Lect. 5. Precise numerical integration and differentiation

Conventional numerical integration methods:

Trapezoidal rule: $\bar{a}_1^{(T)} = 0, \quad \bar{a}_k^{(T)} = \bar{a}_{k-1}^{(T)} + (a_{k-1} + a_k)/2$

Simpson rule: $\bar{a}_1^{(S)} = 0, \ \bar{a}_k^{(S)} = \bar{a}_{k-2}^{(S)} + (a_{k-2} + 4a_{k-1} + a_k)/3$

3/8 Simpson rule: $\bar{a}_0^{(3/8S)} = 0, \ \bar{a}_k^{(3/8S)} = \bar{a}_{k-3}^{(3/8S)} + 3(a_{k-3} + 3a_{k-2} + 3a_{k-1} + a_k)/8$

Boole's rule $\bar{a}_0^{(Bl)} = 0, \ \bar{a}_k^{(Bl)} = \bar{a}_{k-4}^{(Bl)} + 2(7a_{k-4} + 32a_{k-3} + 12a_{k-2} + 32a_{k-1} + 7a_k)/45$

Weddle's Rule $\bar{a}_0^{(Wdl)} = 0, \ \bar{a}_k^{(Wdl)} = \bar{a}_{k-6}^{(Wdl)} + 3(a_{k-6} + 5a_{k-5} + a_{k-4} + 6a_{k-3} + a_{k-2} + 5a_{k-1} + a_k)/10$

Hardy's rule: $\bar{a}_0^{(Hrd)} = 0, \ \bar{a}_k^{(Hrd)} = \bar{a}_{k-6}^{(Hrd)} + (a_{k-6} + 162a_{k-5} + 220a_{k-4} + 162a_{k-1} + 28a_k)/100$

Cubic spline: $\bar{a}_{k+1}^{(CS)} - \bar{a}_k^{(CS)} = \frac{1}{2}(a_k + a_{k+1}) - \frac{1}{24}(m_k + m_{k+1}) \ \{(m_{k-1} + 4m_k + m_{k+1}) = 6(a_{k+1} - 2a_k + a_{k-1})\}$

Discrete frequency responses of 4 numerical integration methods ($\eta_0^{(T)} = 0$):

Trapezoidal rule: $\eta_r^{(T)} = i\cos(\pi r/N)/2\sin(\pi r/N), \quad r = 1,...,N-1;$

Simpson rule: $\eta_r^{(S)} = i[\cos(2\pi r/N) + 2]/3\sin(2\pi r/N), \quad r = 1,...,N-1;$

3/8 Simpson rule: $\eta_r^{(3S)} = i[\cos(3\pi r/N) + 3\cos(\pi r/N)]/\sin(3\pi r/N), \ r = 1,...,N-1;$

Cubic spline integration: $\eta_r^{(CS)} = i\cos\left(\frac{\pi r}{N}\right)\left[1 + \frac{3}{\cos(2\pi r/N) + 2}\right] \Big/ 4\sin\left(\frac{\pi r}{N}\right), r = 1,..,N-1;$

The ideal continuous integrator frequency response: $H_{\text{int}}(f) = i/2\pi f$

FFT based numerical integrators: Even N / Odd N

$$\eta_r^{(\text{int})} = \eta_{N-r}^* = \begin{cases} 0, & r = 0 \\ iN/2\pi r, & r = 1,2...,N/2-1 \\ -1/2\pi, & r = N/2 \end{cases} ; \quad \eta_r^{(\text{int})} = \eta_{N-r}^* = \begin{cases} 0, & r = 0 \\ iN/2\pi r, & r = 1,2,...,(N-1)/2 \end{cases}$$

DFT-based integrator: $\{\bar{a}_k\} = \text{IDFT}\{\eta_r^{(\text{int})} \bullet \text{DFT}\{a_k\}\}$

DCT-based integrator: $\tilde{a}_k = \begin{cases} a_k, & k = 0,1,...,N-1 \\ a_{2N-1-k}, & k = N,...,2N-1 \end{cases} \quad \eta_r^{(\text{int})} = \begin{cases} 0, & r = 0 \\ iN/\pi r, & r = 1,2...,N-1 \\ -1/\pi, & r = N \\ \eta_{N-r}^*, & r = N+1,...,2N-1 \end{cases}$

$$\bar{a}_k = -\frac{\sqrt{N}}{\pi\sqrt{2}}\sum_{r=1}^{N-1}\frac{\alpha_r^{(DCT)}}{r}\sin\left(\pi\frac{k+1/2}{N}r\right) = -(-1)^k\frac{\sqrt{N}}{\pi\sqrt{2}}\sum_{r=1}^{N-1}\frac{\alpha_{N-r}^{(DCT)}}{N-r}\cos\left(\pi\frac{k+1/2}{N}r\right)$$

Numerical differentiation methods: $\dot{a}_k = \sum_{n=0}^{N_h-1} h_n^{diff} a_{k-n}$

Conventional numerical differentiation methods:

D1: $h_n^{diff(1)} = [-1, 1];$ $\Rightarrow$ $\eta_r^{diff(1)} \propto \sin(\pi r/N);$

DD1: $\bar{h}_n^{diff(1)} = ([-1, 1, 0] + [0, -1, 1])/2$ $\Rightarrow$ $\bar{\eta}_r^{diff(1)} \propto \sin(2\pi r/N)$

D2: $h_n^{diff(2)} = [-1/12, 8/12, 0, -8/12, 1/12];$ $\Rightarrow$ $\eta_r^{diff(2)} \propto \frac{8\sin(2\pi r/N) - \sin(4\pi r/N)}{12}$

Frequency response of the ideal continuous differentiator: $H_{diff}(f) = -i2\pi f$

FFT based differentiation method (ramp-filtering): $\{\dot{a}_k\} = \text{IDFT}(\{\eta_r^{diff}\} \cdot \text{DFT}(\{a_k\}))$

Odd N: / Even

$$\eta_r^{diff} = \begin{cases} -i2\pi r/N, & r = 0,1,...,(N-1)/2 \\ i2\pi(N-r)/N, & r = (N+1)/2,...,N-1 \end{cases} \quad \eta_r^{diff} = \begin{cases} -i2\pi r/N, & r = 0,1,...,N/2-1 \\ -\pi/2, & r = N/2 \\ i2\pi(N-r)/N, & r = N/2+1,...,N-1 \end{cases}$$

DCT-based differentiation:

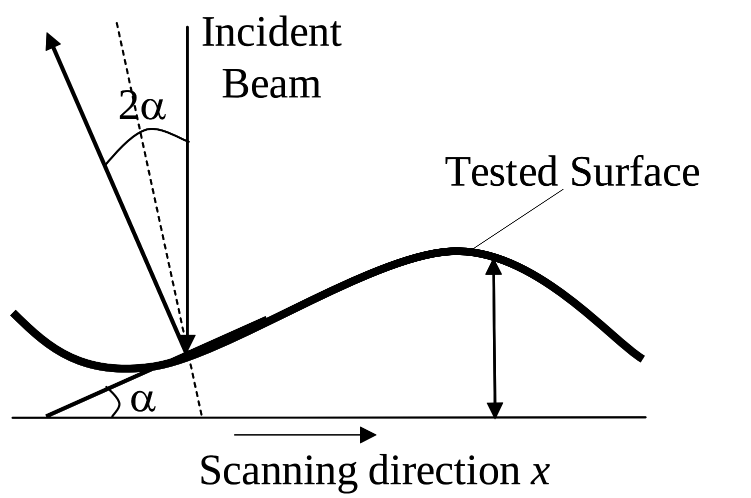
$$a_k^{(diff)} = -\frac{2\pi}{N\sqrt{2N}}\left\{\sum_{r=1}^{N-1} r\alpha_r^{(DCT)}\sin\left(\pi\frac{k+1/2}{N}r\right)\right\} = (-1)^{k+1}\frac{2\pi}{N\sqrt{2N}}\left\{\sum_{r=1}^{N-1}(N-r)\alpha_{N-r}^{(DCT)}\cos\left(\pi\frac{k+1/2}{N}r\right)\right\}$$

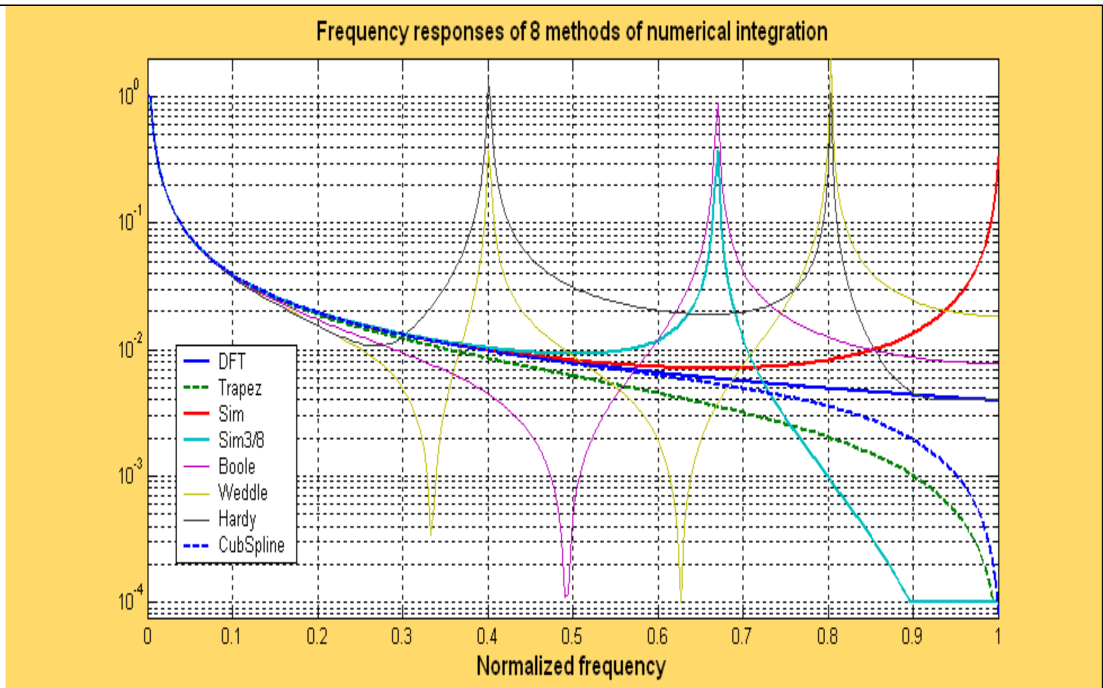
Integration and differentiation as an inverse problem: $H_{opt}(f) = H_{inverse}(f)\frac{SNR(f)}{SNR(f)+1}$

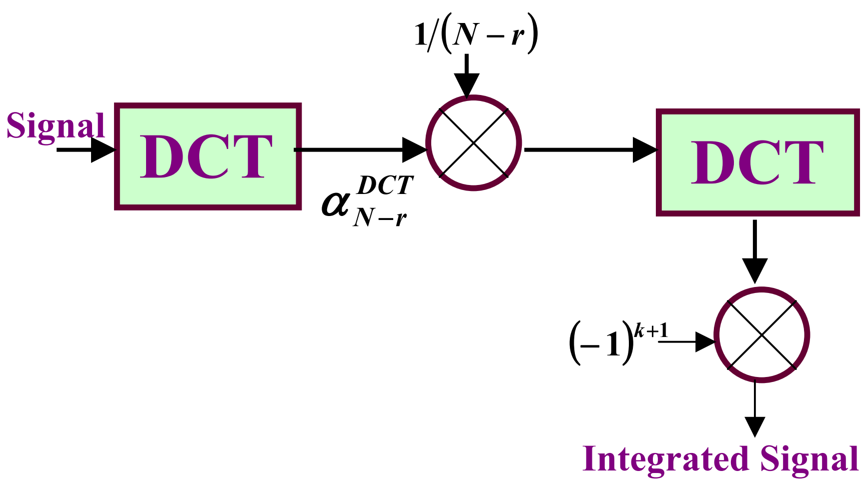
Filtered back projection algorithm for image reconstruction from projections

Numerical integration in optical metrology

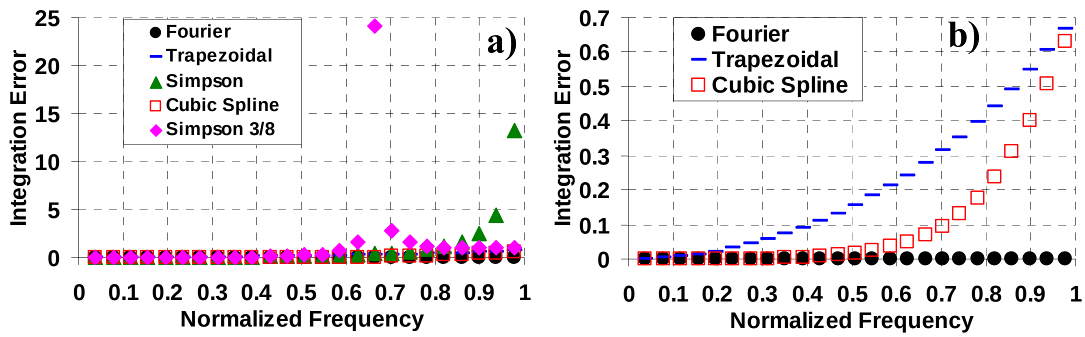
Laser deflectometry is a technique of measuring profile of surfaces. It is based on measuring deviation of the incident light caused by its reflection from the surface. This deviation contains the slope data information of the profile of the test surface. The surface profile can then be obtained by integration of the slope data.



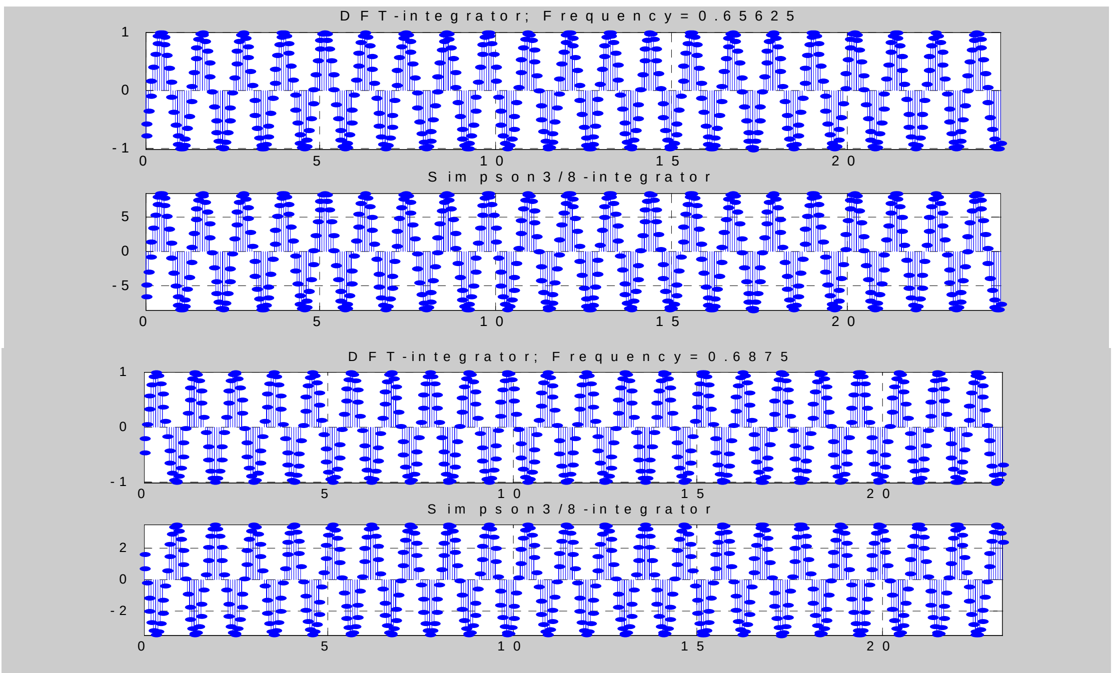




DCT-based integration algorithm

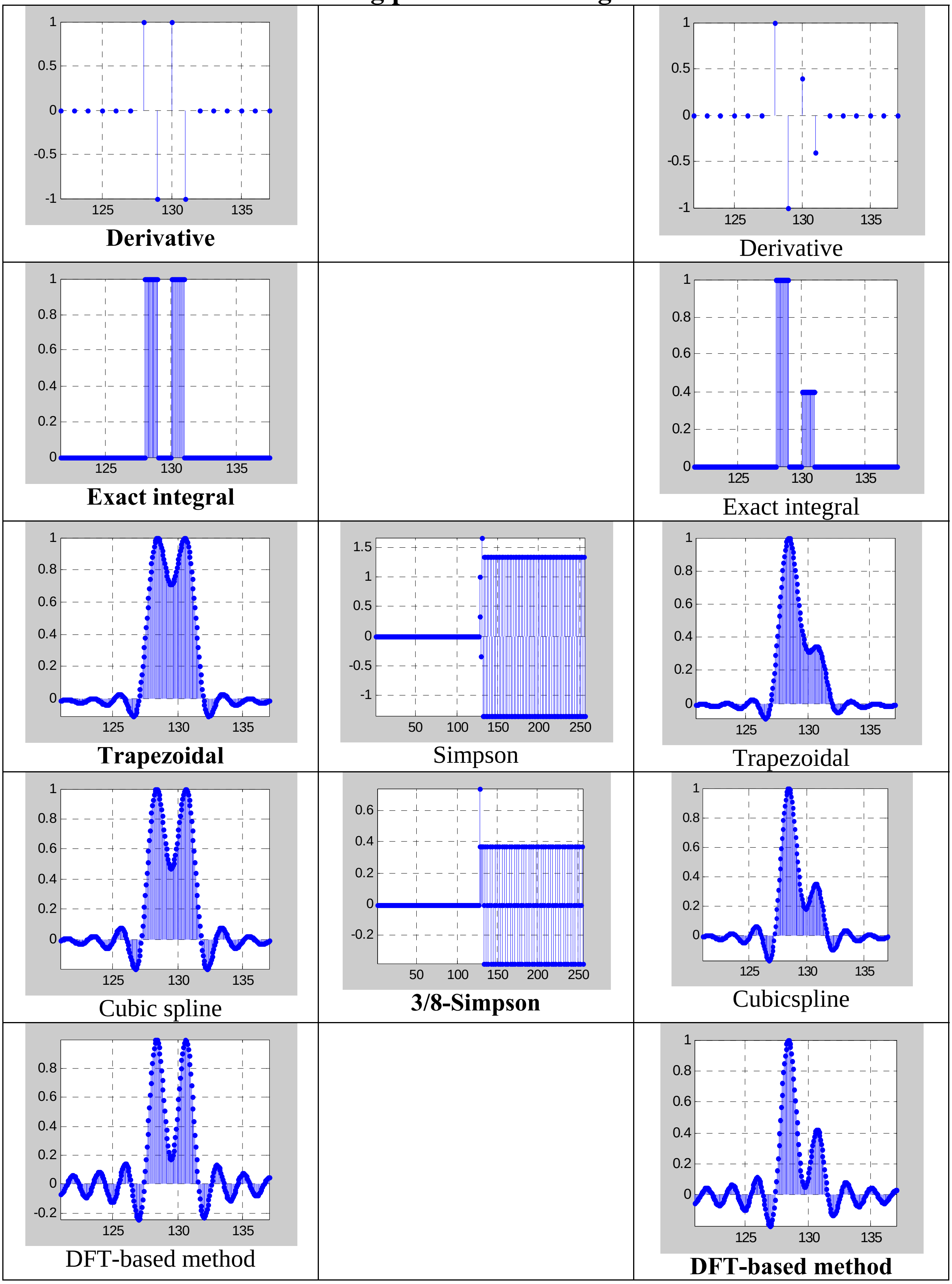


Integration error of periodic sinusoidal signals as a function of the normalized frequency: (a) for all methods; (b) only for DFT-based, trapezoidal and cubic spline methods



Phase inversion phenomenon for 3/8-Simpson method

Resolving power of the integrators



Numerical differentiation in optical metrology and video processing: "Optical flow" computation

The principle of optical flow computation. Let $I(x,y,t)$ be image intensity defined in spatial (x,y) and time t coordinates and during time interval Δt pixel (x,y) moves, due to the object motion, to point $(x+\Delta x, y+\Delta y)$. Let also assume that the object motion causes no changes in pixel intensity, and the changes may occur solely due to random factors such as additive signal independent white Gaussian noise that can be attributed to image sensor. Then, given image intensity measurements $I(x,y,t)$ and $I(x+\Delta x, y+\Delta y, t+\Delta t)$ in two time moments t and $t+\Delta t$, statistically optimal maximum likelihood estimation of movement vector $(\Delta x, \Delta y)$ is found as a solution of the equation:

$$(\Delta x, \Delta y, t) = \underset{(\Delta x, \Delta y)}{\mathbf{arg\,min}} \iint_{(\xi,\eta)\in ARM} \left[I(\xi,\eta,t) - I(\xi+\Delta x, \eta+\Delta y, t+\Delta t)\right]^2 d\xi d\eta$$

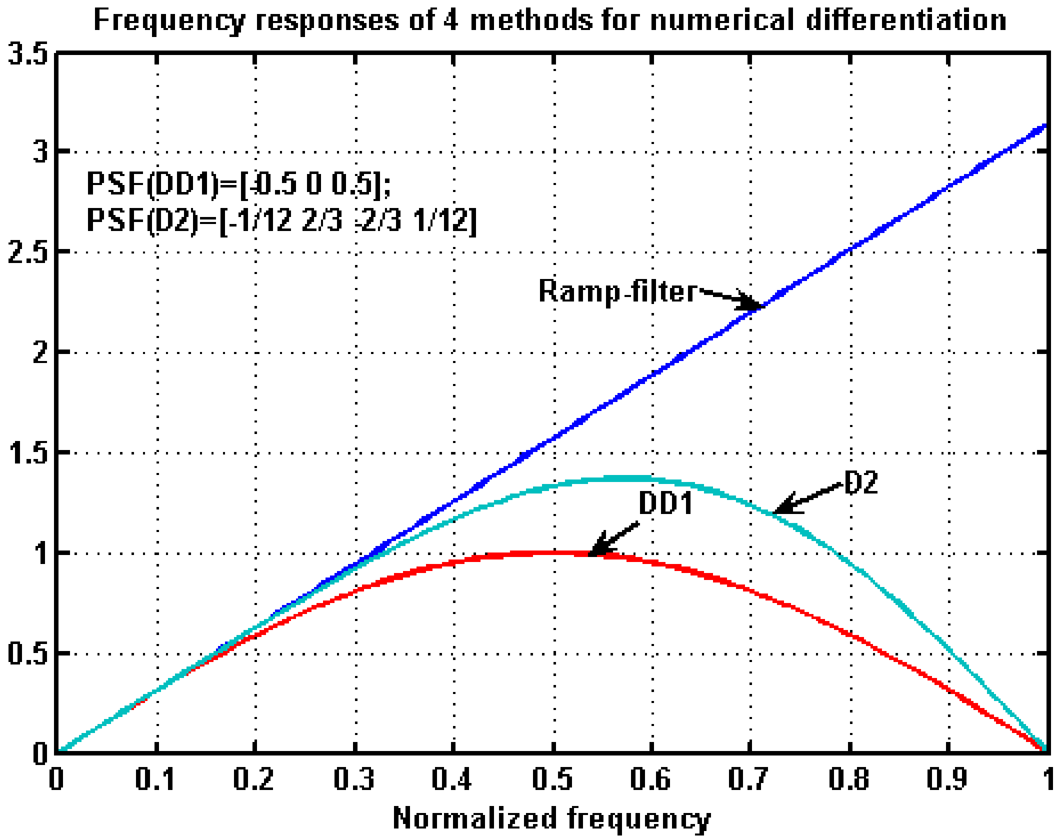
where $ARM(x,y)$ is the object Area of Rigid Motion centered at the point (x,y). Within the accuracy of the Taylor series expansion of the image intensity function $I(x,y,t)$:

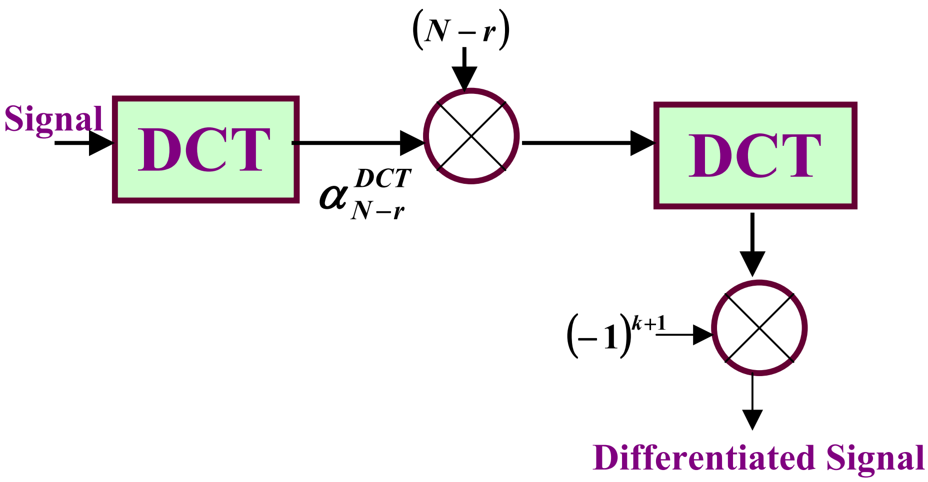
$$(\Delta x, \Delta y, t) = \underset{(\Delta x, \Delta y)}{\mathbf{arg\,min}} \iint_{(\xi,\eta)\in ARM} \left[\frac{\partial I(\xi,\eta,t)}{\partial \xi}\Delta x + \frac{\partial I(\xi,\eta,t)}{\partial \eta}\Delta y + \frac{\partial I(\xi,\eta,t)}{\partial t}\Delta t\right]^2 d\xi d\eta =$$

$$\underset{(\Delta x, \Delta y)}{\mathbf{arg\,min}} \iint_{(\xi,\eta)\in ARM} \left(\dot{\mathbf{I}}_{\xi,\eta,t} \bullet \mathbf{\Delta XYT}\right)^2 d\xi d\eta ,$$

where $\mathbf{\Delta XYT}$ is a vector of space-time shifts $(\Delta x; \Delta y; \Delta t)$ and $(.\bullet.)$ is a scalar (inner) vector product, $\dot{\mathbf{I}}_{\xi,\eta,t}$ is a vector of image intensity function space-time derivatives:

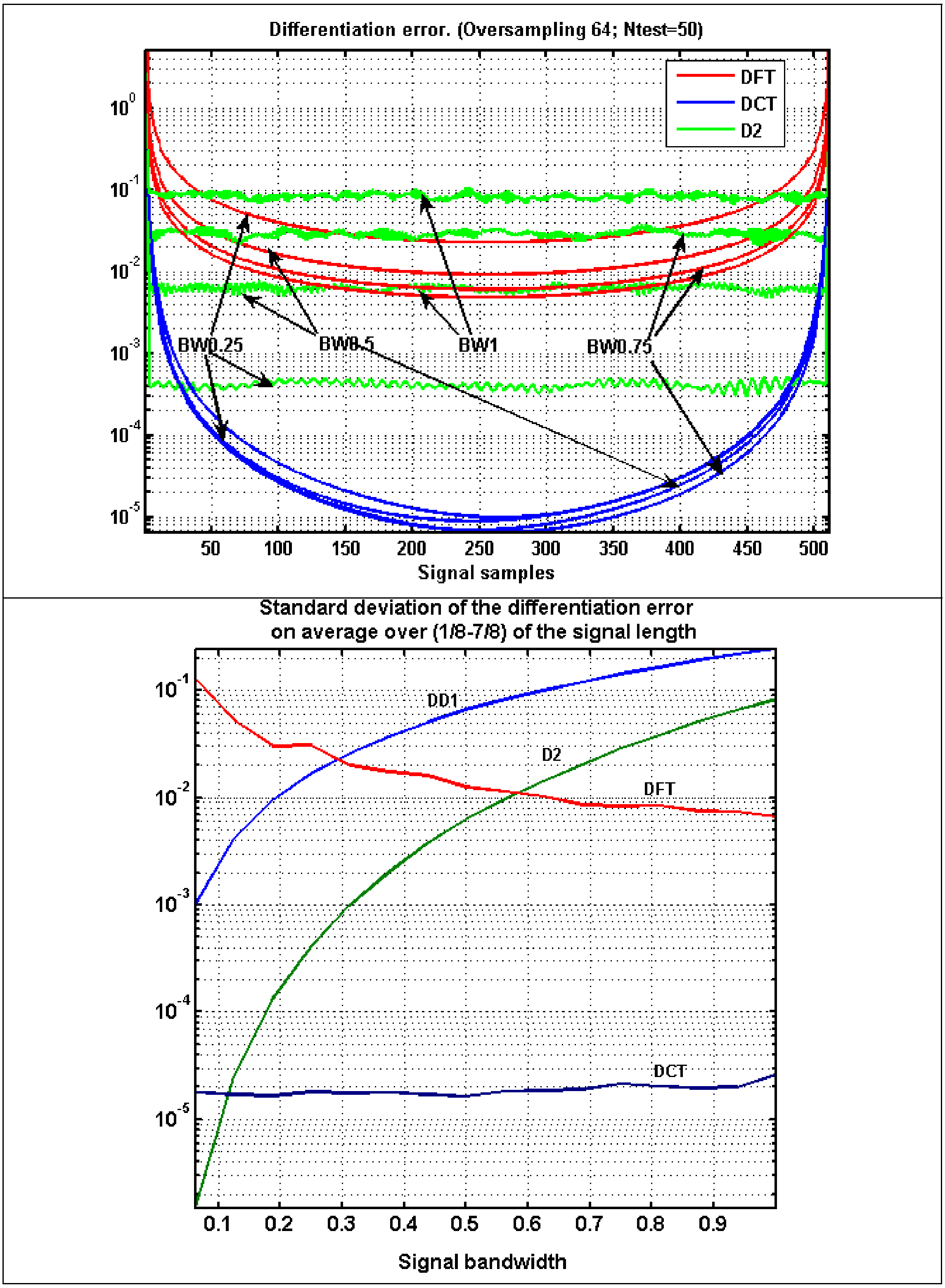
$$\dot{\mathbf{I}}_{\xi,\eta,t} = \left(\frac{\partial}{\partial \xi} I(\xi,\eta,t);\ \frac{\partial}{\partial \eta} I(\xi,\eta,t);\ \frac{\partial}{\partial t} I(\xi,\eta,t)\right) .$$





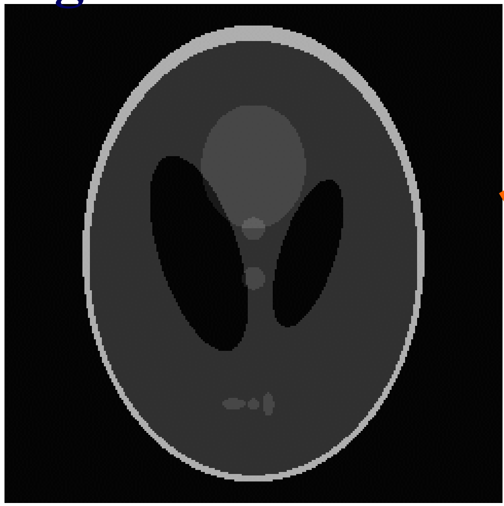
DCT-based differentiation algorithm

Comparison of the accuracy of methods of numerical differentiation

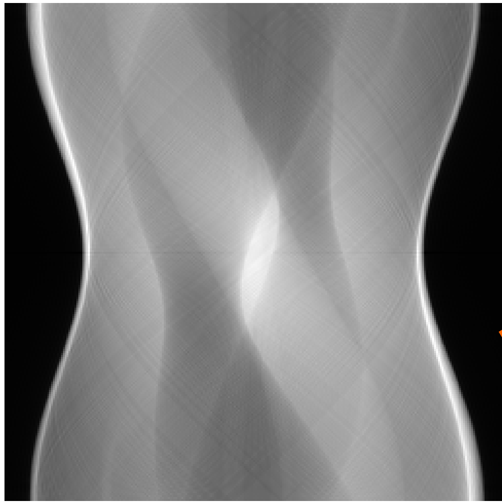


[ERR1,ERR2,ERR_d1,ERR_d2]=differentiation_test(N,M,Ntest);

”Ramp”-filtering in filtered back projection algorithm for image reconstruction from projections



Radon transform: rotation and directional summation



Tomographic reconstruction: ramp-filtering projections, back projecting, rotation and summation

Ramp-filtering of projections

$$\left\{\hat{a}_{k,l}\right\} = \sum_{s=0}^{N_\theta - 1} \mathbf{ROT}_\theta \left\{\left\{\mathbf{IDFT}\left(H_r^{diff} \cdot \mathbf{DFT}\left(\left\{p_k^{(\theta_s)}\right\}\right)\right)\right\} \otimes \left\{\vec{1}_l\right\}\right\}$$

where $\left\{p_k^{(\theta_s)}\right\}$ are sampled image projections obtained for angle θ_s , $\left\{\vec{1}_l\right\}$ is a vector-column of ones , $\otimes$ symbolizes matrix Kronecker product and ROTè is image rotation operator through angle θ

radon_invradon_demo;