_prev(1); % The second previous noise-spike filtered value
                         % used in equation 1
Gfm2_prev   = filt_prev(2); % The second previous low filterd value used in
                         % equation 2

% Minimum values used in equation 4
Gmin1 = Gmin(1);
Gmin2 = Gmin(2);
Gmin3 = Gmin(3);

% The two previous measured gluscose values and the one at control state
Gm2     = G_vec(1);              % The second previous glucose value
Gm1     = G_vec(2);              % The previous glucose value
G       = G_vec(3);              % The glucose value at control state


% COMPUTING

% The noise-spike filter at the 3 sampling times
Gfnsm2 = spikefilt_func(Gm2,Gfnsm2_prev,delta_G);
Gfnsm1 = spikefilt_func(Gm1,Gfnsm2,delta_G);
Gfns   = spikefilt_func(G,Gfnsm1,delta_G);

% The low filter at the 3 sampling times
Gfm2 = lowfilt_func(tau,tspan,Gfnsm2,Gfm2_prev);
Gfm1 = lowfilt_func(tau,tspan,Gfnsm1,Gfm2);
Gf   = lowfilt_func(tau,tspan,Gfns,Gfm1);

% Inisializing input for estimate_lagrange
Gf_vec = [ Gfm2 , Gfm1 , Gf ];

% Computing the first derivative using lagrange
Gfm    = estimate_lagrange(t_vec,Gf_vec); % Returns the derivative

% The detection part from equation 4
if (Gf > Gmin1) && ...
   ((Gfm > Gmin3) && (Gfm_m1 > Gmin3) && (Gfm_m2 > Gmin3) ...
   || (Gfm > Gmin2) && (Gfm_m2 > Gmin2) )
```

```
    zero_one = 1; % A meal has been detected

else

    zero_one = 0; % No meal has been detected

end

% Output
filt_prev = [Gfnsm2,Gfm2]; % Outputting the updated filtered values
Gfm_vec = [Gfm_m1,Gfm]; % Outputting the updated derivative values


% COUNTING PART

if flag > 0

 % flag larger than 0 means that a meal has been detected within 120 min
 % implying that a meal should not be detected again already. So zero_one
 % is set to 0. Therefore, flag is subtracted by -1, such that it will
 % count down so a meal can be detetected again after 120 min.

    flag = flag-1;
    zero_one = 0;

elseif flag == 0 && zero_one == 1

  % flag equal to zero means a meal may be detected again
  % but only is if zero_one equals 1. When this happen flag start over
  % counting down 120 min.

    flag = 120/tspan;

end


end
```