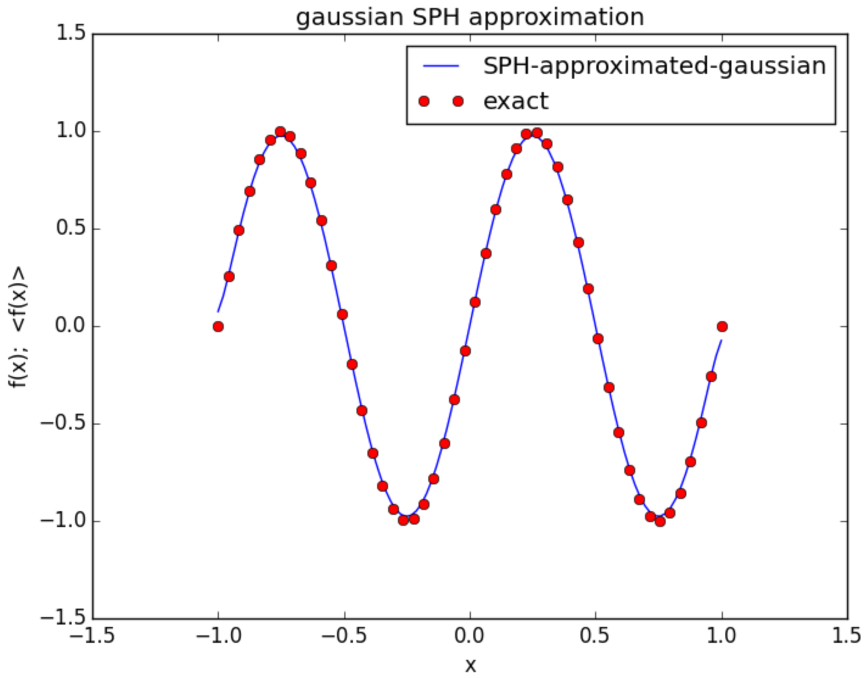
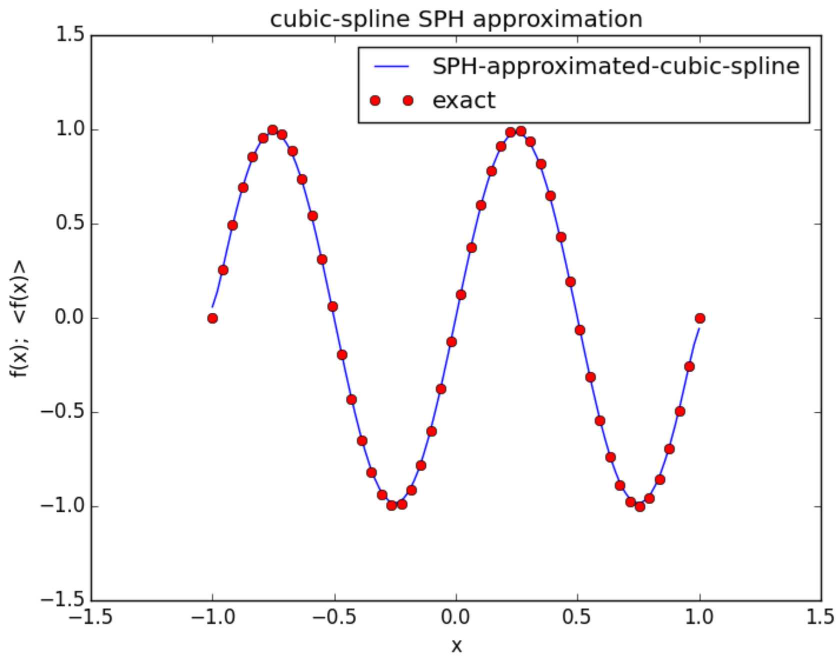
Assignment 6 - SPH Approximation
by Naveen Himthani
120010001

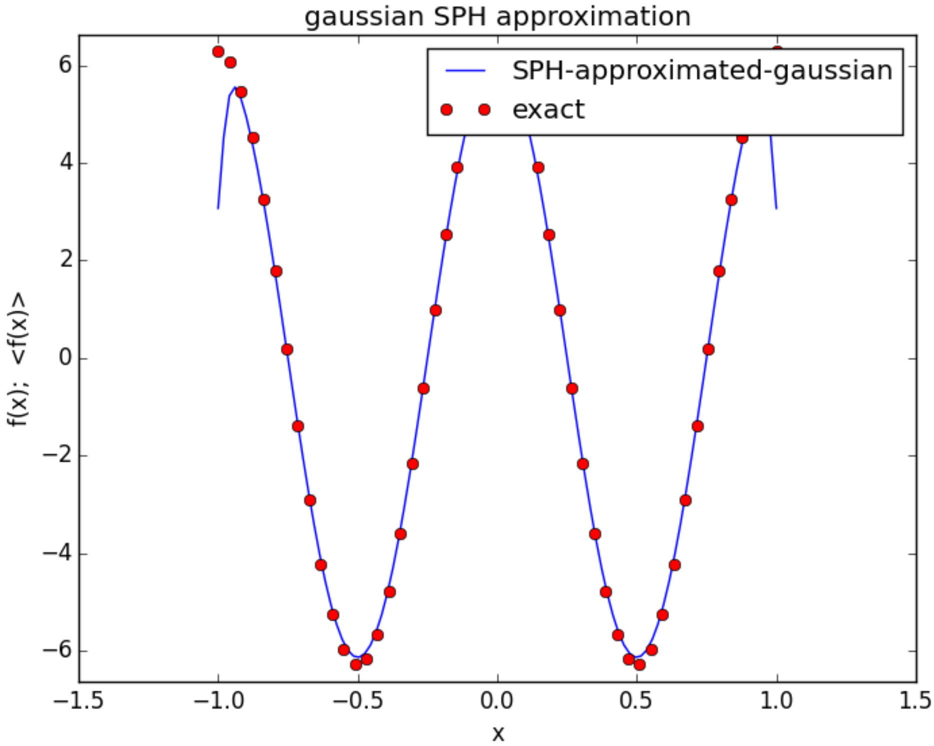
1. SPH approximation using Gaussian Kernel (h_factor=1.2; n_base=50; n_interp=100)



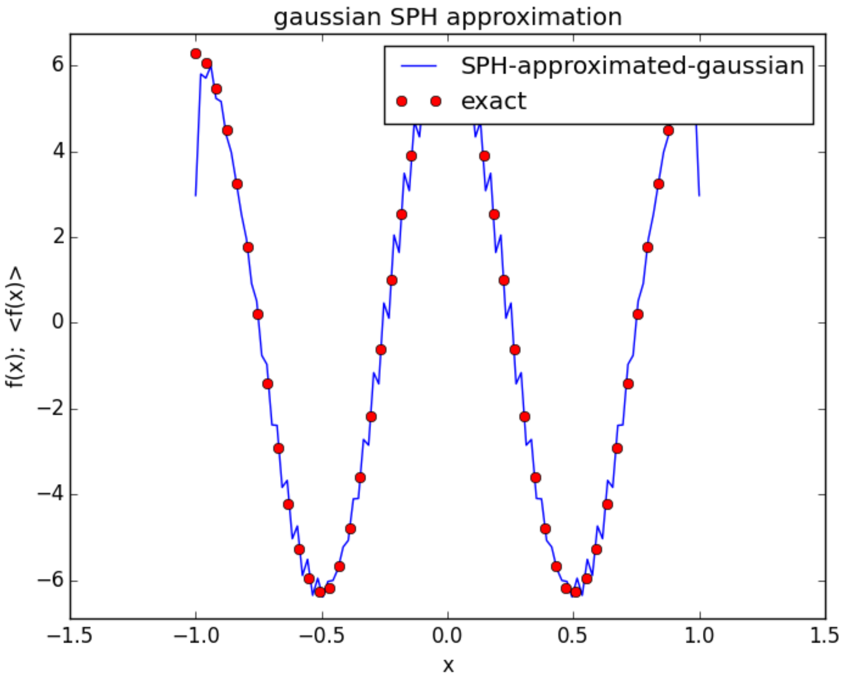
2) Cubic - Spline Kernel (h_factor=1.2; n_base=50; n_interp=100)



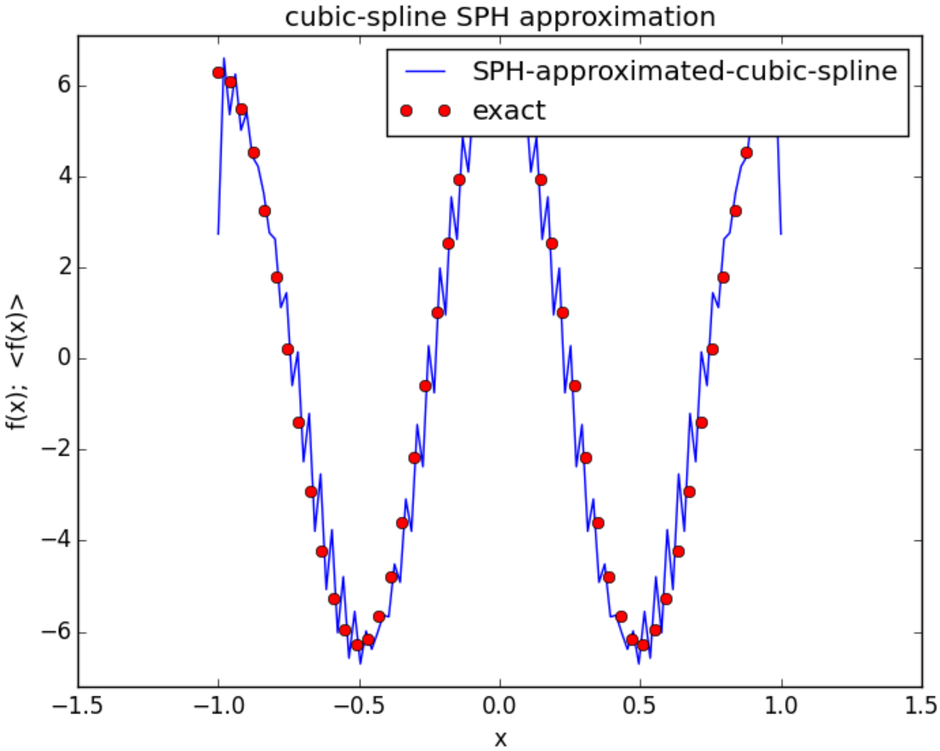
3) Gaussian Kernel - derivative (h_factor = 1.2; n_base=50; n_interp=100)



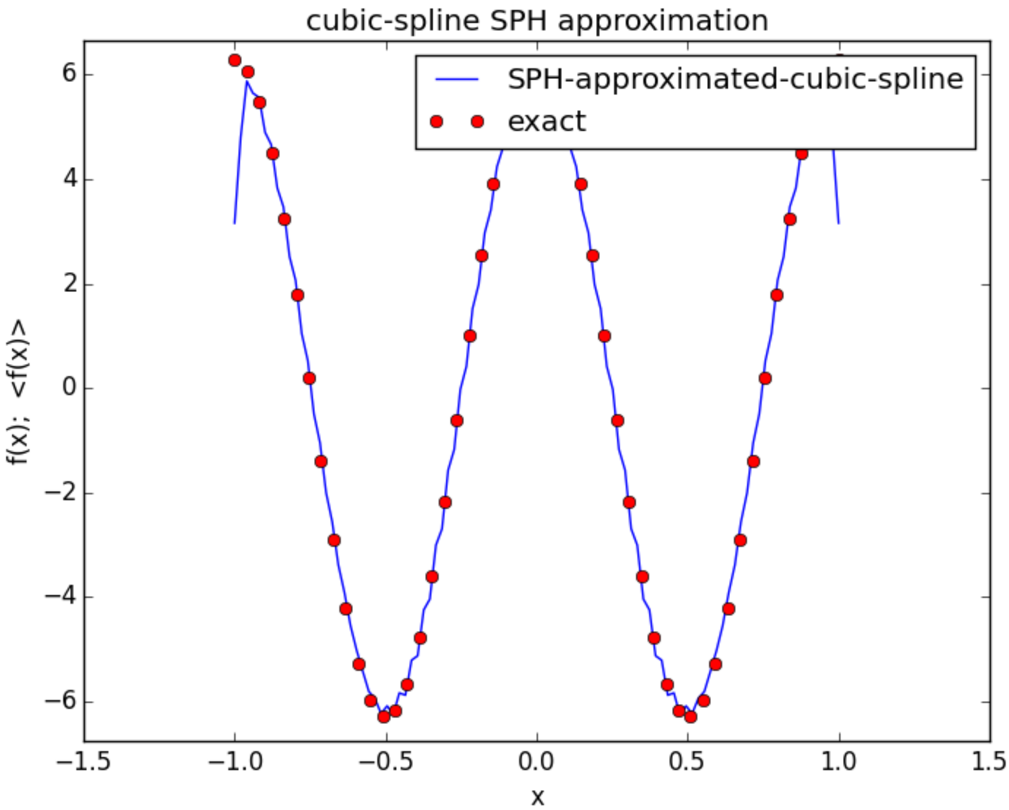
4) Gaussian Kernel - derivative (h = 0.8; n_base = 50; n_interp = 100)



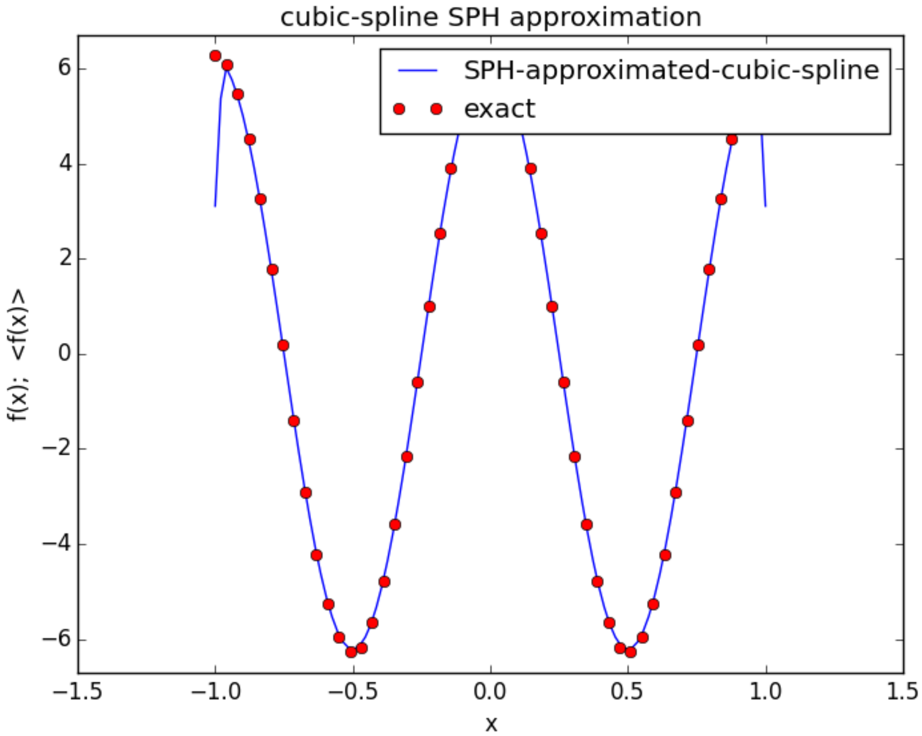
5) Spline derivative (h_factor = 0.8; n_base = 50; n_interp = 100)



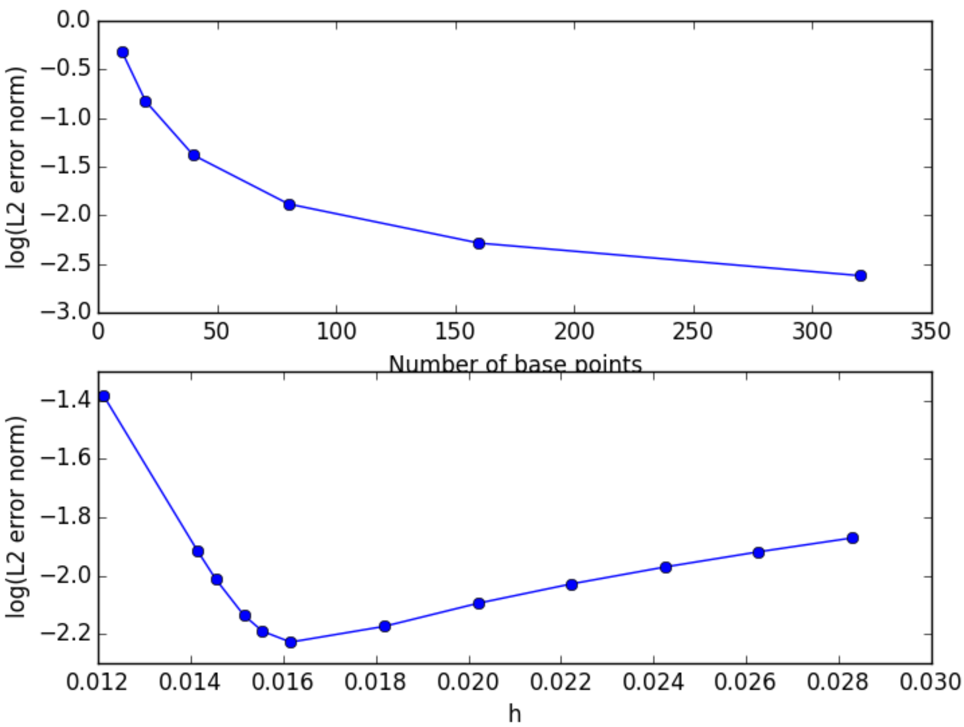
6) Spline derivative (h_factor = 1.2; n_base = 50; n_interp = 100)



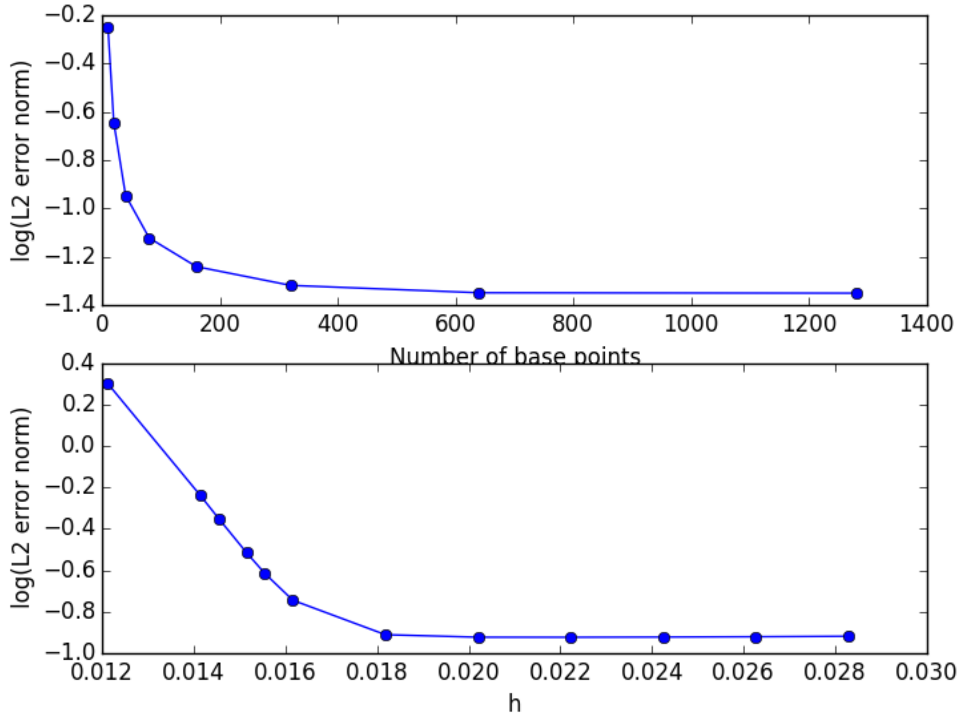
7) Spline derivative (h_factor = 1; n_base=50; n_interp = 100)



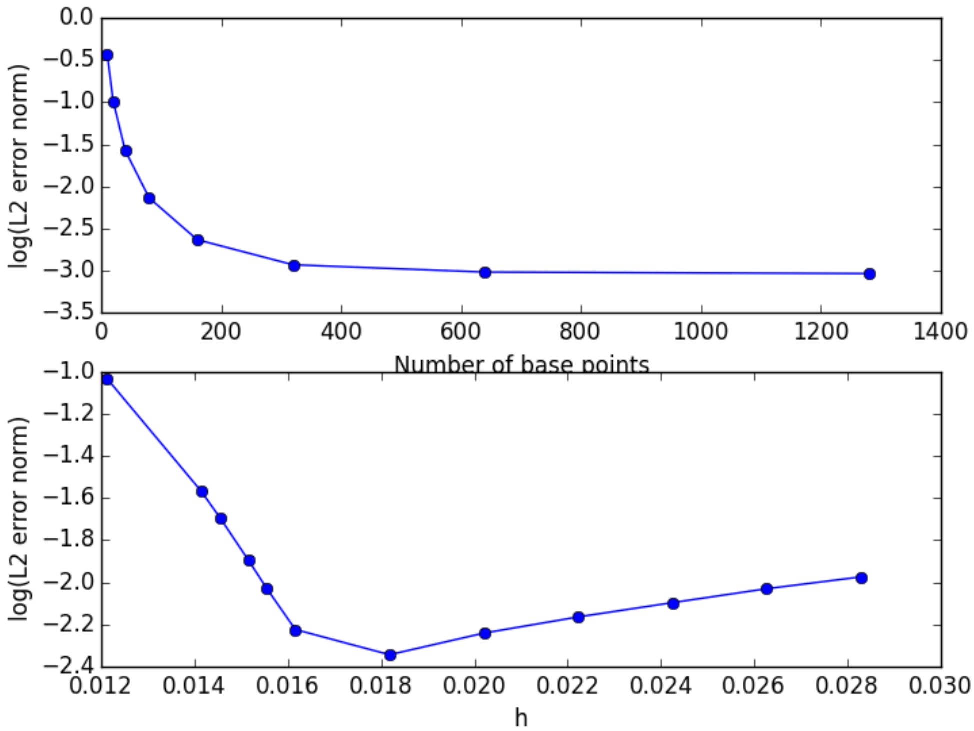
8) Gaussian - no derivative : error plot (n_interp = 100; h_factor = 1.2)



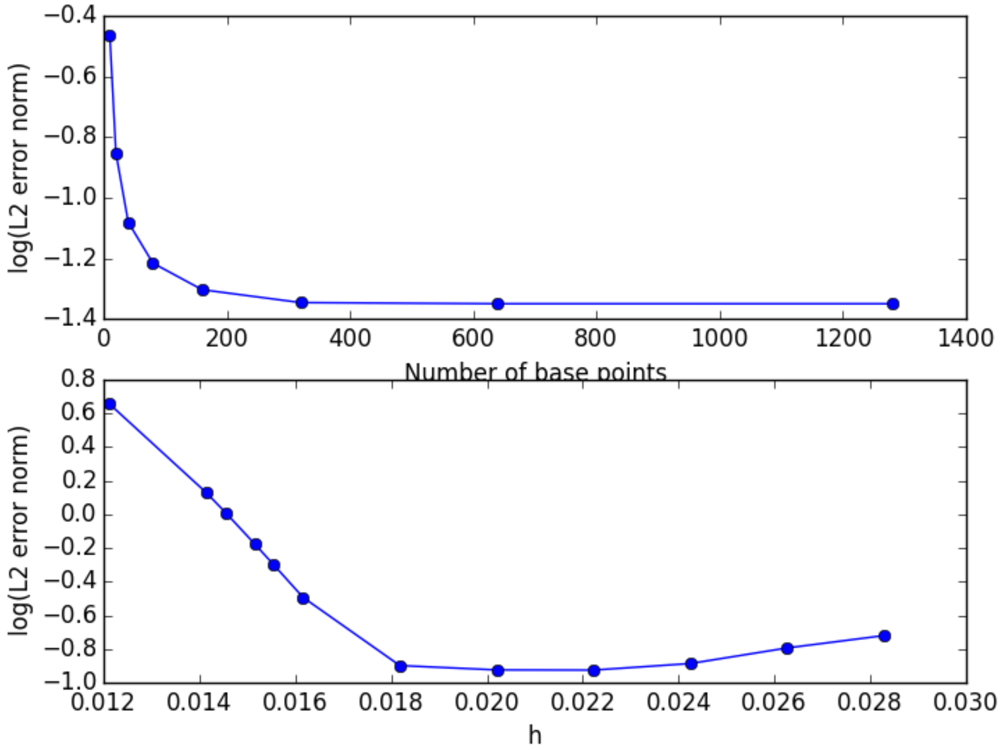
9) Gaussian - derivative : error plot (h_factor =1.2; n_interp = 500)



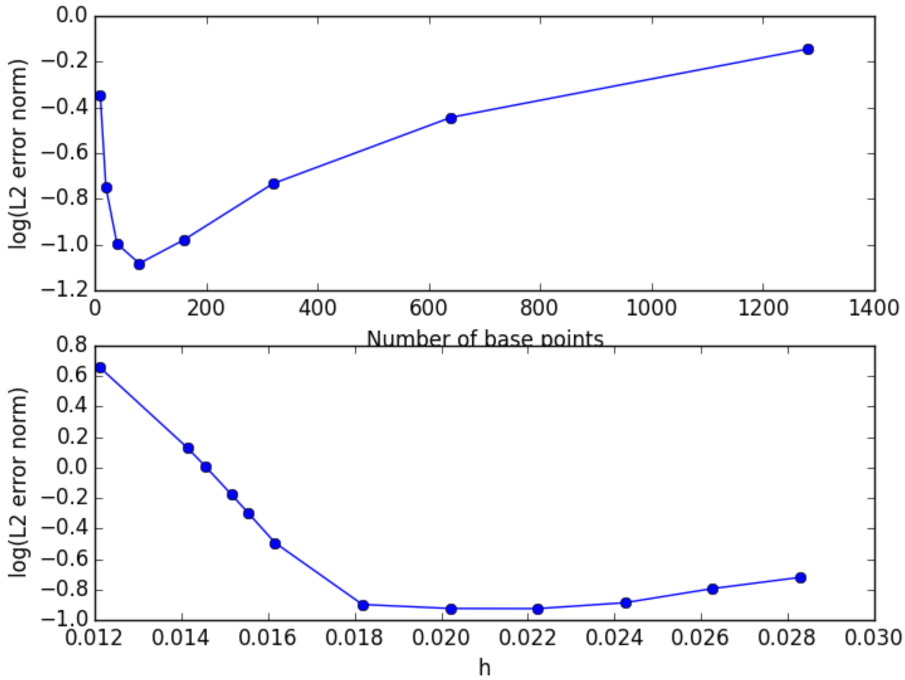
10) Spline - no derivative : error plot (h_factor = 1.2; n_interp = 100)



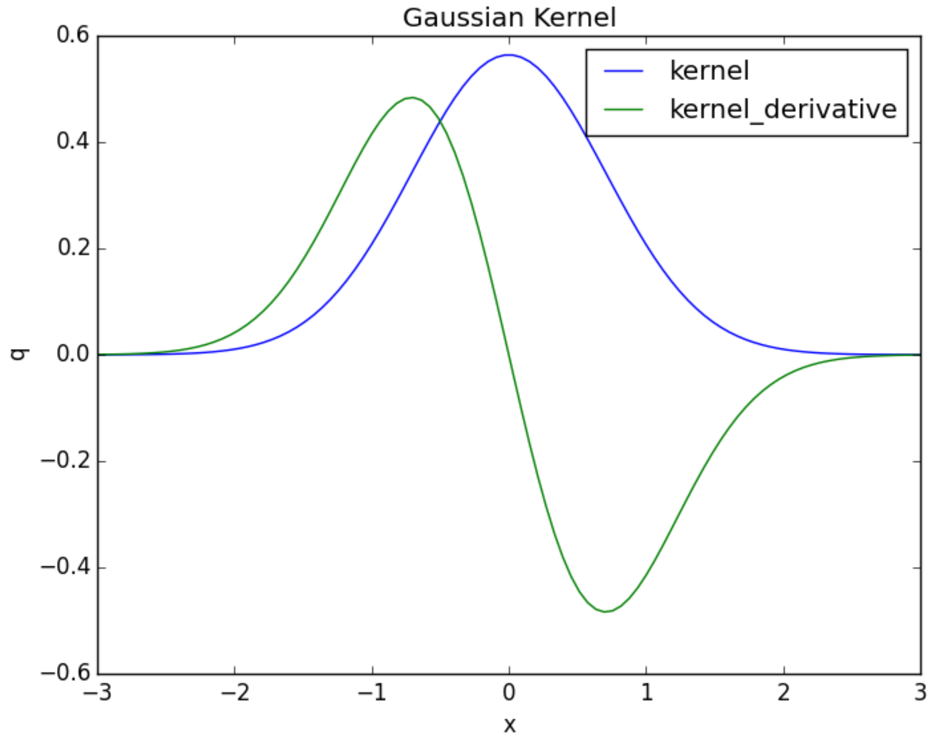
11) Spline derivative - error plot (h_factor = 1; n_interp= 500)



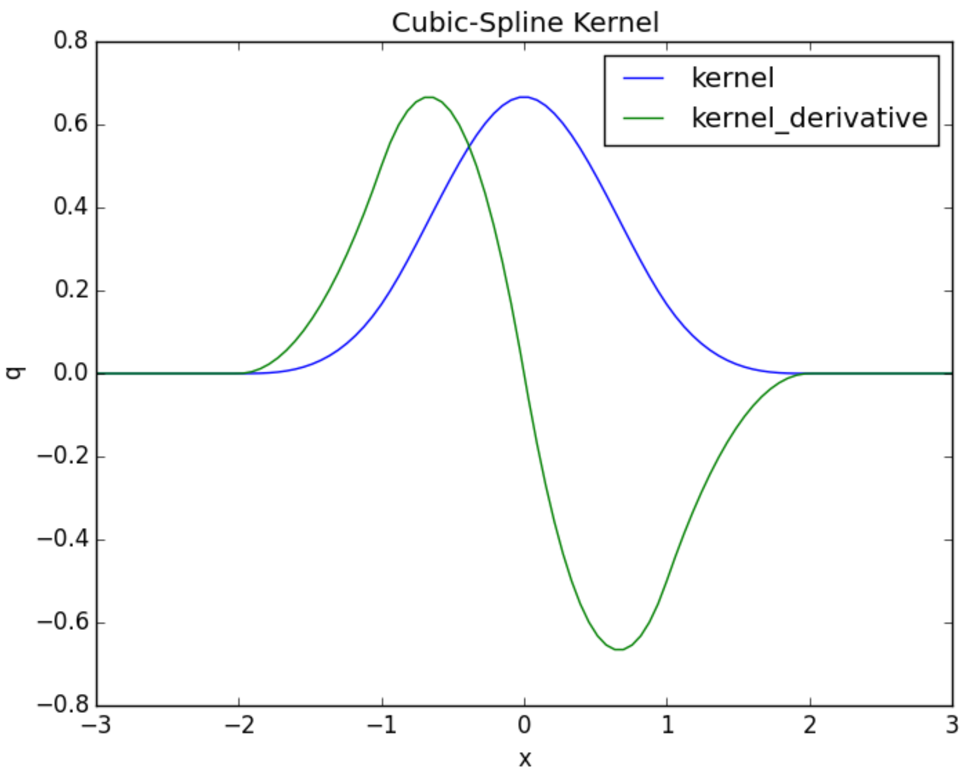
12) Spline derivative - error plot (h_factor = 1.2; n_interp = 500)



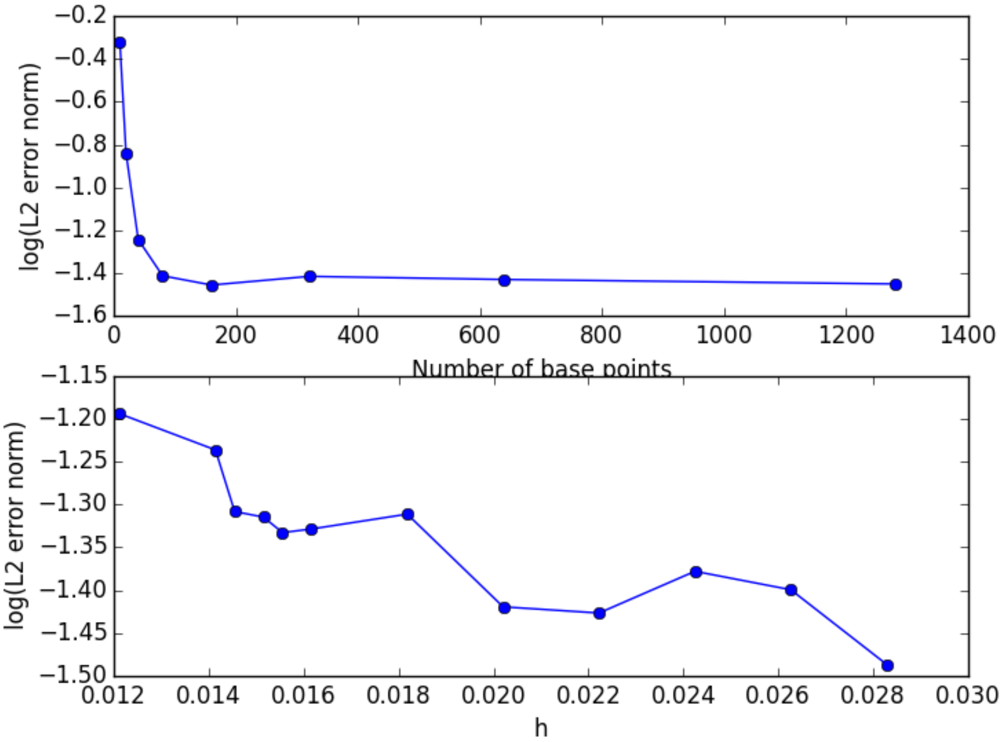
13) Gaussian Kernel and derivative



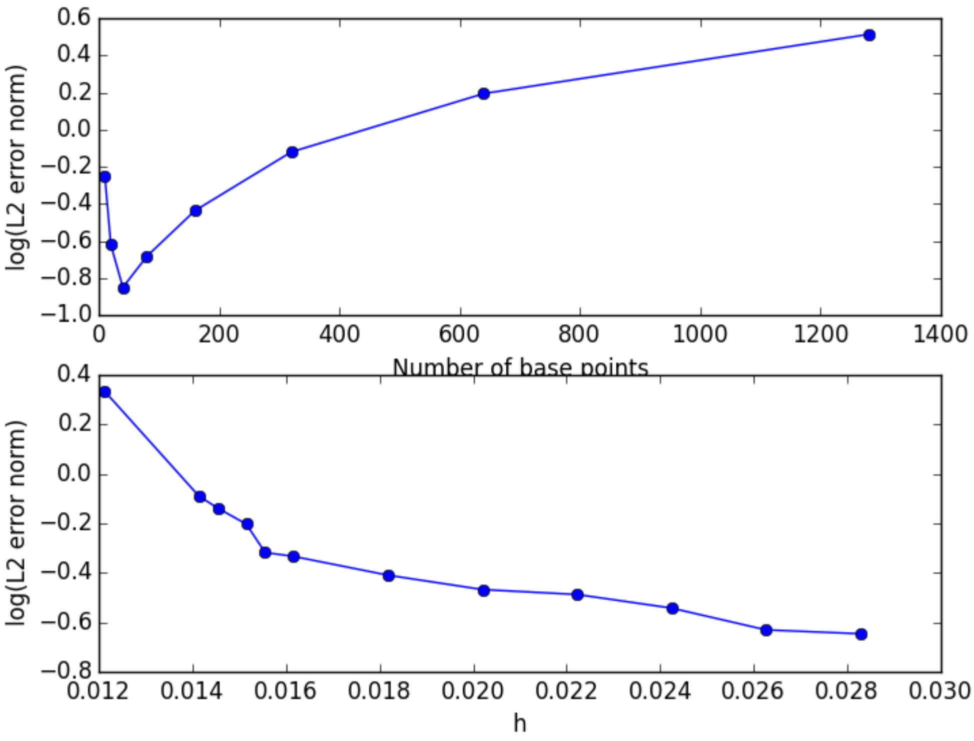
14) Spline kernel and derivative



15) Gaussian - with noise (variance = 0.002, mean = 0, normally distributed)
(h_factor = 1.2; n_interp = 500)



16) Gaussian derivative - with noise (variance = 0.002, mean = 0, normally distributed)(h_factor = 1.2; n_interp = 500)



observations:

1. High frequency noise is noted in the SPH approximation of function using gaussian kernel at h_factor < 1, similarly for spline kernel
2. High frequency noise is noted in the SPH approximation of the derivative of the function using the spline kernel at h_factor other than 1. For h_factor other than 1, we observe high frequency noise with increasing h and decreasing h for same n_base and n_interp
3. in the derivative approximation, the function drops of at the edges because, the edge effect is not taken into account. We need to copy the function as it is on both he sides of the domain to take care of this.
4. In the error plots, the L2-error decreases as the number of base points decreases in all cases except the one where spline is used for derivative approximation and h other than one is used
5. There is an optimum h value for function approximation using any kernel
6. The derivative is well approximated using the spline kernel with h_factor = 1
7. The kernel plots are added for visualisation
8. When random noise is added the same effect on the error with increasing n_base is seen, where as for the increasing h value case, it decreases instead of increasing