# Numerical Derivatives Using Savitsky-Golay Floating Cubic Interpolants

measure position variable xi at times ti (note that the subscript i refers to time not body number)

ti

ti+1

ti-1

ti-2

x

xi-2

xi-1

xi-3

t

xi+2

xi+1

xi+3

xi

ti+3

ti+2

ti-3

for fixed time step h, 

xi-2

xi-1

xi-3



xi+2

xi+1

xi+3

xi

0

+1

-1

-2

+3

+2

-3

postulate x = b0 + b1  + b2 2 + b3 3

 = ( b1 + 2 b2  + 3 b3 2 ) / h  = ( 2 b2 + 6 b3  ) / h2  = 6 b3 / h3

using values for xi  [ 1 i i2 i3 ]  



linear least-squares solution {} = ( [X]T [X] )-1 [X]T {Y}



interpolated values at  = 0 xi\*= b0 i\* = b1 / h i\* = 2 b2 / h2 i\* = 6 b3 / h3

xi\* = ( - 2 xi+3 + 3 xi+2 + 6 xi+1 + 7 xi + 6 xi-1 + 3 xi-2 - 2 xi-3 ) / 21

i\* = ( - 22 xi+3 + 67 xi+2 + 58 xi+1 - 58 xi-1 - 67 xi-2 + 22 xi-3 ) / 252 h

i\* = ( 5 xi+3 - 3 xi+1 - 4 xi - 3 xi-1 + 5 xi-3 ) / 42 h2

i\* = ( xi+3 - xi+2 - xi-1 + xi-1 + xi-2 - xi-3 ) / 6 h3

for first three values b0 = x4\* b1 = 4\* h b2 = 4\* h2 / 2 b3 = 4\* h3 / 6

x1\* = b0-3b1+9b2–27b3 1\* = (b1–6b2+27b3)/h 1\* = (2b2–18b3)/ h2 2\* = 4\*

x2\* = b0-2b1+4b2–8b3 2\* = (b1–4b2+12b3)/h 2\* = (2b2–12b3)/ h2 2\* = 4\*

x3\* = b0-b1+b2-b3 3\* = (b1–2b2+3b3)/h 3\* = (2b2–6b3)/ h2 3\* = 4\*

for last three values b0 = xn-3\* b1 = n-3\* h b2 = n-3\* h2 / 2 b3 = n-3\* h3 / 6

xn-2\* = b0+b1+b2+b3 n-2\* = (b1+2b2+3b3) / h n-2\* = (2b2+6b3) / h2 n-2\* = n-3\*

xn-1\* = b0+2b1+4b2+8b3 n-1\* = (b1+4b2+12b3) / h n-1\* = (2b2+12b3) / h2 n-1\* = n-3\*

xn\* = b0+3b1+9b2+27b3 n\* = (b1+6b2+27b3) / h n\* = (2b2+18b3) / h2 n\* = n-3\*

function [ p, v, a, j ] = filt\_7pt\_mat( x, h )

% Savitsky-Golay 7-point cubic interpolant and derivatives

% -3dB low-pass cutoff at 16% of sampling frequency

%

% USAGE

% [ p, v, a, j ] = filt\_7pt\_mat( x, h )

%

% INPUTS

% x - kxn matrix of raw samples

% k = number of coordinates - scalar k=1, 2D k=2, 3D k=3, etc.

% n = number of samples

% h - sampling interval

%

% OUTPUTS

% p - kxn position matrix

% v - kxn velocity matrix

% a - kxn acceleration matrix

% j - kxn jerk matrix

% HJSIII, 11.02.08 - tested under MATLAB v7.5

%

% H.J. Sommer III, Ph.D., Professor of Mechanical Engineering, 337 Leonhard Bldg

% The Pennsylvania State University, University Park, PA 16802

% (814)863-8997 FAX (814)865-9693 hjs1@psu.edu www.mne.psu.edu/sommer/

% number of samples

[ k, n ] = size(x);

nm1 = n - 1;

nm2 = n - 2;

nm3 = n - 3;

nm4 = n - 4;

nm5 = n - 5;

nm6 = n - 6;

% initialize

p = zeros(k,n);

v = zeros(k,n);

a = zeros(k,n);

j = zeros(k,n);

% Savitsky-Golay 7 point cubic interpolant coefficients

% deriv x(i+3) x(i+2) x(i+1) x(i) x(i-1) x(i-2) x(i-3) divisor

% 0 -2 +3 +6 +7 +6 +3 -2 21

% 1 -22 +67 +58 -58 -67 +22 252\*h

% 2 +5 -3 -4 -3 +5 42\*h\*h

% 3 +1 -1 -1 +1 +1 -1 6\*h\*h\*h

p(:,4:nm3) = ( -2\*x(:,7:n) +3\*x(:,6:nm1) +6\*x(:,5:nm2) +7\*x(:,4:nm3) ...

+6\*x(:,3:nm4) +3\*x(:,2:nm5) -2\*x(:,1:nm6) ) /21;

v(:,4:nm3) = ( -22\*x(:,7:n) +67\*x(:,6:nm1) +58\*x(:,5:nm2) ...

-58\*x(:,3:nm4) -67\*x(:,2:nm5) +22\*x(:,1:nm6) ) /252 /h;

a(:,4:nm3) = ( +5\*x(:,7:n) -3\*x(:,5:nm2) -4\*x(:,4:nm3) ...

-3\*x(:,3:nm4) +5\*x(:,1:nm6) ) /42 /h /h;

j(:,4:nm3) = ( x(:,7:n) -x(:,6:nm1) -x(:,5:nm2) ...

+x(:,3:nm4) +x(:,2:nm5) -x(:,1:nm6) ) /6 /h /h /h;

% first three

b0 = p(:,4);

b1 = v(:,4)\*h;

b2 = a(:,4)\*h\*h/2;

b3 = j(:,4)\*h\*h\*h/6;

p(:,1:3) = [ b0-3\*b1+9\*b2-27\*b3 b0-2\*b1+4\*b2-8\*b3 b0-b1+b2-b3 ];

v(:,1:3) = [ b1-6\*b2+27\*b3 b1-4\*b2+12\*b3 b1-2\*b2+3\*b3 ]/h;

a(:,1:3) = [ 2\*b2-18\*b3 2\*b2-12\*b3 2\*b2-6\*b3 ]/h/h;

j(:,1:3) = [ j(:,4) j(:,4) j(:,4) ];

% last three

b0 = p(:,nm3);

b1 = v(:,nm3)\*h;

b2 = a(:,nm3)\*h\*h/2;

b3 = j(:,nm3)\*h\*h\*h/6;

p(:,nm2:n) = [ b0+b1+b2+b3 b0+2\*b1+4\*b2+8\*b3 b0+3\*b1+9\*b2+27\*b3 ];

v(:,nm2:n) = [ b1+2\*b2+3\*b3 b1+4\*b2+12\*b3 b1+6\*b2+27\*b3 ]/h;

a(:,nm2:n) = [ 2\*b2+6\*b3 2\*b2+12\*b3 2\*b2+18\*b3 ]/h/h;

j(:,nm2:n) = [ j(:,nm3) j(:,nm3) j(:,nm3) ];

return

% bottom of filt\_7pt\_mat.m



% freq\_7pt.m - frequency response for 7-point cubic interpolant

% HJSIII, 03.04.30

clear

% j\*2\*pi\*f/fs

fdfs = ( 0 : 0.001 : 0.5 )';

t = j \* 2 \* pi \* fdfs;

% transfer function

G = ( -2\*exp(3\*t) +3\*exp(2\*t) +6\*exp(t) +7 +6\*exp(-t) +3\*exp(-2\*t) -2\*exp(-3\*t) ) / 21;

amp = abs( G );

dB = 20 \* log10( amp );

phi = unwrap( angle( G )) \* 180 / pi;

figure( 1 )

subplot( 2,1,1 )

plot( fdfs,dB )

ylabel( 'Gain [dB]' )

title( '7-point floating cubic interpolant' )

subplot( 2,1,2 )

plot( fdfs,phi )

xlabel( 'f / fs' )

ylabel( 'Phase [deg]' )