```
% finiteDifference: calculate the value of phi_n_plus_1 based on the finite
% difference method.
%
% Output parameters:
%   resultingPhi = phi_n_plus_1
%
% Input parameters:
%   phi = phi_n
%   speed = constant speed
%   deltaT = time step
%   grid = the grid used for approximation

function resultingPhi = finiteDifference(phi, speed, deltaT, grid)

% Calculate gplus only if speed > 0, gminus only if speed < 0. Both are
% zero otherwise.
if(speed > 0)
    gplus = (max(DxMinus(phi, grid), 0).^2 + min(DxPlus(phi, grid), 0).^2 + ...
            max(DyMinus(phi, grid), 0).^2 + min(DyPlus(phi, grid), 0).^2).^0.5;
    gminus = 0;
elseif(speed < 0)
    gplus = 0;
    gminus = (max(DxPlus(phi, grid), 0).^2 + min(DxMinus(phi, grid), 0).^2 + ...
             max(DyPlus(phi, grid), 0).^2 + min(DyMinus(phi, grid), 0).^2).^0.5;
else
    gplus = 0;
    gminus = 0;
end

resultingPhi = phi - deltaT * (max(speed, 0) * gplus + min(speed, 0) * gminus);
```