

L. Yaroslavsky

**Numerical Integration
and Differentiation
of Sampled Data**

Problem formulation

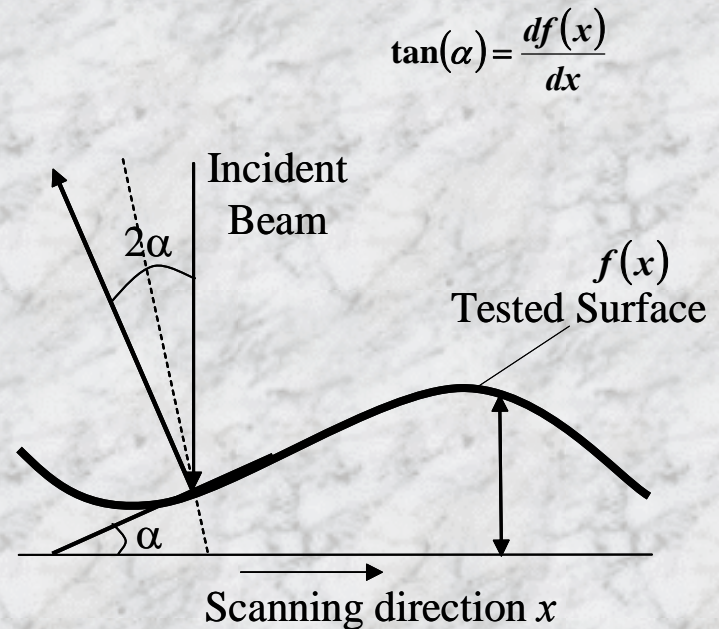
While carrying out numerical computation with sampled data one should realize that sampled data represent, with certain accuracy, data that are originally continuous functions, and the numerical algorithms approximate certain continuous transformation of those functions. Consequently, results of the computation should be evaluated with respect to that continuous transformation. In other words, given computational algorithm applied to sampled data, one should find out to what continuous transformation of continuous functions that are represented by the sampled data this algorithm corresponds.

In the presentation, we

- address this problem for the case of numerical integration and differentiation of sampled data
- compare, from these point of view, different known methods for numerical integration and differentiation
- describe methods that can be regarded as “gold standards” for numerical integration and differentiation of sampled data
- Suggest efficient computational algorithms for their implementation

Numerical integration for profilometry and laser deflectometry: determination of surface profile from its slope

Numerical integration has numerous applications in several optical fields such as the wave-front reconstruction from wave-front slope measurements. Another application of the numerical integration is the optical surfaces determination by laser deflectometry. This technique of measuring profile of surfaces 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. As these data are available in form of slope samples taken at finite number of sampling points, numerical integration is required.



Traditional methods of numerical integration:

Newton-Cotes quadratures

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

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

In these integration methods, a linear, quadratic, cubic and higher order interpolations, respectively, are assumed between the sampled slope data.

Cubic spline interpolation:

In the cubic spline interpolation, a cubic polynomial is evaluated between every couple of points [and then, an analytical integration of these polynomials is performed.

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

where the coefficients $\{m_k\}$ are determined by the system of linear equations:

$$\{(m_{k-1} + 4m_k + m_{k+1}) = 6(a_{k+1} - 2a_k + a_{k-1})\}$$

Given set of sampled data, how can one compare different methods of numerical integration with the ideal (exact) integrator and characterize their accuracy ?

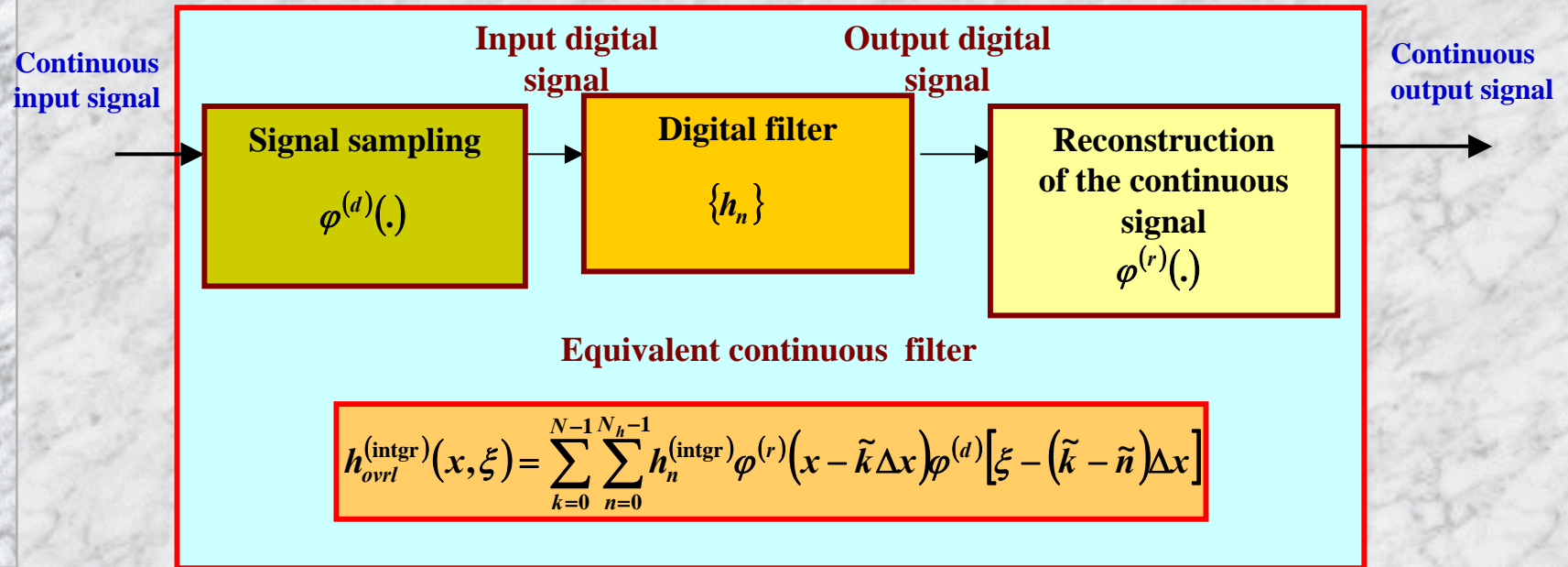
Integration of functions can be regarded as a convolution of the functions with a corresponding integration kernel, or point spread function. Different numerical integration algorithms correspond to different approximations of the ideal integration point spread function. In particular, they will differ in terms of their resolving power, that is, of their capability to resolve between two close sharp peaks in the functions.

Thanks to the convolution theorem for Fourier Transform, convolution integral can be treated in Fourier transform domain as a product of Fourier spectrum of the function and of that of the convolution kernel called convolution frequency transfer function, or Frequency Response. On account of that, one can also characterize numerical integration algorithms in terms of the accuracy of approximating the ideal integration Frequency Response.

Overall PSF of the digital integrator

In digital processing, the integrating filtering is implemented in signal domain as

$$\{\bar{a}_k\} = \left\{ \sum_n h_n^{(\text{intgr})} a_{k-n} \right\}$$



Continuous frequency response of the digital filter

Definition: Overall frequency response of the digital filter

$$H_{eq}(f, p) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} h_{ovrl}(x, \xi) \exp[i2\pi(fx - p\xi)] dx d\xi =$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \sum_{k=0}^{N_b-1} \sum_{n=0}^{N_h-1} h_n \varphi^{(r)}(x - \tilde{k}\Delta x) \varphi^{(d)}[\xi - (\tilde{k} - \tilde{n})\Delta x] \exp[i2\pi(fx - p\xi)] dx d\xi =$$

$$\left[\sum_{n=0}^{N_h-1} h_n \exp(i2\pi p \tilde{n} \Delta x) \right] \int_{-\infty}^{\infty} \varphi^{(r)}(x) \exp(i2\pi f x) dx \int_{-\infty}^{\infty} \varphi^{(d)}(\xi) \exp(-i2\pi p \xi) d\xi \left[\sum_{k=0}^{N-1} \exp[i2\pi(f - p)\tilde{k}\Delta x] \right]$$

$$H_{eq}(f, p) = CFR(p) \cdot \Phi^{(r)}(f) \cdot \Phi^{(d)}(-p) \cdot SV(f, p)$$

where

$$CFR(p) = \sum_{n=0}^{N_h-1} h_n \exp(i2\pi p \tilde{n} \Delta x) = \sum_{n=0}^{N_h-1} h_n \exp[i2\pi p(n + u^d)\Delta x]$$

$$\Phi^{(r)}(f) = \int_{-\infty}^{\infty} \varphi^{(r)}(x) \exp(i2\pi f x) dx$$

$$\Phi^{(d)}(-p) = \int_{-\infty}^{\infty} \varphi^{(d)}(x) \exp(-i2\pi p x) dx$$

$$SV(f, p) \propto \frac{\sin[\pi(f - p)N\Delta x]}{\sin[\pi(f - p)\Delta x]} = N \text{sincd}[N, \pi(f - p)N\Delta x]$$

Definition: Continuous frequency response of the digital filter

Frequency response of the signal reconstruction device

Frequency response of the signal sampling device

Term responsible for filter space variance

Continuous frequency response of the digital filter

Represent $\{h_n\}$ through their SDFT($u,0$) coefficients $\{\eta_r\}$:

$$h_n = \sum_{r=0}^{N_h-1} \eta_r \exp\left(-i2\pi \frac{n+u}{N_h} r\right) = \frac{1}{\sqrt{N_h}} \sum_{r=0}^{N_h-1} \left[\eta_r \exp\left(-i2\pi \frac{ur}{N_h}\right) \right] \exp\left(-i2\pi \frac{nr}{N_h}\right)$$

Then obtain: $CFR(f) \propto \sum_{r=0}^{N_h-1} \eta_r \exp\left[i2\pi \left(f\Delta x u^{(d)} - \frac{ur}{N_h}\right)\right] \sum_{n=0}^{N_h-1} \exp\left[i2\pi \left(f\Delta x - \frac{r}{N_h}\right)n\right] =$

$$\sum_{r=0}^{N_h-1} \eta_r \frac{\sin\left[\pi \left(fN_h\Delta x - r\right)\right]}{\sin\left[\pi \frac{\left(fN_h\Delta x - r\right)}{N_h}\right]} \exp\left[i\pi \left(\frac{N_h-1}{2} - u^{(d)}\right) f\Delta x\right] \exp\left[-i2\pi \left(u + \frac{N_h-1}{2}\right) \frac{r}{N_h}\right]$$

or, with selection: $u = -u^{(s)} = (N_h - 1)/2$

$$CFR(f) \propto \sum_{n=0}^{N_h-1} \eta_r \operatorname{sincd}\left[N_h; \left(f\Delta x - \frac{r}{N_h}\right)\right]$$

Continuous frequency response of the digital filter $\{h_n\}$

Theorem 1.

SDFT coefficients of the digital filter impulse response are samples of its Continuous Frequency Response

Theorem2.

Continuous frequency response of the digital filter is a discrete sinc-interpolated function of its samples

Theorem 3

Given signal sampling and reconstruction devices, overall frequency response of the digital filter is determined by its Continuous Frequency Response

Continuous frequency responses of 4 numerical integration methods:

Trapezoidal rule:

$$\bar{a}_k^{(Tr)} - \bar{a}_{k-1}^{(Tr)} = \frac{1}{2}(a_{k-1} + a_k)$$



$$\eta_r^{(Tr)} = \frac{\bar{\alpha}_r^{(Tr)}}{\alpha_r} = \begin{cases} 0, & r = 0, \\ -\frac{\cos(\pi r / N)}{2i \sin(\pi r / N)}, & r = 1, \dots, N-1 \end{cases}$$

Simpson rule:

$$\bar{a}_k^{(S)} - \bar{a}_{k-2}^{(S)} = \frac{1}{3}(a_{k-2} + 4a_{k-1} + a_k)$$



$$\eta_r^{(S)} = \frac{\bar{\alpha}_r^{(S)}}{\alpha_r} = \begin{cases} 0, & r = 0 \\ -\frac{\cos(2\pi r / N) + 2}{3i \sin(2\pi r / N)}, & r = 1, \dots, N-1 \end{cases}$$

3/8- Simpson rule:

$$\bar{a}_k^{(3/8S)} - \bar{a}_{k-3}^{(3/8S)} = \frac{3}{8}(a_{k-3} + 3a_{k-2} + 3a_{k-1} + a_k)$$



$$\eta_r^{(3/8S)} = \begin{cases} 0, & r = 0 \\ -\frac{\cos\left(\frac{3\pi r}{N}\right) + 3\cos\left(\frac{\pi r}{N}\right)}{i \sin\left(\frac{3\pi r}{N}\right)}, & r = 1, \dots, N-1 \end{cases}$$

Cubic spline interpolation:

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

$$\{(m_{k-1} + 4m_k + m_{k+1}) = 6(a_{k+1} - 2a_k + a_{k-1})\}$$



$$\eta_r^{(CS)} = \begin{cases} 0, & r = 0 \\ -\frac{1}{4i} \frac{\cos\left(\frac{\pi r}{N}\right)}{\sin\left(\frac{\pi r}{N}\right)} \left[1 + \frac{3}{\cos\left(2\pi \frac{r}{N}\right) + 2} \right], & r = 1, \dots, N-1 \end{cases}$$

A gold standard for numerical integration: DFT-based method

The ideal continuous signal integrator:

$$\bar{a}(x) = \int a(x) dx = \int_{-\infty}^{\infty} \frac{i}{2\pi f} \alpha(f) \exp(-i2\pi f x) dx = \int_{-\infty}^{\infty} H_{\text{int}}(f) \alpha(f) \exp(-i2\pi f x) df$$

where $a(x)$ and $\alpha(f)$ are signal and its Fourier transform spectrum, respectively

The integrating filter frequency response $H_{\text{int}}(f) = i/2\pi f$

According to Theorems 1-2, the ideal numerical integrator is determined by samples of its Continuous Frequency Response

For even N :

$$\eta_r^{(\text{int})} = \begin{cases} 0, & r = 0 \\ -\frac{N}{i2\pi r}, & r = 1, 2, \dots, N/2 - 1 \\ -\frac{1}{2\pi}, & r = N/2 \\ \eta_{N-r}^*, & r = N/2 + 1, \dots, N - 1 \end{cases}$$

For odd N :

$$\eta_r^{(\text{int})} = \begin{cases} 0, & r = 0 \\ -\frac{N}{i2\pi r}, & r = 1, 2, \dots, (N-1)/2 \\ \eta_{N-r}^*, & r = N/2 + 1, \dots, N - 1 \end{cases}$$

One can, using FFT algorithm, implement the integration in DFT domain:

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

DFT-based numerical integration method

Signal integration is an operation with infinitesimal signal increments. Therefore numerical signal integration assumes interpolation of sampled data. Integration can also be treated as signal convolution. The DFT based digital signal integration automatically implies signal discrete sinc-interpolation

DCT-based integrator

DFT-based integration (FI) method is the closest approximation to the continuous integrator. Yet this method suffers from boundary effects since it implements cyclic convolution rather than shift-invariant convolution.

Boundary effects exhibit themselves in form of oscillations around signal discontinuities that may occur between samples at the beginning and the end of the available signal realization. One can substantially decrease influence of boundary effects by means of signal extension to double length with its “mirror reflected” copy:

$$\tilde{a}_k = \begin{cases} a_k, & k = 0, 1, \dots, N-1 \\ a_{2N-1-k}, & k = N, \dots, 2N-1 \end{cases}$$

For such a signal of double length, integrator frequency response is defined as

$$\eta_r^{(\text{int})} = \begin{cases} 0, & r = 0 \\ -\frac{N}{i\pi r}, & r = 1, 2, \dots, N-1 \\ -\frac{1}{\pi}, & r = N \\ \eta_{N-r}^*, & r = N+1, \dots, 2N-1 \end{cases}$$

The doubling of the number of signal samples in this implementation of the DFT based method does not double the computational complexity of computation. $2N$ -DFT convolution of signals obtained by mirror reflection extension can be carried out using fast algorithms of Discrete Cosine Transform and of associated with it Discrete Cosine/Sine Transform for signals of N samples.

Computation of convolution of signals with “mirror reflection” extension

DFT of signals with “even” symmetry $\tilde{a}_k = \begin{cases} a_k, & k = 0, 1, \dots, N-1 \\ a_{2N-1-k}, & k = N, \dots, 2N-1 \end{cases}$ is as follows:

$$\tilde{\alpha}_r = \frac{1}{\sqrt{2N}} \sum_{k=0}^{2N-1} \tilde{a}_k \exp\left(i2\pi \frac{kr}{2N}\right) = \left\{ \frac{2}{\sqrt{2N}} \sum_{k=0}^{N-1} a_k \cos\left[\pi \frac{(k+1/2)r}{N}\right] \right\} \exp\left(-i\pi \frac{r}{2N}\right) = \alpha_r^{(DCT)} \exp\left(-i\pi \frac{r}{2N}\right)$$

where $\alpha_r^{(DCT)} = DCT\{a_k\} = \frac{2}{\sqrt{2N}} \sum_{k=0}^{N-1} a_k \cos\left(\pi \frac{k+1/2}{N} r\right)$ is Discrete Cosine Transform

Therefore, the DFT spectrum of signal extended by “mirror reflection” can be computed via DCT using Fast DCT algorithm.

For computing convolution, the signal spectrum should be multiplied by the filter frequency response and then the inverse DFT should be computed for the first N samples:

$$b_k = \frac{1}{\sqrt{2N}} \sum_{r=0}^{2N-1} \alpha_r^{(DCT)} \exp\left(-i\pi \frac{r}{2N}\right) \eta_r \exp\left(-i2\pi \frac{kr}{2N}\right)$$



Computation of convolution of signals with “mirror reflection” extension (Ctnd)

The coefficients are the DFT of samples of the filter PSF that are real numbers. Therefore they feature the complex conjugate symmetry property: $\{\eta_r = \eta_{2N-r}^*\}$. From properties of DCT it follows that DCT spectra feature the following symmetry property: $\alpha_N^{(DCT)} = 0$, $\alpha_u^{(DCT)} = \alpha_{2N-u}^{(DCT)}$. Then one can obtain:

$$b_k = \frac{1}{\sqrt{2N}} \left\{ \alpha_0^{(DCT)} \eta_0 + \sum_{r=1}^{N-1} \alpha_r^{(DCT)} \eta_r \exp\left(-i2\pi \frac{k+1/2}{2N} r\right) + \sum_{r=1}^{N-1} \alpha_{2N-r}^{(DCT)} \eta_{2N-r} \exp\left[-i2\pi \frac{k+1/2}{2N} (2N-r)\right] \right\} =$$

$$\frac{1}{\sqrt{2N}} \left\{ \alpha_0^{(DCT)} \eta_0 + \sum_{r=1}^{N-1} \alpha_r^{(DCT)} \left[\eta_r \exp\left(-i2\pi \frac{k+1/2}{2N} r\right) + \eta_r^* \exp\left(i2\pi \frac{k+1/2}{2N} r\right) \right] \right\}$$

from which obtain the final formula for digital convolution in DCT domain:

$$b_k = \frac{1}{\sqrt{2N}} \left\{ \alpha_0^{(DCT)} \eta_0 + \sum_{r=1}^{N-1} \alpha_r^{(DCT)} \eta_r^{re} \cos\left(\pi \frac{k+1/2}{N} r\right) - \sum_{r=1}^{N-1} \alpha_r^{(DCT)} \eta_r^{im} \sin\left(\pi \frac{k+1/2}{N} r\right) \right\}$$

First two terms of this expression constitute inverse DCT of the product $\{\alpha_r^{(DCT)} \eta_r^{re}\}$ while the third term is Discrete Cosine/Sine Transform (DcST) of the product $\{\alpha_r^{(DCT)} \eta_r^{im}\}$. Both transforms can be computed using fast algorithms

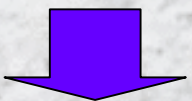
DCT-based integration algorithm

Frequency response of the iterator for signals of $2N$ samples:

$$\eta_r^{(int)} = \begin{cases} 0, & r = 0 \\ -\frac{N}{i\pi r}, & r = 1, 2, \dots, N-1 \\ -\frac{1}{\pi}, & r = N \\ \eta_{N-r}^*, & r = N+1, \dots, 2N-1 \end{cases}$$

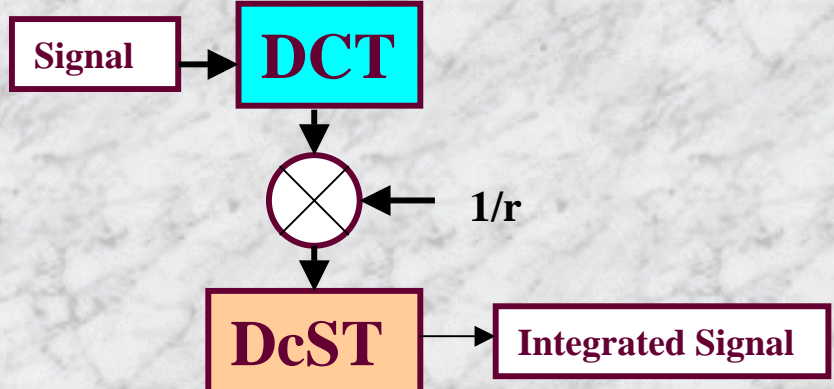
Convolution of signals extended to double length by "mirror reflection":

$$b_k = \frac{1}{\sqrt{2N}} \left\{ \alpha_0^{(DCT)} \eta_0 + \sum_{r=1}^{N-1} \alpha_r^{(DCT)} \eta_r^{re} \cos\left(\pi \frac{k+1/2}{N} r\right) - \sum_{r=1}^{N-1} \alpha_r^{(DCT)} \eta_r^{im} \sin\left(\pi \frac{k+1/2}{N} r\right) \right\}$$



INTEGRATION ALGORITHM

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



Experimental verification: Integration of periodic sinusoidal signals

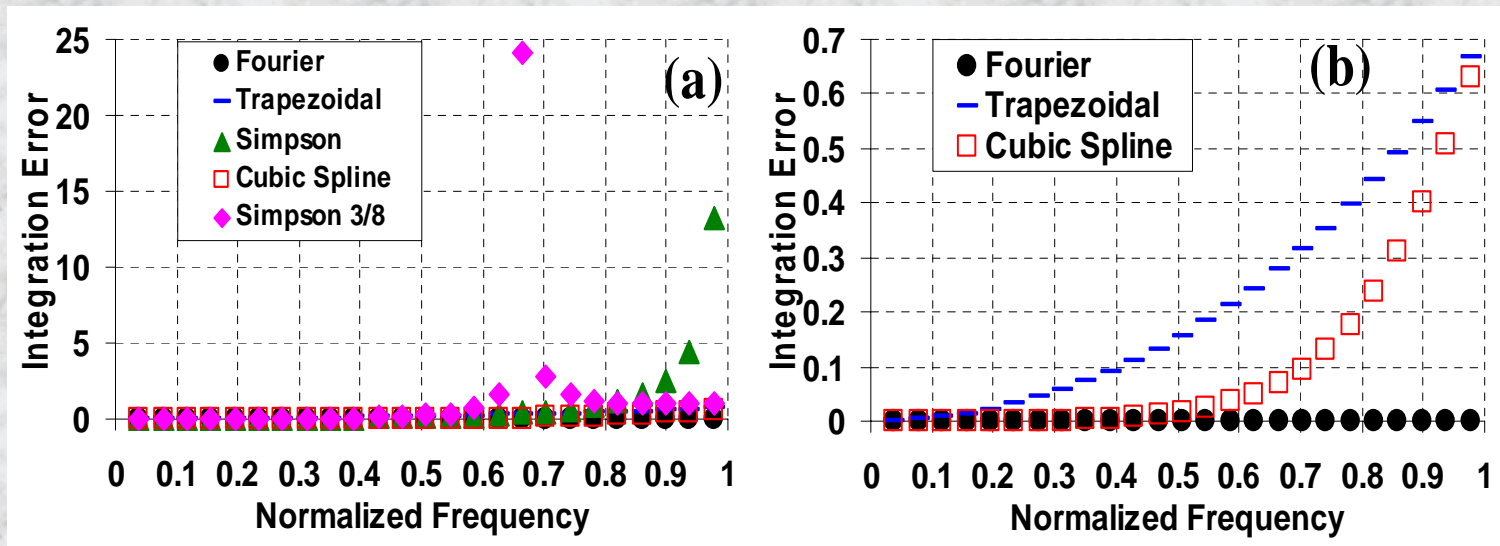
Test signals:

$$a_0(x) = \cos\left(2\pi \frac{P}{N} x + \text{randomphase}\right)$$

$$a'_0(x) = -2\pi \frac{P}{N} \sin\left(2\pi \frac{P}{N} x + \text{randomphase}\right)$$

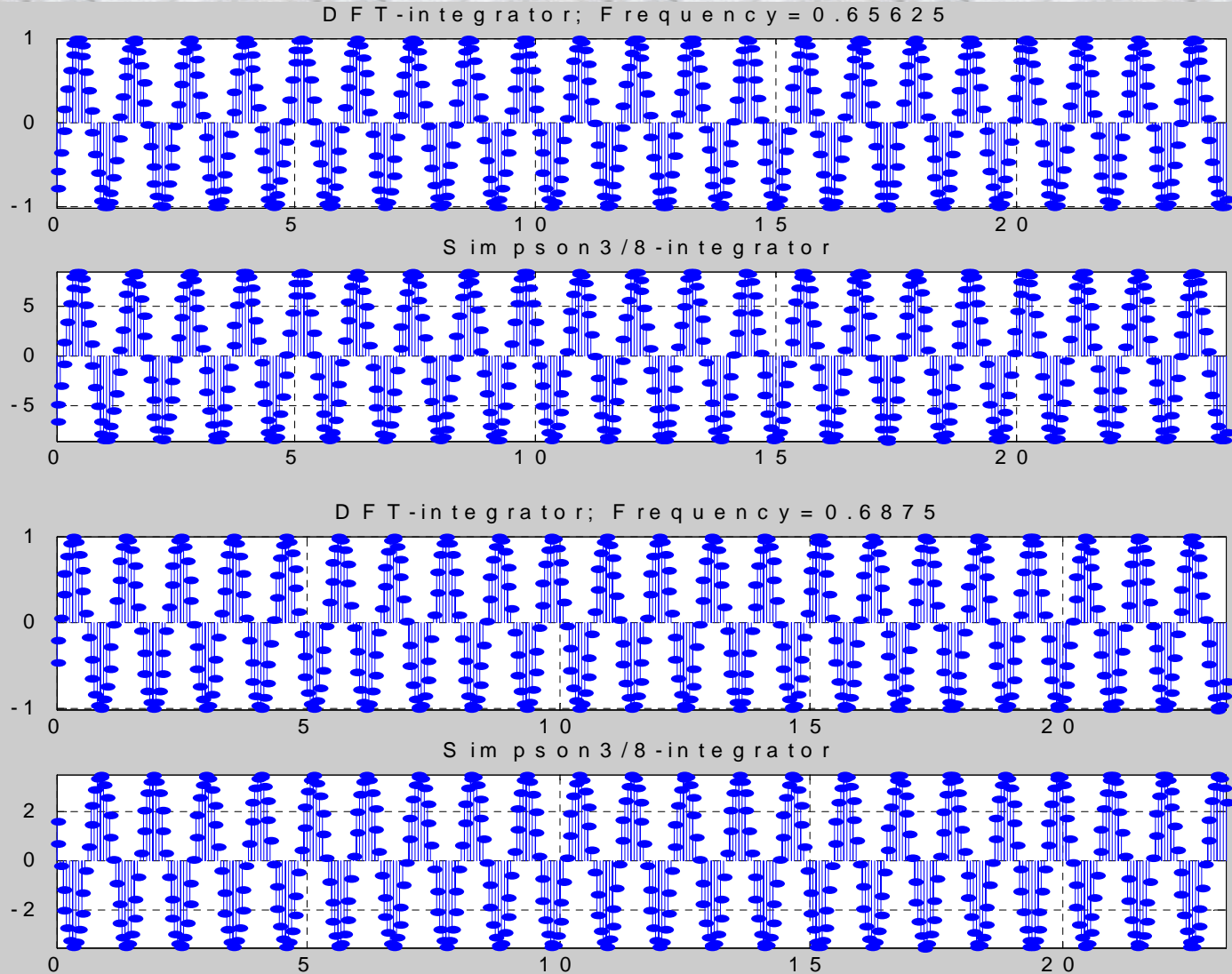
Error measure:

$$\text{error} = \sqrt{\frac{1}{N} \sum_{k=0}^N [a_i(k) - a_0(k)]^2}$$



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

Experimental verification: Phase inversion phenomenon for 3/8-Simpson method



Experimental verification: Aperiodic signals and boundary effects

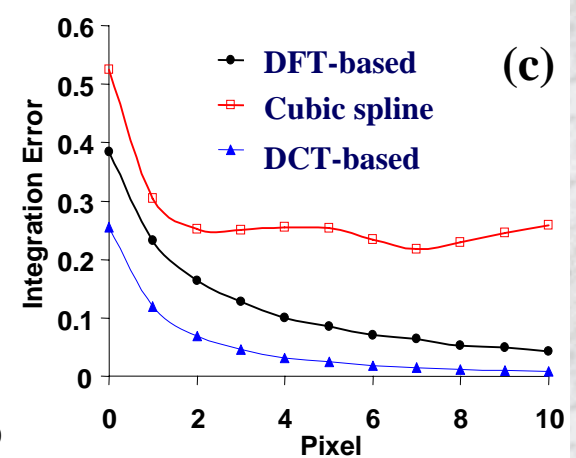
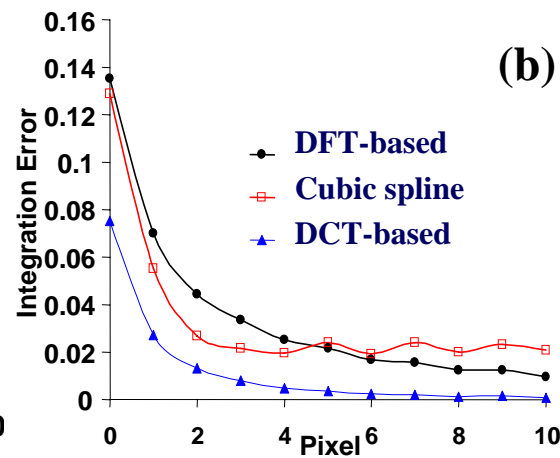
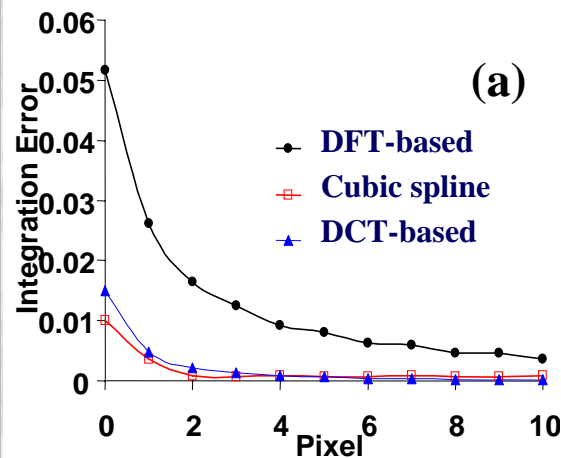
Test signals:

$$a_0(x) = \cos\left(2\pi \frac{p_{\text{int}} + s/n}{N} x + \text{randomphase}\right)$$

$$a_0'(x) = -2\pi \frac{p}{N} \sin\left(2\pi \frac{p_{\text{int}} + s/n}{N} x + \text{randomphase}\right)$$

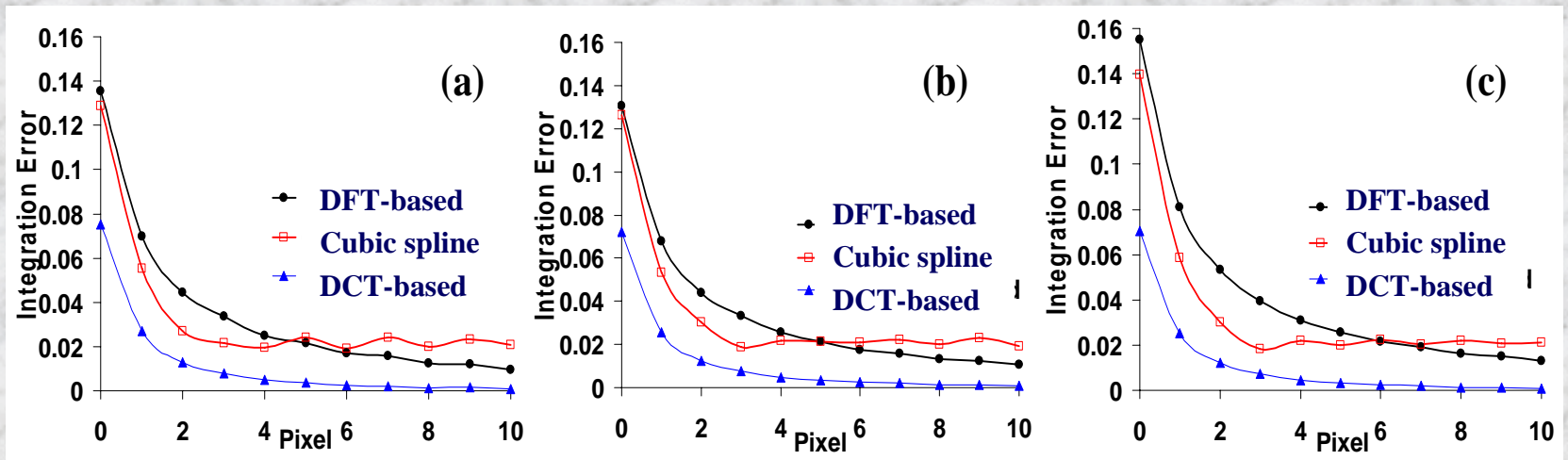
Sample-wise error measure:

$$\text{error}_{p_{\text{int}}}(k) = \sqrt{\frac{1}{n} \sum_{s=0}^{n-1} [a_i^{p_{\text{int}}+s/n}(k) - a_0^{p_{\text{int}}+s/n}(k)]^2}$$



Experimentally obtained integration error versus sample k for DFT-based (FI) method (black) CSI method (red) and DCT-based (Extended) method (blue). Normalized initial frequency: (a) $\nu = 0.273$. (b) $\nu = 0.547$ and (c) $\nu = 0.820$

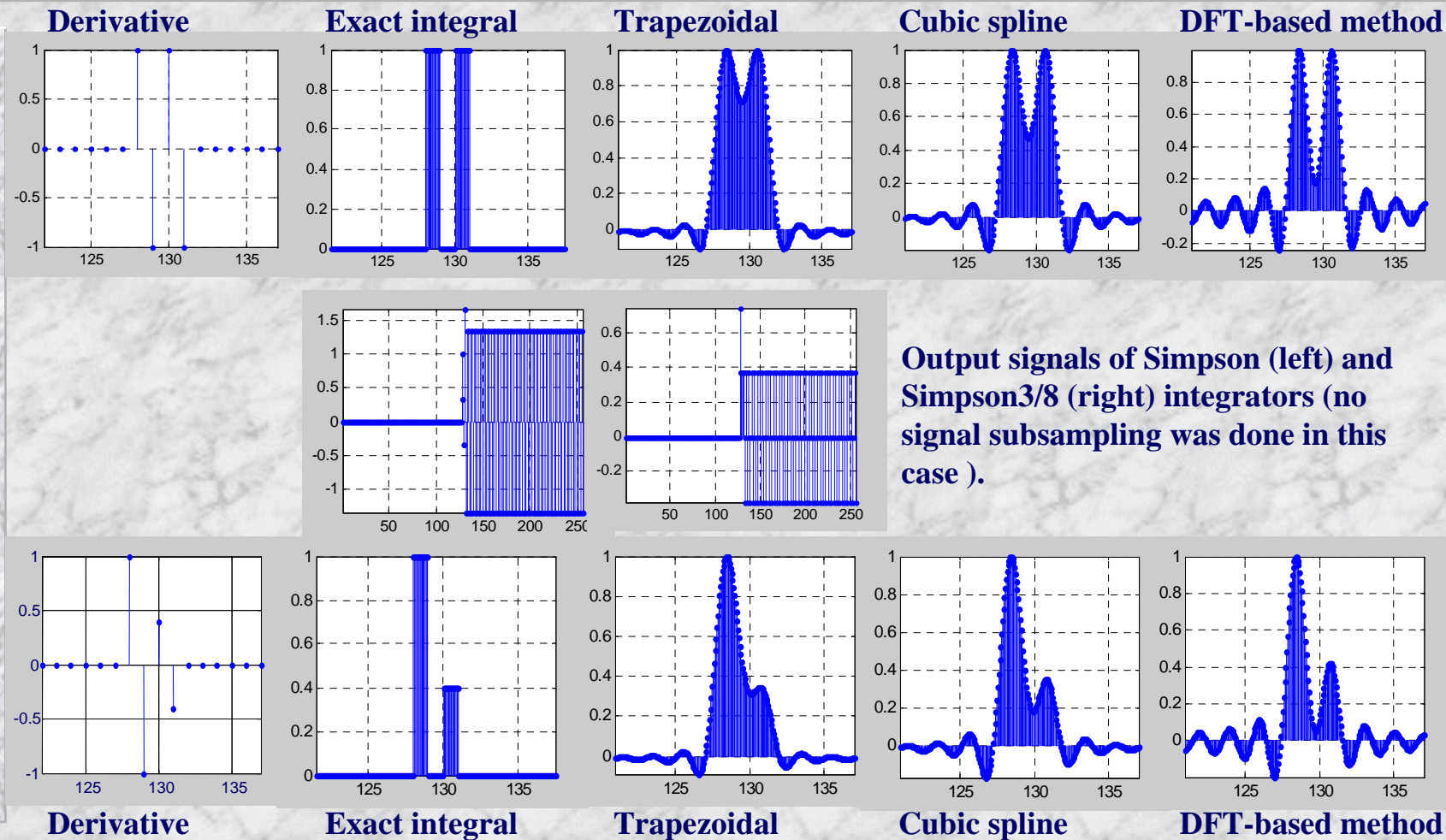
Experimental verification: Aperiodic signals and boundary effects (Ctnd)



Average error evaluated in the 10 first samples of the domain for initial normalized frequency ($p = 0.547$) and different N : (a) $N = 256$, (b) $N = 512$, (c) $N = 1024$.

Resolving power of integrators

Resolving power of integrators characterizes their capability to resolve between close sharp impulses in the integrated data



Conclusion

- **The DFT based method provides the best numerical approximation to the ideal continuous integrator and outperforms other integrators in terms of the resolving power, although it is vulnerable to boundary effects due to the fact that it implements signal cyclic convolution rather than regular shift invariant convolution.**
- **Conventional higher order numerical integration methods (Simpson, Simpson 3/8 and alike) produce substantial artifacts for signal that contain high frequency components and have poor resolving power**
- **The DCT-based method, being even slightly better than the DFT-based method in terms of the approximating the ideal continuous integrator, is also substantially more robust to boundary effects.**
- **Boundary effect errors for DFT and DCT-based methods do not propagate more than to about 10 first and last signal samples, or about 10 sampling intervals.**

Numerical differentiation of sampled data

Similarly to signal integration, signal differentiation is an operation with infinitesimal signal increments. Therefore numerical signal differentiation assumes interpolation of sampled data as well.

Differentiation can also be treated as signal convolution. In signal and Fourier transform domains, it is described as

$$\dot{a}(x) = \frac{d}{dx} a(x) = \int_{-\infty}^{\infty} [(-i2\pi f)\alpha(f)] \exp(-i2\pi f x) df = \int_{-\infty}^{\infty} [H_{diff}(f)\alpha(f)] \exp(-i2\pi f x) df$$

where $a(x)$ and $\alpha(f)$ are signal and its Fourier transform spectrum, respectively, and $H_{diff} = -i2\pi f$ is the differentiating filter frequency response.

Similarly to integration, numerical differentiating can be implemented either in signal domain as

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

or in the domain of Discrete Fourier Transform (DFT) as

$$\{\dot{a}_k\} = \text{IDFT}\{\eta_r^{(diff)} \bullet \text{DFT}\{a_k\}\}$$

where $\{a_k\}$, $k=0, \dots, N-1$ is a set of samples of the input signal to be integrated, DFT and IDFT are operators of direct and inverse Discrete Fourier Transforms, and sign \bullet denotes element-wise product of vectors

Point spread functions and continuous frequency responses of numerical differentiation methods

Conventional signal domain differentiation methods:

$$(1) \quad h_n^{diff(1)} = [-1, 1]$$

$$\eta_r^{diff(1)} \propto \sin(\pi r / N)$$

$$(2) \quad h_n^{diff(2)} = [-1/12, 8/12, 0, -8/12, 1/12]$$

$$\eta_r^{diff(2)} \propto \frac{8\sin(2\pi r / N) - \sin(4\pi r / N)}{12}$$

DFT/FFT-based differentiation method ("Ramp"-fiter):

$$\{\dot{a}_k\} = \text{IDFT}\left(\{\eta_r^{diff}\} \cdot \text{DFT}(\{a_k\})\right)$$

Even N:

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

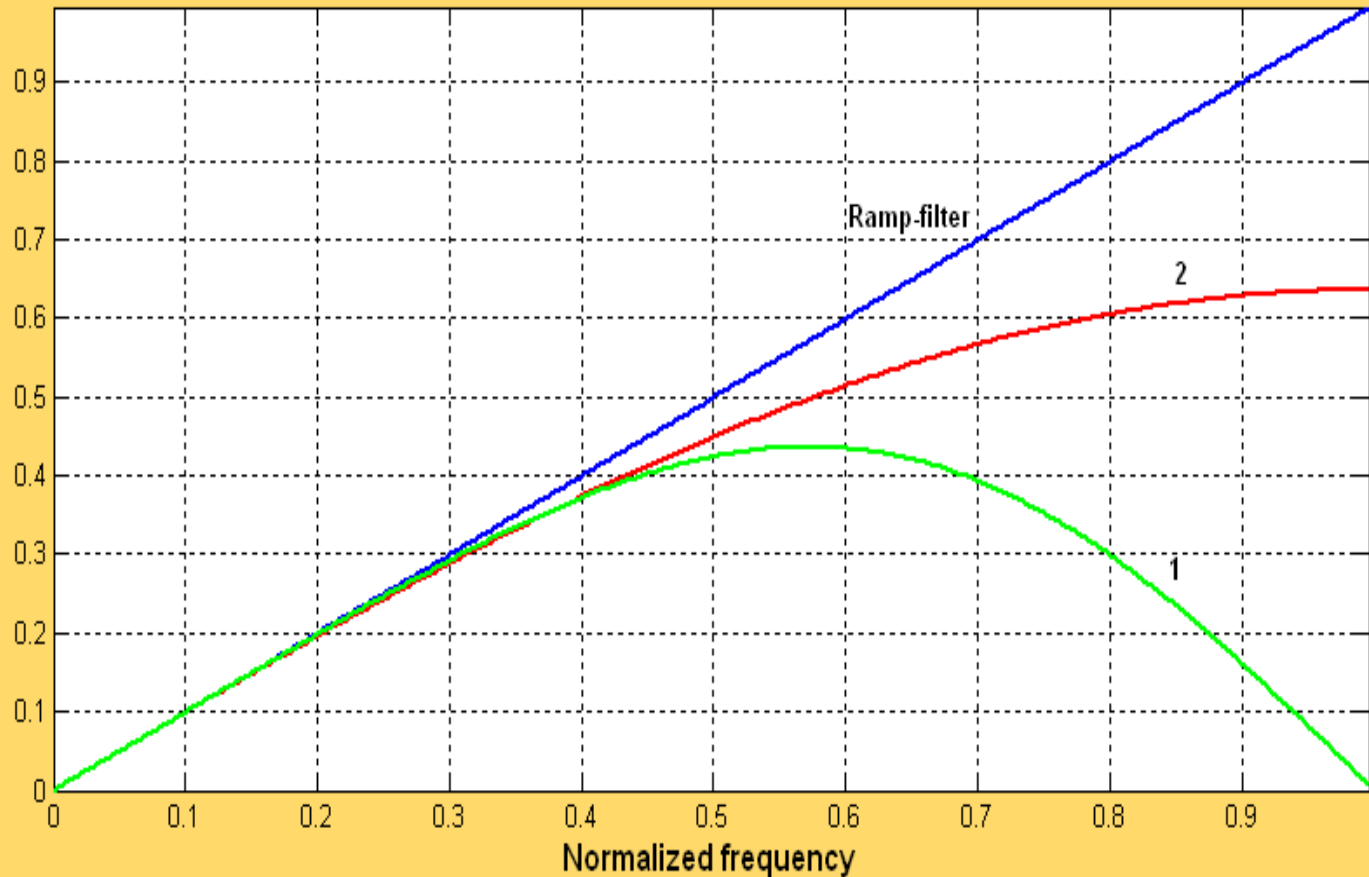
Odd N:

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

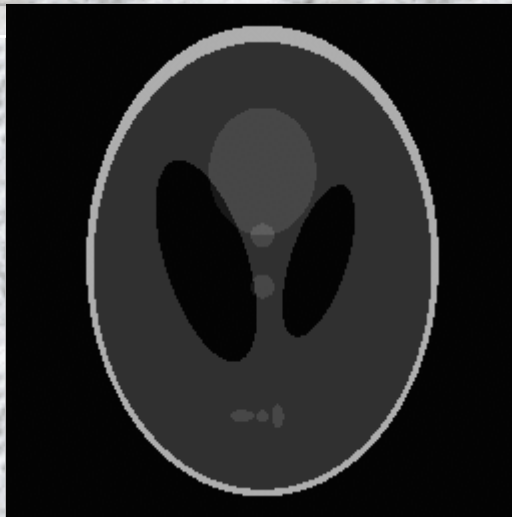
According to Theorem 1-2, DFT-based method is a gold standard as its continuous frequency response, in the signal base-band, is a discrete sinc-interpolated function of samples of the ideal continuous integrator

Frequency responses of numerical differentiating filters: comparison

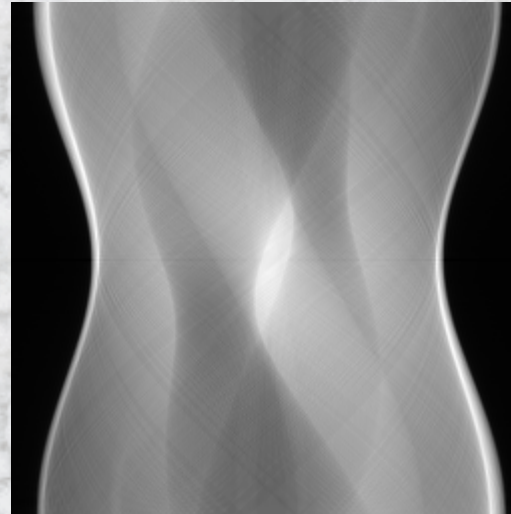
Frequency responses of 3 types of numerical differentiation



”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

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

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

References

1. **L. Yaroslavsky, Digital Holography and Digital Image Processing, Kluwer Academic Publishers, Boston, 2005**
2. **<http://mathworld.wolfram.com>**
3. **Joseph L. Zachary Introduction to Scientific Programming Computational Problem Solving Using: Maple and C, Mathematica and C**
4. **J. H. Mathews, K. D. Fink, Numerical methods using Matlab, Prentice Hall, 1999**
5. **W. H. Press, B. P. Flannery, S. A. Teukovsky, W. T. Wetterling, Numerical recipes. The art of scientific computing, Cambridge University Press, Cambridge, 1987**
6. **L. Yaroslavsky, Digital Holography and Digital Image Processing, Kluwer Academic Publishers, Boston, 2004**
7. **L. Yaroslavsky, Alfonso Moreno, Juan Campos, Frequency responses and resolving power of numerical integration of sampled data, Optics Express, v. 13, No. 8, 18 Apr. 2005, p. 2892-2905**

Leonid Yaroslavsky

DIGITAL HOLOGRAPHY AND DIGITAL IMAGE PROCESSING

Principles, Methods, Algorithms



Kluwer Academic Publishers
Optics

DIGITAL HOLOGRAPHY AND DIGITAL IMAGE PROCESSING Principles, Methods, Algorithms

Digital holography and digital image processing are twins born by computer era. They share origin, theoretical base, methods and algorithms. The book describes these common fundamental principles, methods and algorithms including image and hologram digitization, data compression, digital transforms and efficient computational algorithms, statistical and Monte-Carlo methods, image restoration and enhancement, image reconstruction in tomography and digital holography, discrete signal resampling and image geometrical transformations, accurate measurements and reliable target localization in images, recording and reconstruction of computer generated holograms, adaptive and nonlinear filters for sensor signal perfecting and image restoration and enhancement.

DIGITAL HOLOGRAPHY AND DIGITAL IMAGE PROCESSING Principles, Methods, Algorithms combines theory, heavily illustrated practical methods and efficient computational algorithms, and is written for senior-level undergraduate and graduate students, researchers and engineers in optics, photonics, opto-electronics and electronic engineering.

ISBN 1-4020-7634-7



9 781402 076343

Kluwer Academic Publishers
1-4020-7634-7

אוניברסיטת תל-אביב



L. Yaroslavsky,
Ph.D., Dr. Sc. Phys&Math,
Professor
Dept. of Interdisciplinary Studies,
Faculty of Engineering, Tel Aviv
University, Tel Aviv, Israel
www.eng.tau.ac.il/~yaro

