```matlab
% DxMinus: calculate the backward differencing value along the x-axis
%
% Output parameters:
%   dxm = the backward differencing term (DxMinus) as a 2D matrix.
%
% Input parameters:
%   phi_i_j = the original function phi as a 2D matrix.
%   grid = the grid used for approximation

function dxm = DxMinus(phi_i_j, grid)

% To be able to use the matrix form to calculate DxMinus we need to get a
% matrix that is equivalent to phi_i_minus_1_j. This is basically shifting
% all rows in phi_i_j down by 1.
%
% Then, we extrapolate the values for the missing row by calculating the slope
% in the original phi_i_j as follows:
%       phi_i_j(1, y) - phi_i_j(2, y) for all y ranging from 1 to size(phi).
% Then the value of phi_i_minus_1_j(1, y) would be equal to the following:
%       phi_i_j(1, y) + slope(phi_i_j)
% Simplifying this formula gives us the value of phi_i_minus_1_j(1, y) =
%       2 * phi_i_j(1, y) - phi_i_j(2, y) for all y ranging from 1 to size(phi).
%
% Note:
% -----
% We use a similar formula for extrapolation in DxPlus and DyMinus and DyPlus.
phi_i_minus_1_j = circshift(phi_i_j, [1 0]);
phi_i_minus_1_j(1, :) = 2 * phi_i_j(1, :) - phi_i_j(2, :);

dxm = (phi_i_j - phi_i_minus_1_j) ./ grid.step(1);
```