```
function [localmat] = DuFun(x_pts, isub)

%
% This function computes, the current approximation for du
% at the requested x_pts points in subinterval isub.
% The vector of values is returned in localmat.
%
%


%%%%%%%%%%%%%%%%%%%%%% Global Variables %%%%%%%%%%%%%%%%%%
global xpts nnds
global Global_r  Global_s  Global_u
global rad_bas_type  str_bas_type  vel_bas_type
global quad_rul

%%%%
% We firstly need to determine the corresponding points on the
% reference interval.

% Description of subinterval.
xleft = xpts(isub) ;
xright = xpts(isub + 1) ;
hsub = xright - xleft ;


% Map the points to the reference triangle.
Rx_pts = (x_pts - xleft)/ hsub ;

% Evaluate Basis Functions and their Gradients at requested points.
[basvals, Gradten1] = feval(vel_bas_type, Rx_pts) ;

% Do appropriate scaling to get the true derivatives.
Gradten1 = Gradten1 / hsub ;

% Extract from the Global solution vector the coefficient values for
% the basis functions.
if strcmp(vel_bas_type, 'd1_CtsLin') == 1
   Vel1 = Global_u([isub isub+1],1) ;

   elseif strcmp(vel_bas_type, 'd1_CtsQuad') == 1
   Vel1 = Global_u([2*isub-1 2*isub 2*isub+1],1) ;

   elseif strcmp(vel_bas_type, 'd1_CtsCub') == 1
   Vel1 = Global_u([3*isub-2 3*isub-1 3*isub 3*isub+1],1) ;

end
```

Multiply the coefficients by the basis values.

```
localmat = Vel1.' * Gradten1 ;
```

Published with MATLAB® R2023b