```
function [accelX,accelY,dispXTSmooth,dispYTSmooth] =↙
func_smoothCalcAccel_NS(...
    dispX,dispY,timeStep,tempKern,tempOrder,padZeros,padRep)
% Author: Lloyd Fletcher
% PhotoDyn Group, University of Southampton
% Date Created: 18/9/2019
% Date Edited: 18/9/2019

% If this option is not specified
if nargin < 7
    padRep = false;
end

% Get the dimensions of the input fields
[sy,sx,st] = size(dispX);
numPts = sy*sx;

% Vectorise the displacements for use with the smoothing filter
dispXTSmooth = reshape(dispX,[numPts,st]);
dispYTSmooth = reshape(dispY,[numPts,st]);

% Pad the data with zeros in the beginning if required
if padRep
    dispXTSmooth = padarray(dispXTSmooth',floor↙
(tempKern/2),'replicate','pre');
    dispYTSmooth = padarray(dispYTSmooth',floor↙
(tempKern/2),'replicate','pre');
elseif padZeros
    dispXTSmooth = padarray(dispXTSmooth',floor(tempKern/2),0,'pre');
    dispYTSmooth = padarray(dispYTSmooth',floor(tempKern/2),0,'pre');
end

% Smooth the displacements in time using a Savitsky-Golay filter
if tempKern > 0
    dispXTSmooth = transpose(sgolayfilt(dispXTSmooth,tempOrder,↙
tempKern));
    dispYTSmooth = transpose(sgolayfilt(dispYTSmooth,tempOrder,↙
tempKern));
else
    dispXTSmooth = transpose(dispXTSmooth);
    dispYTSmooth = transpose(dispYTSmooth);
```

```
end

% Compute the derivatives using the gradient function
velX = gradient(dispXTSmooth,timeStep);
accelX = gradient(velX,timeStep);
velY = gradient(dispYTSmooth,timeStep);
accelY = gradient(velY,timeStep);

% Remove the initial padded zeros if they were added
if padZeros || padRep
    dispXTSmooth = dispXTSmooth(:, floor(tempKern/2) + 1:end);
    dispYTSmooth = dispYTSmooth(:, floor(tempKern/2) + 1:end);
    accelX = accelX(:, floor(tempKern/2) + 1:end);
    accelY = accelY(:, floor(tempKern/2) + 1:end);
end

% Reshape back to a 3D arrays to return variables
dispXTSmooth = reshape(dispXTSmooth, [sy,sx,st]);
dispYTSmooth = reshape(dispYTSmooth, [sy,sx,st]);
accelX = reshape(accelX, [sy,sx,st]);
accelY = reshape(accelY, [sy,sx,st]);

end
```