Selected Filtration Methods of the Standard ISO 16610

Dorothee Hüser, 5 Precision Engineering, PTB

April 13, 2016

In ISO TC 213 / WG 15 a set of filtration methods and concepts have been standardized or will be standardized soon to detrend surface topography with regard to form, waviness, and roughness and with regard to recognize features on different scales referred to as

- detrending and
- multi scale analysis

respectively. The standards provide a variety of numerical methods for band pass filtration.

1. Averaging with a Gaussian kernel
2. Smoothing with cubic splines
3. Smoothing with regression polynomials (Savitzky-Golay)
4. Combined method employing Gaussian kernel together with regression polynomials
5. Combined method employing interpolating polynomials and regression polynomials

1 Filter for Detrending

Regarding an object at different scales, its form, its waviness, and finally its roughness determine the geometry of the body respectively of its surface.

In roughness metrology, usually data have already been filtered with regard to high frequencies. They are cut off to make statistical roughness parameters comparable with compatible band widths of instrumental data, e.g. $\lambda < \lambda_\mathrm{s} = 2.5\ \mu$m. This low pass is carried out before any other high pass filtration, since the instruments often deliver the data being filtered that way. Applying both filters means applying a band pass filter.

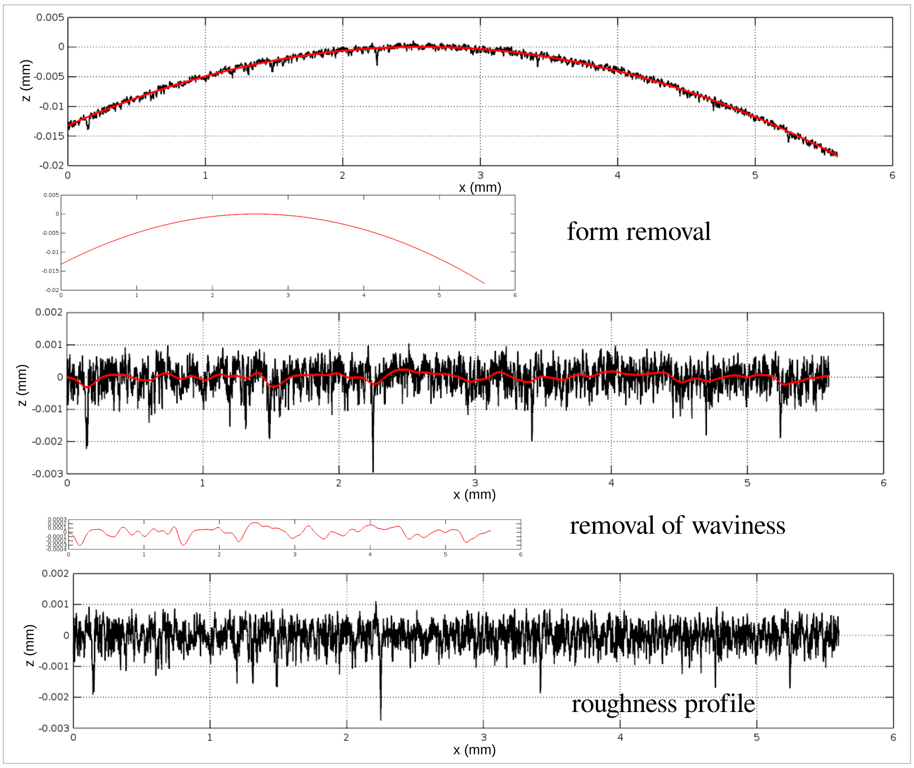


Figure 1: Idea of detrending.

Pattern recognition is essentially the same filtration process, however, using a narrow band. The convolution with a kernel is mathematically aequivalent to evaluating the cross correlation with a template. Be $s(x)$ the kernel, i.e. the template with zero mean and $z(x)$ the signal to be analysed the integration

$$w(x) = \int s(\xi)\, z(x - \xi)\, \mathrm{d}\xi \tag{1}$$

is called convolution and

$$w(x) = \int s(\xi)\; (z(x - \xi) - \bar{z})\; \mathrm{d}\xi \tag{2}$$

is called covariance and if it is normalized, it is called correlation, with $\bar{z}$ being the mean of the signal. Convolution is weighted averaging. It is a low pass.

Calculating weighted averages for each signal value, i.e. its convolution, means to perform as many integration processes as data points of the signal exist. For narrow kernels with only very few positions this usually is the preferred procedure. However, for kernels covering a large number of signal positions, it is more efficient to Fourier transform both the signal and the convolution kernel. The Fourier transform of the kernel is called transfer function $H(\lambda)$ and it then is multiplied to the Fourier transform of the signal. The multiplication is element wise. An inverse Fourier transformation delivers the smoothed signal (eg. waviness).

Fourier transformation presumes that the signal is periodically continued. Fig. 3 therefore gives the signal once more to show the continuation on one side. The same applies for the kernel. Both are

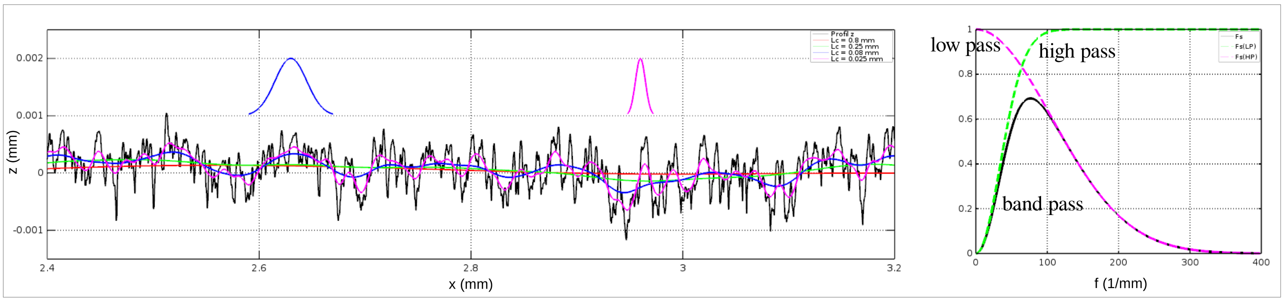


Figure 2: Convolution is weighted averaging. It is a low pass.

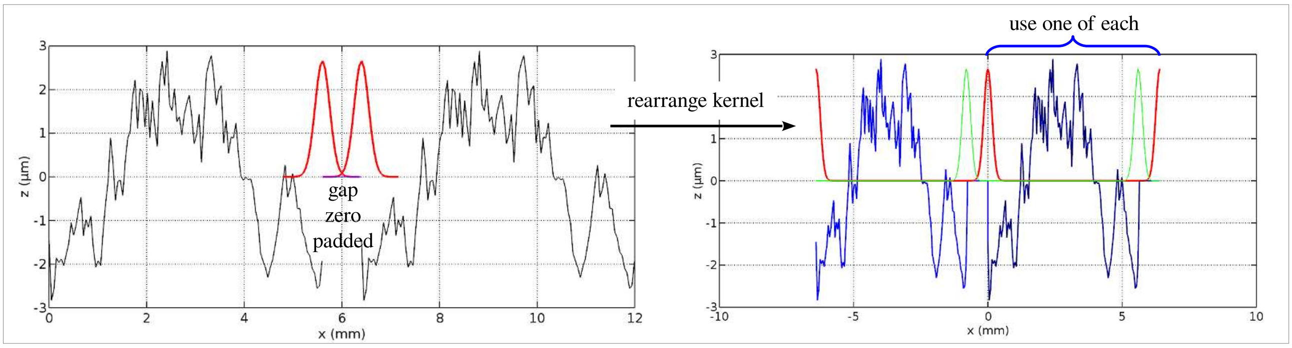


Figure 3: Prepare signal and kernel for appropriate usage in Fourier space.

displayed doubled to show the parts where the signal and kernel are continued. For calculation, it is not doubled, just the gap is filled appropriately. The following routine in Matlab / Gnu-octave illustrates the usage for a band pass

```
function zf = gauss_bandpass_prf(z, dx, lambda_HP, lambda_LP)
% with lambda_HP < lambda_LP
z = z(:);
n = size(z, 1);
nl = floor(lambda_LP/dx) + 1;
nhalf = floor((n-1)/2);
x  = [0:nhalf+nl]'*dx;
nn = size(x,1);

x2  = [x;-x(nn:-1:2)];
n2  = size(x2,1);

% filter constant
alpha = sqrt(log(2)/pi);

% Gaussian kernels for low and high
```

```
%  pass respectively , Fourier transforms
%  deliver their transfer functions
s = exp(-pi*x2.^2/(alpha*lambda_LP)^2)/(alpha*lambda_LP);
H_LP = fft(s)*dx;
s = exp(-pi*x2.^2/(alpha*lambda_HP)^2)/(alpha*lambda_HP);
H_HP = 1 .- fft(s)*dx;

% transfer function for band pass
H = H_HP .* H_LP ;

% Fourier transform of signal
%  with zero padded gap of size of
%  larger kernel (kernel of high pass)
z2  = [z; zeros(n2-n, 1)];
Fz  = fft(z2);

% back tp spatial domain
z3 = real(ifft(Fz.*H));
zf = z3(1:n);
```

In case of this kind of kernel the evaluation of the kernels and their Fourier transformations

```
s = exp(-pi*x2.^2/(alpha*lambda_LP)^2)/(alpha*lambda_LP);
H_LP = fft(s)*dx;
```

do not need to be performed, since the analytical solution of the Fourier transform of a Gaussian exists.

The Gaussian kernel according to ISO 16610-21 is given by

$$s(\xi) = \frac{1}{\alpha\lambda_{\mathrm{c}}} e^{-\pi\left(\frac{\xi}{\alpha\lambda_{\mathrm{c}}}\right)^2} \tag{3}$$

with s normalized such that $\int_{\xi_1}^{\xi_2} s(\xi) d\xi = 1$ if $\xi_1 = -\infty$ und $\xi_2 = +\infty$, but sufficiently well being 1 for $\xi_1 = -\lambda_{\mathrm{c}}$ und $\xi_2 = +\lambda_{\mathrm{c}}$. Then the Fourier transform of s, i.e. the transfer function of the low pass for $\alpha = \sqrt{\frac{\ln(2)}{\pi}}$ is

$$H(\lambda) = 2^{-(\frac{\lambda_{\mathrm{c}}}{\lambda})^2} = e^{-\pi(\alpha\frac{\lambda_{\mathrm{c}}}{\lambda})^2} \tag{4}$$

such that it can be implemented as follows

```
f0    = lambda_LP/(n2*dx);
f     = [0:nhalf+nl]'*f0;
Fw    = 2.^(-f.*f);
H_LP = [Fw; Fw(nn:-1:2)];
```

instead.

Using the Spline filtration method as low pass according to ISO 16610-22 can either be obtained by replacing the Gaussian kernel s by following function

$$s(\xi) = \sqrt{2}\ \sin(\pi\,\sqrt{2}\,\frac{1}{\lambda_\mathrm{c}}\,|\xi|\, +\, \frac{\pi}{4})\ exp(-\pi\,\sqrt{2}\,\frac{1}{\lambda_\mathrm{c}}\,|\xi|) \tag{5}$$

with following implementation example

```
Fb = 1/lambda_LP;
s = sqrt(2) * sin(pi*sqrt(2)*Fb*abs(x)+pi/4) .* ...
  exp(-pi*sqrt(2)*abs(Fb*x));
H_LP = fft(s)*dx;
```

or by following equation:

$$\left(\mathbf{E}\, +\, \alpha^4\mathbf{Q}\right)\mathbf{z}^{\mathtt{W}} = \mathbf{z}^{\mathtt{P}} \qquad \text{with} \qquad \alpha = \frac{1}{2\sin(\pi\Delta x/\lambda_\mathrm{c})} \tag{6}$$

The relation between Eq. (5) and Eq. (6) cannot be derived straight forward. Terzopoulos at Massechusetts Institute of Technology and Unser at the Biomedical Imaging Group, Swiss Federal Institute of Technology Lausanne have investigated the relation between spline numerics and signal theory, hence the transfer function of different spline filters [1, 2].

This writeup is restricted to only give an idea of what is a spline, in particular a smoothing spline.

Splines are polynomials to approximate a signal $z(x)$ between its knots x_i. The condition to be fullfilled is its *smoothness* within each knot as in case of wooden beams fixed at distant positions. The wooden splines can be described as cubic functions with natural constraints. The tension has to be at minimum.

Within each interval between the fixes (the nails or knots) $[x_i,\, x_{i+1})$ the shape of the wood is described by a polynomial $p_i(x)$ of degree 3 with the constraint

$$p_i(x_i) = z(x_i) \tag{7}$$

and the constraints

$$p_i(x_{i+1}) = p_{i+1}(x_{i+1}) \qquad \frac{\partial}{\partial x}p_i(x)|_{x_{i+1}} = \frac{\partial}{\partial x}p_{i+1}(x)|_{x_{i+1}} \qquad \frac{\partial^2}{\partial x^2}p_i(x)|_{x_{i+1}} = \frac{\partial^2}{\partial x^2}p_{i+1}(x)|_{x_{i+1}} \tag{8}$$

with $p_{i+1}(x)$ being the polynomial of the successing interval $[x_{i+1},\, x_{i+2})$.

As Hermite spline, for instance, it may look like:

$$p(x) = \sum_{i=1}^{n-1}\, p_i(x) \qquad p_i(x) = \left\{ \begin{array}{cl} \sum_{\nu=0}^{3} c_{\nu,i}\, x^\nu & \text{if}\ \ x \in [x_i,\, x_{i+1}) \\ 0 & \text{else} \end{array} \right. \tag{9}$$

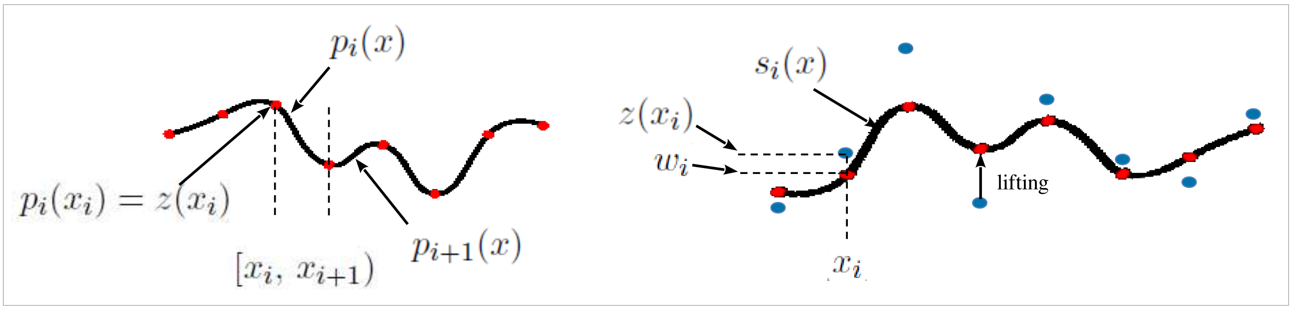


Figure 4: Principle of spline interpolation (*left*) and smoothing spline approximation (*right*).

Alternatively a representation of Splines by recursively employing Neville polynomials has been proposed by Carl R. de Boor [3] in 1970 being more stable than the straight forward representation in Eq. (9). Neville polynomials are described in *Numerical Recipes* [4], chapter 3.1, and in the Annex of this write up.

For data points $z(x_i)$ that scatter significantly the polynomials are oscillating. A smoothing by a sort of averaged passing the data points no longer fullfilling $p_i(x_i) = z(x_i)$ will be desired. That kind of splines is referred to as *smoothing splines* $s(x)$ that meet some points w_i of the smoothed profile, the so-called waviness, with

$$s(x) = \sum_{i=1}^{n-1} s_i(x) \qquad s_i(x) = \begin{cases} \sum_{\nu=0}^{3} c_{\nu,i}\, x^{\nu} & \text{if } \ x \in [x_i,\, x_{i+1}) \\ 0 & \text{else} \end{cases} \qquad s_i(x_i) = w_i \tag{10}$$

and

$$s_i(x_{i+1}) = s_{i+1}(x_{i+1}) \qquad \frac{\partial}{\partial x} s_i(x)|_{x_{i+1}} = \frac{\partial}{\partial x} s_{i+1}(x)|_{x_{i+1}} \qquad \frac{\partial^2}{\partial x^2} s_i(x)|_{x_{i+1}} = \frac{\partial^2}{\partial x^2} s_{i+1}(x)|_{x_{i+1}} \tag{11}$$

The smoothing is based on a maximum likelihood optimization that minimizes the sum of least squares of the differences between signal values and their partners being lifted for smoothing and of the tension (Tikhonov regularisation):

$$\mathcal{E} = \sum_{i=1}^{n-1} (z(x_i) - w_i)^2 + r \int_0^L \left(\frac{\partial^2}{\partial x^2} s(x) \right)^2 dx \to \min_{w_i} \tag{12}$$

with r called regularisation parameter ($r > 0$). For $r = 0$ the spline would be a straight line. The regularisation parameter determines the waviness, hence the wavelength of the spline filter. In ISO 16610-22, the polynomials are chosen such that after differentiation we obtain:

$$(\mathbf{E} + r\,\mathbf{Q})\,\mathbf{z}^{\mathtt{W}} = \mathbf{z}^{\mathtt{P}} \tag{13}$$

with $\mathbf{z}^{\mathtt{W}}$ being the column vector of the w_i and $\mathbf{z}^{\mathtt{P}}$ of the primary signal values $z(x_i)$, furthermore with $\mathbf{E}$ being the eye matrix and $\mathbf{Q}$ the pentagonal matrix as differential operator.

Transforming the equation to Fourier space or Laplace space, it is possible to define a cut off wavelength [1], i.e. to get the relation between a cut off wavelength and the regularisation parameter r. The representation given in ISO 16610-22 has been suggested by Krystek in 1997 [5] as $r = \alpha^4$ with

$$\alpha = \frac{1}{2\sin(\pi\Delta x/\lambda_{\mathrm{c}})} \tag{14}$$

with Δx being the (average of all) sampling interval sizes. The representation of Eq. (14) has to be taken care of, if the sampling intervals are too many orders of magnitude smaller than the cut off wavelength, i.e. if the ratio $\Delta x/\lambda_{\mathrm{c}}$ will be very small such that $1/\alpha^4$ get close to the resolution of the representation of the floating points!

Including tension terms with a factor called tension parameter β makes its transfer characteristics closer to that of the Gaussian filter:

$$\left(\mathbf{E} \;+\; \beta\alpha^2\mathbf{P} \;+\; (1-\beta)\alpha^4\mathbf{Q}\right)\mathbf{z}^{\mathtt{W}} = \mathbf{z}^{\mathtt{P}} \tag{15}$$

with $\mathbf{z}^{\mathtt{P}} := (z(x_1), \ldots, z(x_n))^T$ being the column vector of the original signal (height values of primary profile) and $\mathbf{z}^{\mathtt{W}} := (w_1, \ldots, w_n)^T$ the resultant smoothed signal (waviness). Furthermore $\mathbf{E}$ is the eye matrix, $\mathbf{P}$ the tridiagonal matrix of the differential quotiant for the first derivatives and $\mathbf{Q}$ the pentadiagonal matrix for the second derivatives.

The applications regarded by ISO 16610-22 are those for aequidistantly sampled data, but Eq (15) can as well be implemented if the band matrices are filled with the differing quotients.

Changing the parameter β, which may lie between 0 and 1, causes a change of the shape of the waviness, i.e. its curvature. Chosing $\beta = 0.625242$ delivers a transfer function whose shape is close to that of Eq. (4), which is the Fourier transform of the Gaussian kernel.

An extension of the profilometric spline of ISO 16610-22 for areas could be implemented as follows, but a document with ISO-Number 16610-62 has not yet been projected:

```
function W = iso16610_22_areal(Z, dlateral, lambda, tbeta)
  dx = dlateral(1);
  dy = dlateral(2);
  [n, m] = size(Z);
% n number of rows, hence y-axis
%   i.e. the number of knots in a column vector
% m number of columns, hence x-axis,
%   d.h. the number of knots in a row vector
  alphasquare = (0.5/sin(pi*dx/lambda))^2;
  p1 = tbeta*alphasquare;
  p2 = (1-tbeta)*alphasquare^2;
  A = zeros(n,n);
  for k=3:n-2
```

```
    A(k,k+2)= p2;
    A(k,k+1)=-4.0*p2-p1;
    A(k,k)  = 1.0+6.0*p2+2.0*p1;
    A(k,k-1)=-4.0*p2-p1;
    A(k,k-2)= p2;
 end
 A(1,1)    =  1.0+p2+p1; A(1,2)      = -2.0*p2-p1;
       A(1,3)      =  p2;
 A(2,1)    = -2.0*p2-p1; A(2,2)      = 1.0+5.0*p2+2.0*p1;
       A(2,3)      = -4.0*p2-p1; A(2,4)      = p2;
 A(n-1,n)  = -2.0*p2-p1; A(n-1,n-1)  = 1.0+5.0*p2+2.0*p1;
       A(n-1,n-2)  = -4.0*p2-p1; A(n-1,n-3)  = p2;
 A(n,n)    =  1.0+p2+p1; A(n,n-1)    = -2.0*p2-p1;
       A(n,n-2)    =  p2;
 Ainv = inv(A);
 if ( (n != m) || (dx != dy) )
    alphasquare = (0.5/sin(pi*dy/lambda))^2;
    p1 = tbeta*alphasquare;
    p2 = (1-tbeta)*alphasquare^2;
    B = zeros(m,m);
    for k = 3:n-2
      B(k+2,k)= p2;
      B(k+1,k)=-4.0*p2-p1;
      B(k,  k)= 1.0+6.0*p2+2.0*p1;
      B(k-1,k)=-4.0*p2-p1;
      B(k-2,k)= p2;
    end
    B(1,1)    =  1.0+p2+p1; B(2,1)      = -2.0*p2-p1;
         B(3,1)      =  p2;
    B(1,2)    = -2.0*p2-p1; B(2,2)      = 1.0+5.0*p2+2.0*p1;
         B(3,2)      = -4.0*p2-p1; B(4,2)        = p2;
    B(m,m-1)  = -2.0*p2-p1; B(m-1,m-1)  = 1.0+5.0*p2+2.0*p1;
         B(m-2,m-1)  = -4.0*p2-p1; B(m-3,m-1)    = p2;
    B(m,m)    =  1.0+p2+p1; B(m-1,m)    = -2.0*p2-p1;
         B(m-2,m)    =  p2;
    Binv = inv(B);
 else
    Binv = Ainv';
 end
 W = Ainv * Z * Binv;
```

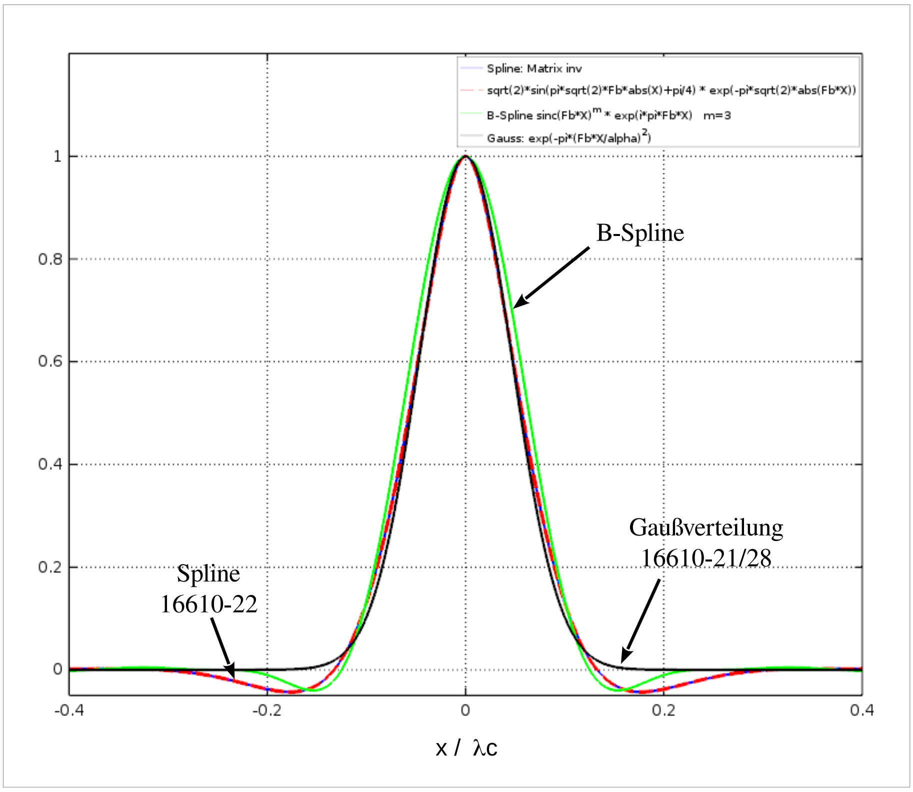


Figure 5: Different filter kernels having different shapes.

For larger matrices A and B a sparce matrix inversion procedure for band matrices is required.

2 Shape of Convolution Kernels

The various filters of ISO 16610 standards series allow different filter kernels having different shapes. Therefore, the choice has to be taken thoroughly depending on the application and its sensitivity to the curvature of the waviness.

If one of two waviness profiles of the same cut off wavelength has slightly higher amplitudes, which means a greater curvature, then the remaining residuals of the one waviness has less values in the tails of the amplitude density distribution than those of the other. The residuals, i.e. the differences between the values of the primary signal and its waviness, are the roughness. Hence if an identical primary profile is filtered with differently shaped kernels, the resultant waviness profiles differ as well, consequently the residuals, i.e. the roughness as is illustrated for one example in Figs. 5 and 6.

Savitzky-Golay smoothing filters use regression polynomials to obtain appropriate weights for a kernel to preserve statistical moments to a desired degree. Higher moments are preserved if the degree of the polynomial is higher retaining the shape of a signal. A detailed derivation of Savitzky-Golay filters is presented in Numerical Recipes [4], chapter 14.8, and in the Annex of this write up. For

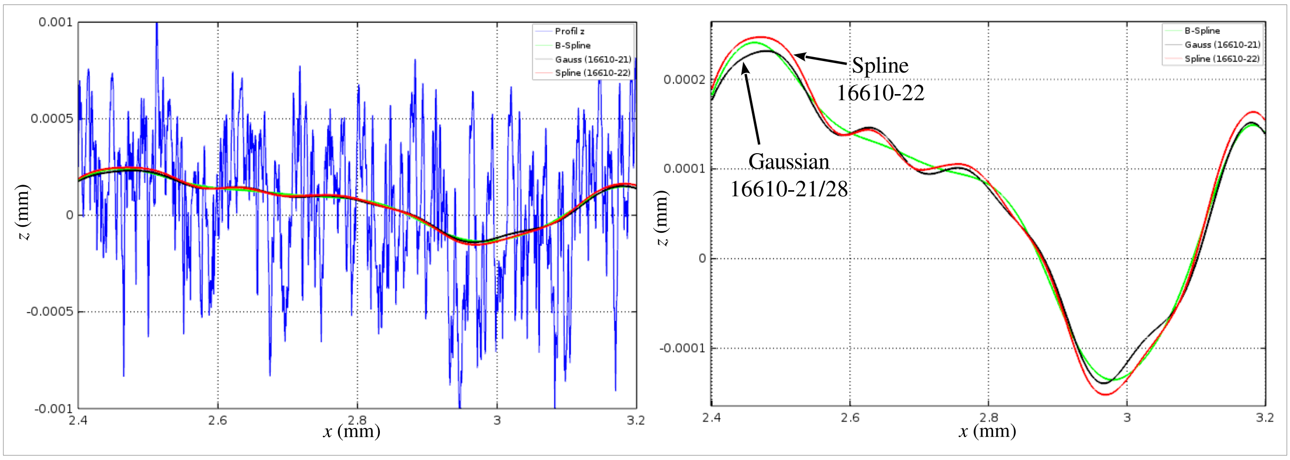


Figure 6: Different filter kernels having different shapes yield differently shaped waviness profiles: these plots shows the shapes of the waviness of one profile for the three different kernels displayed in Fig. 5.

uniform sampling intervals of size Δx the filter coefficients c_ν are only dependent on the kernel size and the degree of the polynomial, thus can be tabulated in general independent of specific data. The filter $w = f(z)$ is a linear filter with coefficients (i.e. with a discrete kernel):

$$w_k = \sum_{\nu=-n_L}^{n_R} c_\nu z_{k+\nu} \tag{16}$$

with n_L and n_R for the number of neighboring data points contributing to the weighted averaging. The polynomial of degree p be

$$w_k = \sum_{j=0}^{p} \beta_{k,j} (x - x_k)^j \tag{17}$$

Regard sampling positions x for signal values $z(x)$ at following aequidistantly given knots $x_k = k\Delta x$ with $k = 0, \ldots, N-1$.

To obtain the filter coefficients c_ν a Maximum-Likelihood estimation for the coefficients $\vec{\beta} = (\beta_0, \ldots \beta_p)^T$ of the polynomial is performed

$$\min_{\vec{\beta}} \{\mathcal{R}\} \qquad \text{with} \qquad \mathcal{R} = \sum_{\nu=-n_L}^{n_R} (z_{k+\nu} - w_{k+\nu}(\vec{\beta}))^2 \tag{18}$$

With writing down $w_{k+\nu}(\vec{\beta})$ we want to express that w is a function of the polynomial coefficients β.

Differentiation with respect to $\vec{\beta}_k$ to solve the optimization problem Eq. (18) means $\frac{\partial \mathcal{R}}{\partial \beta_{k,l}} = 0$ with

$l = 0, \dots, p$:

$$\mathbf{M}\vec{\beta} = \begin{pmatrix} \vdots \\ \sum_{\nu=-n_L}^{n_R} (\nu \Delta x)^l z_{k+\nu} \\ \vdots \end{pmatrix} \tag{19}$$

with

$$\mathbf{M} = \begin{pmatrix} & \vdots & \\ \dots & \sum_{\nu=-n_L}^{n_R} (\nu \Delta x)^l (\nu \Delta x)^j & \dots \\ & \vdots & \end{pmatrix} \tag{20}$$

Introducing a new symbol $\mathbf{G} = \mathbf{M}^{-1}$

$$\vec{\beta}_k = \mathbf{G} \begin{pmatrix} \vdots \\ \sum_{\nu=-n_L}^{n_R} (\nu \Delta x)^l z_{k+\nu} \\ \vdots \end{pmatrix} \tag{21}$$

with g_{jl} being the elements of the $(p+1) \times (p+1)$ matrix. i.e. $\mathbf{G} = (g_{jl})$:

$$w_k = \sum_{j=0}^{p} \beta_{k,j} (x - x_k)^j = \sum_{j=0}^{p} \sum_{j=0}^{p} g_{jl} \sum_{\nu=-n_L}^{n_R} (\nu \Delta x)^l z_{k+\nu} (x_k - x_k)^j \tag{22}$$

such that the desired filter coefficients are

$$c_\nu = \sum_{j=0}^{p} \sum_{l=0}^{p} g_{jl} (\nu \Delta x)^l 0^j \tag{23}$$

and with $0^j = 0$ for $j \neq 0$ and $0^0 = 1$ it is:

$$c_\nu = \sum_{l=0}^{p} g_{jl} (\nu \Delta x)^l \tag{24}$$

The Δx cancel such that universal filter coefficients

$$c_\nu = \sum_{l=0}^{p} g_{jl} (\nu)^l \tag{25}$$

can be obtained from matrix inversion.

$$\mathbf{M} \ = \ \mathbf{M}^{-1} = \begin{pmatrix} & \vdots & \\ \dots & \sum_{\nu=-n_L}^{n_R} \nu^l \nu^j & \dots \\ & \vdots & \end{pmatrix}^{-1} \ = \ \left(\mathbf{X}^{\mathrm{T}} \mathbf{X}\right)^{-1} \tag{26}$$

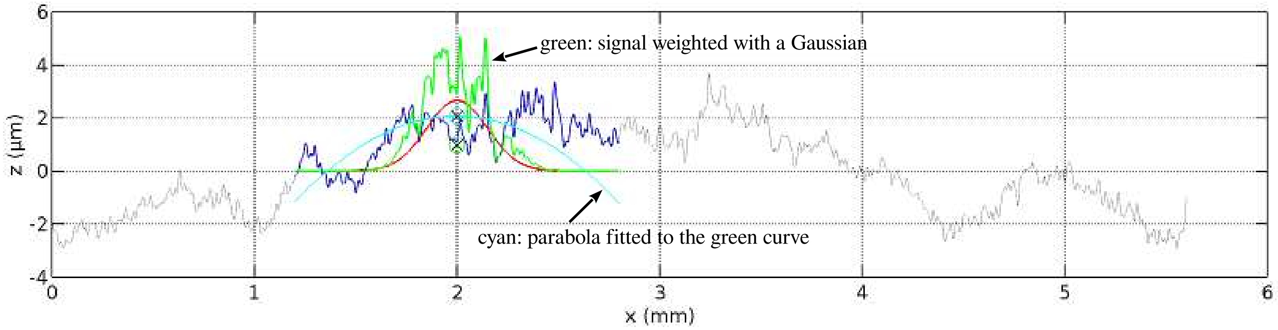


Figure 7: Smoothing with combined Gaussian and Savitzky-Golay filter for parabola, i.e. polynomial degree $p = 2$.

with

$$\mathbf{X} = \begin{pmatrix} 1 & -n_L & \dots & (-n_l)^p \\ 1 & -n_l+1 & \dots & (-n_L+1)^p \\ \vdots & \vdots & \dots & \vdots \\ 1 & n_R-1 & \dots & (n_R-1)^p \\ 1 & n_R & \dots & (n_R)^p \end{pmatrix} \tag{27}$$

ISO 16610-28 and 16610-31 employ a combination of the Savitzky-Golay filter coefficients and the Gaussian kernel according to Seewig [6, 7]

$$s_\nu = \frac{1}{a\lambda_c}\exp\left(-\pi\left(\frac{\nu\Delta x}{a\lambda_c}\right)^2\right) \qquad a = \begin{cases} \alpha = \sqrt{\frac{\ln(2)}{\pi}} & \text{if } p = 0, 1 \\ \gamma = \sqrt{\frac{-1-W(-1/(2e))}{\pi}} & \text{if } p = 2 \end{cases} \tag{28}$$

by

$$\min_{\vec{\beta}} \left\{ \sum_{\nu=-n_L}^{n_R} s_\nu \left(z_{k+\nu} - w_{k+\nu}(\vec{\beta})\right)^2 \right\} \tag{29}$$

yielding coefficients

$$c_\nu = s_\nu \sum_{l=0}^{p} g_{jl}\nu^l \tag{30}$$

with

$$\mathbf{G}^{-1} = \mathbf{M} = \begin{pmatrix} & \vdots & \\ \dots & \sum_{\nu=-n_L}^{n_R} s_\nu \nu^l \nu^j & \dots \\ & \vdots & \end{pmatrix} = \mathbf{X}^{\mathrm{T}}\mathbf{S}\,\mathbf{X} \qquad \text{and} \qquad \mathbf{S} = \begin{pmatrix} s_{-n_L} & 0 & \dots & \dots & 0 \\ 0 & s_{-n_L+1} & \dots & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & s_{n_R} \end{pmatrix} \tag{31}$$

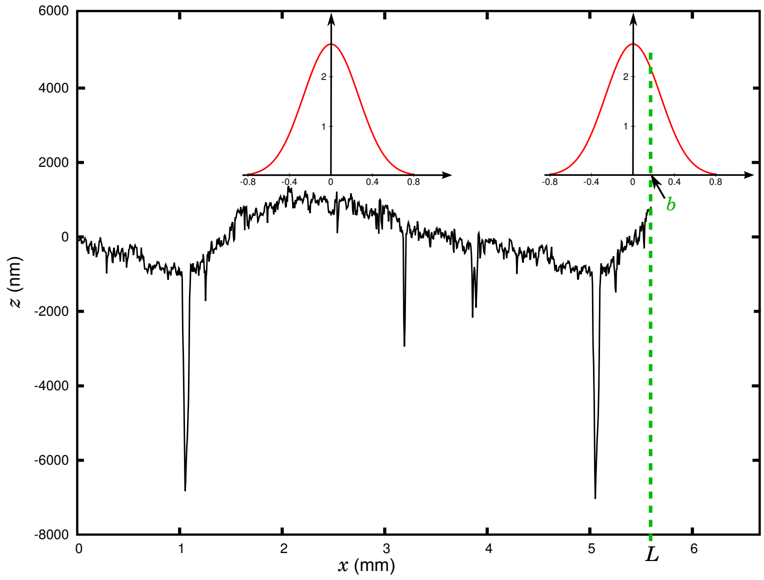


Figure 8: Convolution with a partial Gaussian covering the interval $\xi_2 = b < \lambda_\mathrm{c}$ mit $b = (L - x)$.

Retaining the zeros and first moment means to reduce end effects, see section 4.3 of ISO 16610-28. At the central part of the profile, the moments cancel for $n_L = n_R$, i.e. $\mathbf{M}$ becomes the eye matrix in the central part, while at the borders $\mathbf{X}^\mathrm{T}\mathbf{S}\,\mathbf{X}$ it is no longer the eye matrix, such that the resultant waviness and roughness profiles are identical to those obtained by the method of ISO 16610-21.

The polynomial of degree $p = 0$ denotes a constant, i.e. preserving the mean, which is given as Eq. (15) in section 4.3 of ISO 16610-28. It does the same as spatial convolution with weighting according to the kernel length giving rise to the effect at the borders of a profile. This is a weighted averaging

$$f(r_\mathrm{s}) = \frac{\int \rho(r) r dr}{\int \rho(r) dr} \tag{32}$$

The weight ρ in case of the Gaussian filter is the Gaussian kernel s:

$$w(x) = \frac{\int_{\xi_1}^{\xi_2} s(\xi)\, z(x+\xi) d\xi}{\int_{\xi_1}^{\xi_2} s(\xi) d\xi} \tag{33}$$

At the border regions $[0, \lambda_\mathrm{c}]$ und $[L - \lambda_\mathrm{c}, L]$ the kernel exceeds the profile, see Fig. 8 the partial Gaussian covers the interval $\xi_2 = b < \lambda_\mathrm{c}$ mit $b = (L - x)$.

Eq. (33) delivers

$$w(x) = \frac{\int_{-\lambda_\mathrm{c}}^{b} s(\xi)\, z(x+\xi) d\xi}{\int_{-\lambda_\mathrm{c}}^{b} s(\xi) d\xi} \tag{34}$$

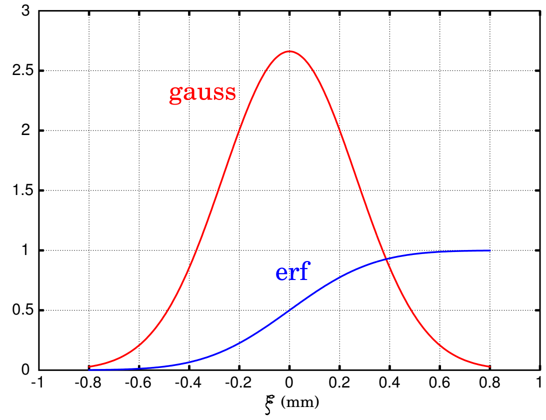


Figure 9: For weighting according to the kernel length the area of the Gaussian kernel, which is the *Gaussian error function* `erf`, is used.

At the left border we obtain $\xi_1 = -b$ and $b = x$:

$$w(x) = \frac{\int_{-b}^{\lambda_c} s(\xi) z(x+\xi) d\xi}{\int_{-b}^{\lambda_c} s(\xi) d\xi} \tag{35}$$

Due to the symmetry of the Gaussian

$$\int_{-b}^{\lambda_c} s(\xi) d\xi = \int_{-\lambda_c}^{b} s(\xi) d\xi \tag{36}$$

which approximates to the *error function* `erf` (Fig. 9) $\int_{-\lambda_c}^{\lambda_c} s(\xi) d\xi \approx 1$.

$$\frac{1}{\alpha\lambda_c} \int_{-\lambda_c}^{b} e^{-\pi\left(\frac{\xi}{\alpha\lambda_c}\right)^2} d\xi \approx \frac{1}{\alpha\lambda_c} \int_{-\infty}^{b} e^{-\pi\left(\frac{\xi}{\alpha\lambda_c}\right)^2} d\xi = \mathrm{erf}\left(\frac{b}{\alpha\lambda_c}\right) \tag{37}$$

Using $p = 1$ preserves the straight line of the averaged signal part, which means that it minimizes the end effect even better than $p = 0$, which is given as Eqn. (13) and (14) in section 4.3 of ISO 16610-28, as illustrated in Fig. 10.

That means that in the central part, $p = 0$ and $p = 1$ deliver identical results to the Gaussian of ISO 16610-21. Employing $p = 2$ preserves the variance, which means that it retains the widths of peaks and pores. The shape retaining effect of the $p = 2$ choice is illustrated in Fig. 7.

3 Nested Filters - Filter Chain

In order to perform detrending within much smaller band widths and to detect structures or features of various sizes, i.e. on different scales, topography signals are *sieved* like gravel stone to separate different grain sizes. The ISO filter toolbox gives two different types of filtration methods, to perfom

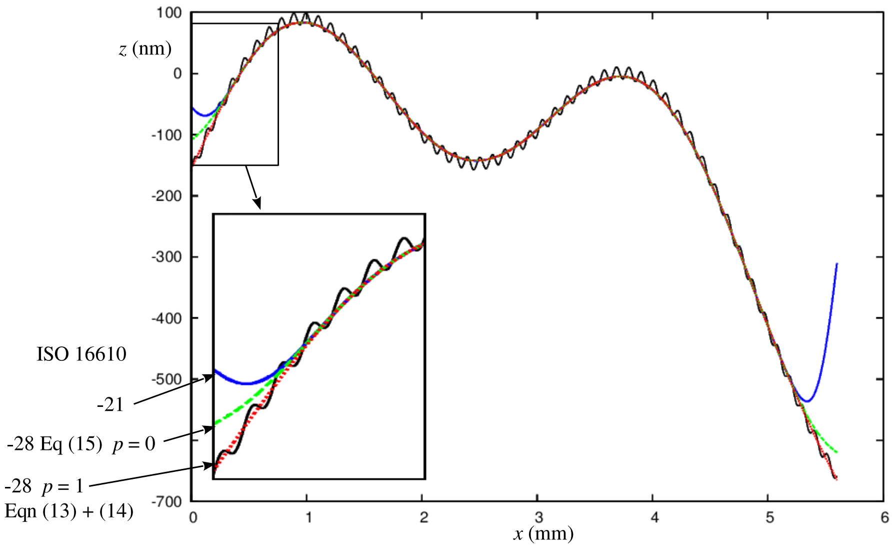


Figure 10: Minimizing end effects by Gaussian regressian. A regressian polynomial of order $p = 0$ is the mean and the regressian polynomial of $p = 1$ is a straight line.

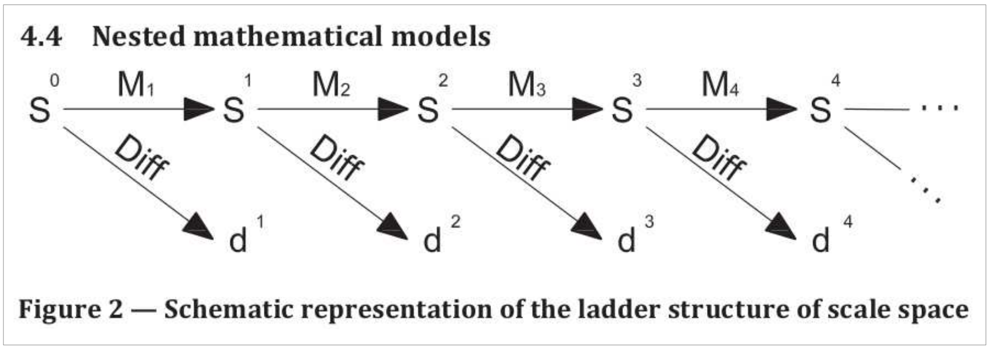


Figure 11: Copy of Fig. 2 of ISO 16610-49 showing the sieving concept for multi-scale analysis.

this

- a linear filter method using kernels similar to those described before (referred to as wavelets),
- and a morphological filter, which is non-linear.

The multi-scale analysis is realized as chain of filtration steps separating the low frequency part of a signal (referred to as smoothed s) from the high frequency signal part of the signal (referred to as residuals or differences d) as illustrated in Fig. 2 of ISO 16610-49 which we have copied to this writeup shown in Fig. 11. The idea of discrete wavelet transformations and multiresolution scaling is to split the data set half and half, the one half for the high passed and the other for the low passed component.

Unser has developed discrete wavelet transformation (DWT) algorithms to process medical data [8, 9, 10]. ISO 16610-29 for geometrical product specification, refers to splines wavelets. Spline wavelets are available as bisplines, biorthogonal wavelets to be found as BIOR, eg for Matlab or Gnu-octave or for python. They differ according to the number of vanishing moments N_R for the number of vanishing moments of the reconstruction / synthesis filter and N_D for the decomposition / analysis filter.

During the 1980s morphological filters have been developed as image processing tool for pattern recognition purposes for a variety of biomedical and automated industrial inspectation applications. They are nonlinear signal transformations that locally modify geometric features of the signal [11]. It was Matheron and Serra in the 1970s who have introduced the *mathematical morphology* into image processing [12]. A few years later they and others extended morphological filters to multilevel signals by shrinking/ expanding operations on binary images to sieve structures of various sizes accordingly.

The science of analysing surface texture and topographies operates with images that have a continuous height axis such that the mathematical morphology needs an appropriate modification. In the field of scanning probe microscopy the nonlinear operation of Minkowski subtraction often has been referred to as a deconvolution of a probe tip. Using the term deconvolution might be misleading for those who specify the term convolution as being a linear operation. Keller and Franke call a dilation result (being the consequence of probing with a finite tip size of a stylus insturment) *envelope* and the erosion result *envelope reconstruction* introducing an algorithm on a continuous height space [13].

For geometric product specification the non-linear morphological filtration procedures are proposed in the ISO standards series 16610-40ff. There are two different filters to remove structures

- for peaks on upper asperities called *opening*
- and for holes in valleys called *closing*.

They are realized by a successive application of the two elementary operations of morphology, namely *dilation* which is as well called Minkowski sum and *erosion*, the Minkowski difference:

- For *opening* first *erosion* is applied then *dilation*.
- For *closing* first *dilation* is applied then *erosion*.

Fig. 12 illustrates the two elementary operations of morphology *dilation* (Minkowski sum) and *erosion* (Minkowski difference). The signal on which the operations are applied is the black curve. The blue curve represents the border of a geometry which is called *structuring element* and which serves to remove geometry from data (like a cutting tool in surface finishing) resp. to add geometry onto data (like the envelope by probing with finited sized stylii). For the algorithm, the set of border points of the *structuring element* has one reference point or origin. For each of the signal coordinate pairs resp. triples (i.e. for each point) the reference point of the structuring element is placed on the regarded signal point. Then the borders are compared: for dilation the outward points are chosen, i.e. if the

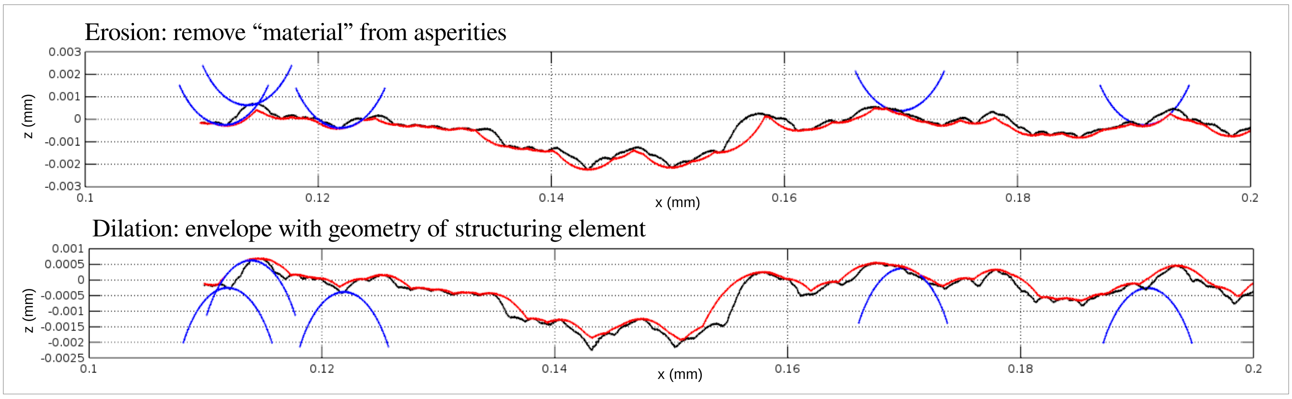


Figure 12: The two elementary operations of morphology are *dilation* (Minkowski sum) and *erosion* (Minkowski difference).

signal points lie outside with respect to the structuring element, then they are chosen, if the points of the border of the structuring element are outside, then they are chosen, see Fig. 13. For erosion the points that lie inside are chosen.

If the procedure is applied to any contour, the direction to define inside vs. outside is needed. For topography measurements, usually, there exists only one surface normal, which is referred to as height or z-axis. Villarubia delivers a C implementation [14] of Keller and Franke's envelope reconstruction method [13] for scan signals with arbitrarily shaped probe tips, i.e. structuring elements. The subsequent listing shows a simplified version of Villarubia's procedure, here assuming the structuring element having a circular shape with radius R:

```
function r = dilate_aequidistant(z, dx, R)
%
% z are heights, height axis is the direction
% to define outside and inside
%
% R radius of spherical structuring element
%
  z = z(:);
  nd = size(z,1);
  r = z;
  tipwidth = 0.8*R;
  nt = 2*floor(0.5*tipwidth/dx);
  x = [-nt:nt]' * dx;
  t = R .- sqrt(R^2 .- x.^2);
  for l=2:nt
    h = z(l) .- t(nt+2-l:2*nt+1);
    iabove = find(r(1:l+nt)<h);
    r(iabove) = h(iabove);
```

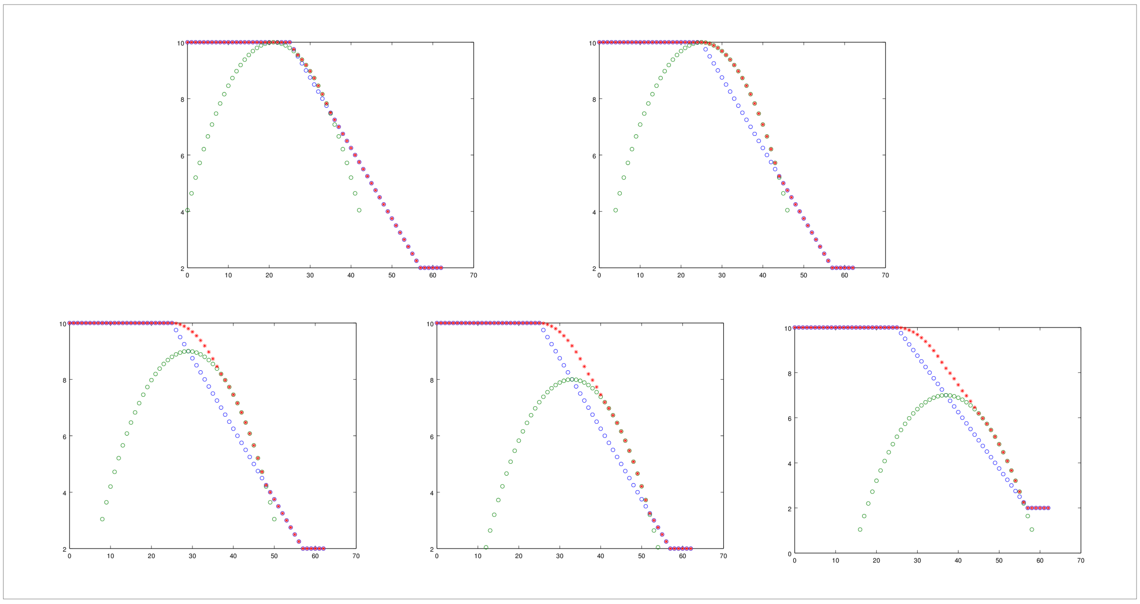


Figure 13: Illustration of the operation of *dilation* (Minkowski sum).

```
end
for l=nt+1:nd-nt-1
  h = z(l) .- t;
  iabove = find(r(l-nt:l+nt)<h);
  r(l-nt-1+iabove) = h(iabove);
end
for l=nd-nt:nd
  h = z(l) .- t(1:nt+1+nd-l);
  iabove = find(r(l-nt:nd)<h);
  r(l-nt-1+iabove) = h(iabove);
end
```

Erosion operates analogously by flipping the sign:

```
  h = z(l) .+ t;
  ibelow = find(r(l-nt:l+nt)>h);
```

The opening filter is carried out by

```
    r = erode_aequidistant(z, dx, Rfilter);
    o = dilate_aequidistant(r, dx, Rfilter);
```

taking off all asperities that are smaller than the structuring element delivering a smoother signal $o(x)$ illustrated as green curve in Fig. 14. The black curve is the input signal $z(x)$ and the red curve the intermediate signal $r(x)$ obtained after erosion.

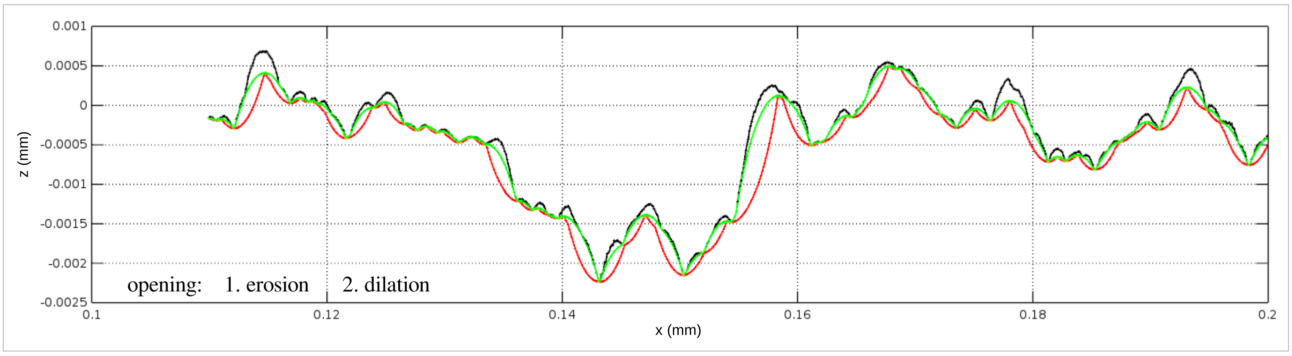


Figure 14: The opening filter is the successive application of *erosion* (Minkowski difference) and then *dilation* (Minkowski sum).

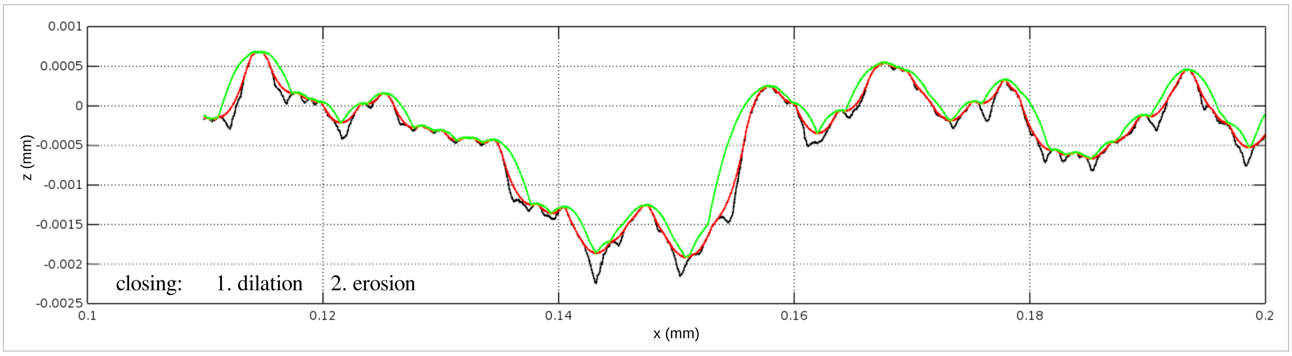


Figure 15: The closing filter is the successive application of *dilation* (Minkowski sum) and then *erosion* (Minkowski difference).

Then the closing is obtained by

```
d = dilate_aequidistant(z, dx, Rfilter);
c = erode_aequidistant(d, dx, Rfilter);
```

filling holes that are smaller than the structuring element delivering a smoother signal $c(x)$ illustrated as red curve in Fig. 15. The black curve is the input signal $z(x)$ and the green curve the intermediate signal $d(x)$ obtained after dilation. ISO standard 16610-41 describes a circular and a linear structuring element.

The filter chains start off with very small features / structures and succesively smooth the signal by removing more and more details meaning enlarging the scale / shrinking the resolution. The sieving is started with the smallest mesh size successively separating grains of growing size. The filtration chain (ladder structure) is described in ISO 16610-49.

4 Robust Estimation for Low Pass Filters

If there is a bias to the distribution of residuals to a model whose parameters are to be estimated with respect to the maximum likelihood (MLE) presumption of Gaussian distributed probabilities showing up as long (sometimes even bumpy) tails, the estimation process needs to be changed. Multiplication of the residuals respectivley their squares with weights that change their distribution such that the tails are suppressed forces the parameters to be estimated without being influenced by outliers (M-estimator). In ISO 16610-31 a filter [6] has been standardized that

1. uses *Tuckey*'s biweight function for M-estimation to be insensitive to strong tails / outliers and
2. the *Gaussian kernel* function combined with *Savitzky-Golay* coefficients for polynomial degree $p = 2$ to preserve the second statistical moment, i.e. the shape of peaks (shape retainment).

The effect of a weight function, here *Tuckey*'s biweight function

$$\delta_l = \begin{cases} \left(1 - \left(\frac{z_l^{\mathtt{P}} - z_l^{\mathtt{W}}}{c_{\mathrm{B}}}\right)^2\right)^2 & \text{if} \quad |z_l^{\mathtt{P}} - z_l^{\mathtt{W}}| \leq c_{\mathrm{B}} \\ 0 & \text{if} \quad |z_l^{\mathtt{P}} - z_l^{\mathtt{W}}| > c_{\mathrm{B}} \end{cases} \qquad c_{\mathrm{B}} \;=\; 4,4478 \,\mathrm{median}(\,|z_k^{\mathtt{P}} - z_k^{\mathtt{W}}|\,) \tag{38}$$

has been shown with an example of a topography that has been eroded (numerically) on its top asperities in Fig. 16.

The weights for the M-estimation are a function of the residuals known *after* estimation but are as well needed in order to estimate, such that the optimization process has to be performed iteratively, as shown in Fig. 17. Furthermore, the weights for the M-estimation are a function of the absolute position of the height values thus they are convolved with the filter kernel of the Gaussian and

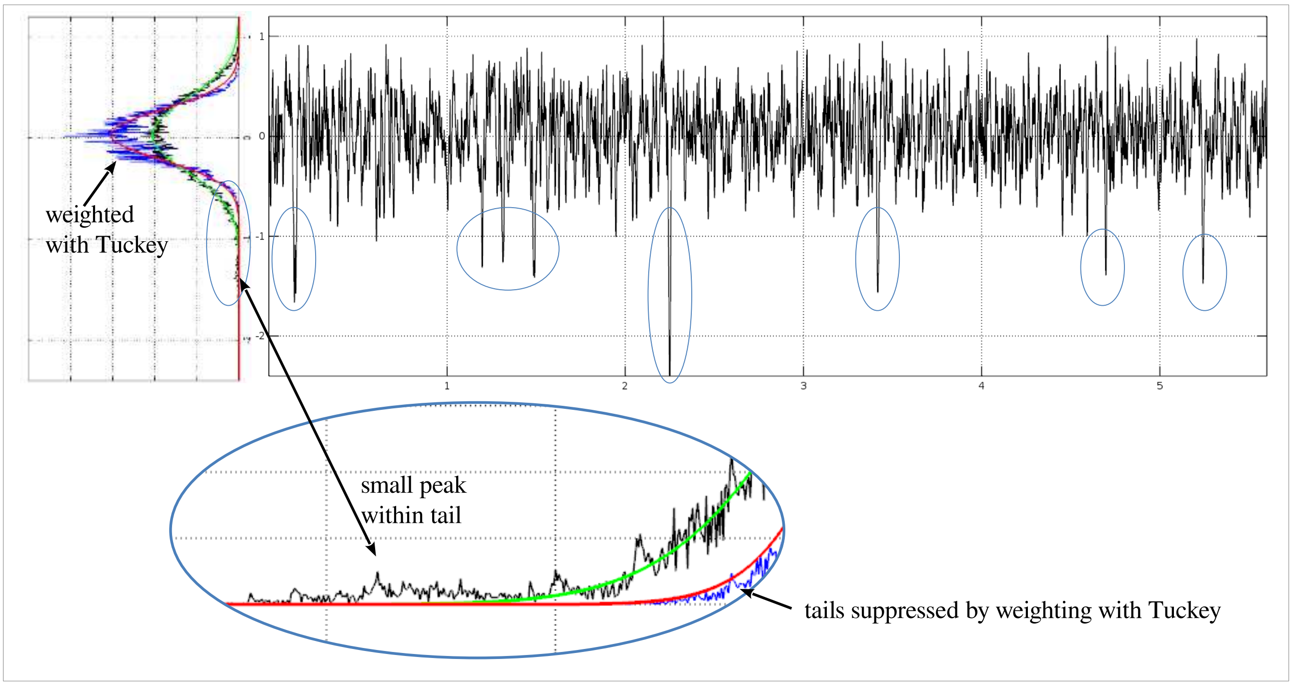


Figure 16: Change of amplitude density (height) distribution by weighting to make it more Gaussian like to be compatible with maximum likelihood estimation. Giving weights to residuals is called M-estimation.

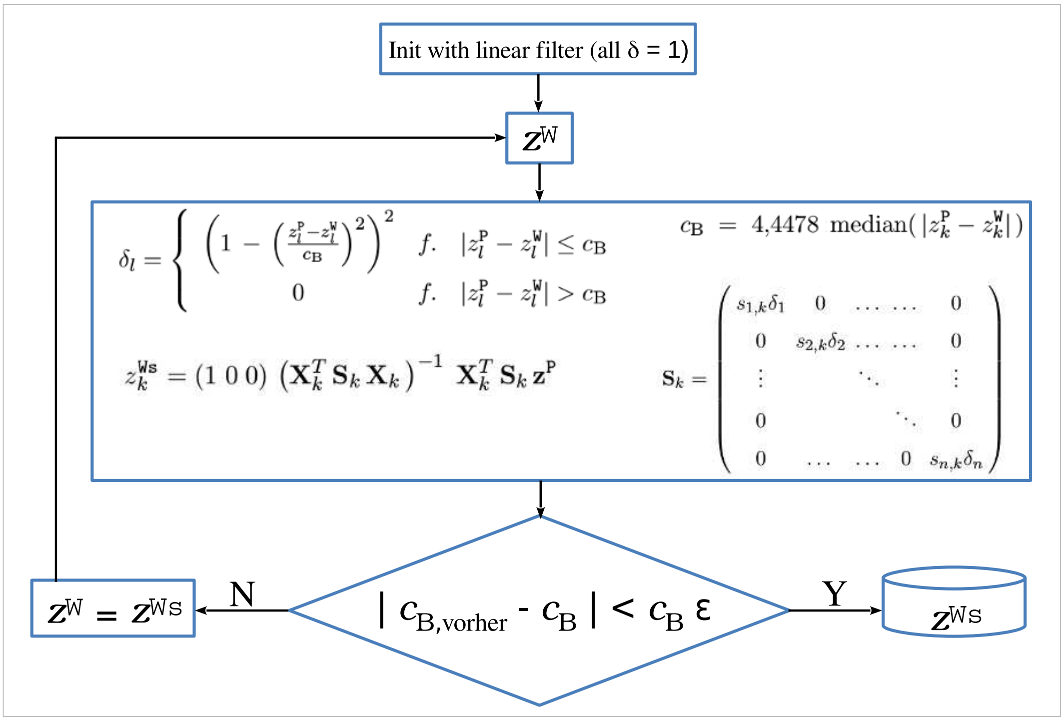


Figure 17: M-estimation uses weights that are a function of the residuals such that the optimization functional is implicit. The solution is searched iteratively.

Savitzky-Golay coefficients:

$$z_k^{\mathtt{Ws}} = (1\ 0\ 0)\ \left(\mathbf{X}_k^T\,\mathbf{S}_k\,\mathbf{X}_k\right)^{-1}\ \mathbf{X}_k^T\,\mathbf{S}_k\,\mathbf{z}^{\mathtt{P}} \tag{39}$$

with

$$\mathbf{X}_k^T\,\mathbf{S}_k\,\mathbf{X}_k = \begin{pmatrix} \sum_{l=1}^n ((l-k)\Delta x)^2\, s_{l,k}\,\delta_l & \sum_{l=1}^n ((l-k)\Delta x)^3\, s_{l,k}\,\delta_l & \sum_{l=1}^n ((l-k)\Delta x)^4\, s_{l,k}\,\delta_l \\ \sum_{l=1}^n ((l-k)\Delta x)\, s_{l,k}\,\delta_l & \sum_{l=1}^n ((l-k)\Delta x)^2\, s_{l,k}\,\delta_l & \sum_{l=1}^n ((l-k)\Delta x)^3\, s_{l,k}\,\delta_l \\ \sum_{l=1}^n\, s_{l,k}\,\delta_l & \sum_{l=1}^n ((l-k)\Delta x)\, s_{l,k}\,\delta_l & \sum_{l=1}^n ((l-k)\Delta x)^2\, s_{l,k}\,\delta_l \end{pmatrix} \tag{40}$$

and with

$$\mathbf{X}_k^T\,\mathbf{S}_k\,\mathbf{z}^{\mathtt{P}} = \begin{pmatrix} \sum_{l=1}^n ((l-k)\Delta x)^2\, s_{l,k}\,\delta_l\, z_l^{\mathtt{P}} \\ \sum_{l=1}^n ((l-k)\Delta x)\, s_{l,k}\,\delta_l\, z_l^{\mathtt{P}} \\ \sum_{l=1}^n\, s_{l,k}\,\delta_l\, z_l^{\mathtt{P}} \end{pmatrix} \tag{41}$$

The lateral position's matrix $\mathbf{X}$ now is defined as

$$\mathbf{X}_k = \begin{pmatrix} 1 & (1-k)\Delta x & ((1-k)\Delta x)^2 \\ 1 & (2-k)\Delta x & ((2-k)\Delta x)^2 \\ \vdots & \vdots & \vdots \\ 1 & (n-k)\Delta x & ((n-k)\Delta x)^2 \end{pmatrix} \tag{42}$$

Furthermore

$$\mathbf{S}_k = \begin{pmatrix} s_{1,k}\delta_1 & 0 & \dots & \dots & 0 \\ 0 & s_{2,k}\delta_2 & \dots & \dots & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & & & \ddots & 0 \\ 0 & \dots & \dots & 0 & s_{n,k}\delta_n \end{pmatrix} \qquad \text{and} \qquad \mathbf{z} = \begin{pmatrix} z_1 \\ z_2 \\ \vdots \\ z_n \end{pmatrix} \tag{43}$$

Each one of the coefficients of the 3×3-matrix $\mathbf{X}_k^T \mathbf{S}_k \mathbf{X}_k$ and of the threedimensional vector $\mathbf{X}_k^T \mathbf{S}_k \mathbf{z}$ represents a convolution of vector

$$\vec{s}_\nu = \begin{pmatrix} 0 \\ (\Delta x)^\nu \, e^{-(\frac{1}{2}\left(\frac{\Delta x}{\lambda_\mathrm{c}\Lambda_\mathrm{TP}}\right)^2)} \\ (2\Delta x)^\nu \, e^{-(\frac{1}{2}\left(\frac{2\Delta x}{\lambda_\mathrm{c}\Lambda_\mathrm{TP}}\right)^2)} \\ \vdots \\ (m\Delta x)^\nu \, e^{-(\frac{1}{2}\left(\frac{m\Delta x}{\lambda_\mathrm{c}\Lambda_\mathrm{TP}}\right)^2)} \\ ((m-1)\Delta x)^\nu \, e^{-(\frac{1}{2}\left(\frac{(m-1)\Delta x}{\lambda_\mathrm{c}\Lambda_\mathrm{TP}}\right)^2)} \\ \vdots \\ (2\Delta x)^\nu \, e^{-(\frac{1}{2}\left(\frac{2\Delta x}{\lambda_\mathrm{c}\Lambda_\mathrm{TP}}\right)^2)} \\ (\Delta x)^\nu \, e^{-(\frac{1}{2}\left(\frac{\Delta x}{\lambda_\mathrm{c}\Lambda_\mathrm{TP}}\right)^2)} \end{pmatrix} \qquad \begin{array}{l} \nu = 0, 1, 2, 3, 4 \\ \\ m = (n + \lambda_c/\Delta x)/2 \end{array} \tag{44}$$

with one of the following vectors accordingly

$$\vec{\delta} = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad \vec{Z}_\delta = \begin{pmatrix} \delta_1 \, z_1^\mathtt{P} \\ \delta_2 \, z_2^\mathtt{P} \\ \vdots \\ \delta_n \, z_n^\mathtt{P} \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{45}$$

The Gaussian kernel as defined in Eq. (28) may be rewritten with factor $\frac{1}{2}$ as known for normal distributions rather than π

$$s_\nu(x) = x^\nu \; e^{-(\pi\left(\frac{x}{\gamma\,\lambda_\mathrm{c}}\right)^2)} = x^\nu \; e^{-(\frac{1}{2}\left(\frac{x}{\lambda_\mathrm{c}\,\Lambda_\mathrm{TP}}\right)^2)} \tag{46}$$

with $\Lambda_\mathrm{TP} = 0.2915922697$, see Seewig [6].

To optimize computing time we perform multiplications in Fourier domain rather than convolving in spatial domain:

$$\sum_{l=1}^{n}((l-k)\Delta x)^{\nu} s_{l,k}\delta_l = \mathcal{R}e\left(\mathcal{F}_k^{-1}(\vec{\mathcal{A}}_\nu)\right) \qquad \sum_{l=1}^{n}((l-k)\Delta x)^{\nu} s_{l,k}\delta_l\, z_l = \mathcal{R}e\left(\mathcal{F}_k^{-1}(\vec{\mathcal{B}}_\nu)\right) \tag{47}$$

with

$$\vec{\mathcal{A}}_\nu = \begin{pmatrix} \mathcal{F}_0(\vec{s}_\nu)\mathcal{F}_0(\vec{\delta}) \\ \mathcal{F}_1(\vec{s}_\nu)\mathcal{F}_1(\vec{\delta}) \\ \vdots \\ \mathcal{F}_{2m}(\vec{s}_\nu)\mathcal{F}_{2m}(\vec{\delta}) \end{pmatrix} \qquad \vec{\mathcal{B}}_\nu = \begin{pmatrix} \mathcal{F}_0(\vec{s}_\nu)\mathcal{F}_0(\vec{Z}) \\ \mathcal{F}_1(\vec{s}_\nu)\mathcal{F}_1(\vec{Z}) \\ \vdots \\ \mathcal{F}_{2m}(\vec{s}_\nu)\mathcal{F}_{2m}(\vec{Z}) \end{pmatrix} \tag{48}$$

with

$$\vec{\mathcal{F}}(\mathbf{a}) = \begin{pmatrix} \mathcal{F}_0(\mathbf{a}) \\ \mathcal{F}_1(\mathbf{a}) \\ \mathcal{F}_2(\mathbf{a}) \\ \vdots \\ \mathcal{F}_{2m}(\mathbf{a}) \end{pmatrix} = \begin{pmatrix} \sum_{l=-\infty}^{\infty} a_l \\ \sum_{l=-\infty}^{\infty} a_l\, e^{-\mathrm{i}\,2\pi\,\frac{1}{L}\,l\Delta x} \\ \sum_{l=-\infty}^{\infty} a_l\, e^{-\mathrm{i}\,2\pi\,\frac{2}{L}\,l\Delta x} \\ \vdots \\ \sum_{l=-\infty}^{\infty} a_l\, e^{-\mathrm{i}\,2\pi\,\frac{2m}{L}\,l\Delta x} \end{pmatrix} = \begin{pmatrix} \int_{-\infty}^{\infty} a(x)\,\mathrm{d}x \\ \int_{-\infty}^{\infty} a(x)\, e^{-\mathrm{i}\,2\pi\,\frac{1}{L}\,x}\,\mathrm{d}x \\ \int_{-\infty}^{\infty} a(x)\, e^{-\mathrm{i}\,2\pi\,\frac{2}{L}\,x}\,\mathrm{d}x \\ \vdots \\ \int_{-\infty}^{\infty} a(x)\, e^{-\mathrm{i}\,2\pi\,\frac{2m}{L}\,x}\,\mathrm{d}x \end{pmatrix} \tag{49}$$

$$\vec{\mathcal{F}}^{-1}(\mathbf{b}) = \begin{pmatrix} \mathcal{F}_0^{-1}(\mathbf{b}) \\ \mathcal{F}_1^{-1}(\mathbf{b}) \\ \mathcal{F}_2^{-1}(\mathbf{b}) \\ \vdots \\ \mathcal{F}_{2m}^{-1}(\mathbf{n}) \end{pmatrix} = \begin{pmatrix} \sum_{l=-\infty}^{\infty} b_l \\ \sum_{l=-\infty}^{\infty} b_l\, e^{\mathrm{i}\,2\pi\,\frac{l}{L}\,\Delta x} \\ \sum_{l=-\infty}^{\infty} b_l\, e^{\mathrm{i}\,2\pi\,\frac{l}{L}\,2\Delta x} \\ \vdots \\ \sum_{l=-\infty}^{\infty} b_l\, e^{\mathrm{i}\,2\pi\,\frac{l}{L}\,2m\Delta x} \end{pmatrix} = \begin{pmatrix} \int_{-\infty}^{\infty} b(f)\,\mathrm{d}f \\ \int_{-\infty}^{\infty} b(f)\, e^{\mathrm{i}\,2\pi\,f\,\Delta x}\,\mathrm{d}f \\ \int_{-\infty}^{\infty} b(f)\, e^{\mathrm{i}\,2\pi\,f\,2\Delta x}\,\mathrm{d}f \\ \vdots \\ \int_{-\infty}^{\infty} b(f)\, e^{\mathrm{i}\,2\pi\,f\,2m\Delta x}\,\mathrm{d}f \end{pmatrix} \tag{50}$$

The terms with the Gaussian kernel can be transformed to Fourier domain analytically and thus implemented as terms that already exist as Fourier transforms without using any numerical Fourier transformation for them:

$$\mathcal{F}_l(\vec{s}_\nu) \equiv F_{\nu,l} = \frac{1}{\sqrt{2\pi}\,\lambda_\mathrm{c}\,\Lambda_\mathrm{TP}} \int_{-\infty}^{\infty} x^\nu\, e^{-(\frac{1}{2}\left(\frac{x}{\lambda_\mathrm{c}\,\Lambda_\mathrm{TP}}\right)^2)}\, e^{-\mathrm{i}\,2\pi\,\frac{l}{L}\,x}\,\mathrm{d}x \tag{51}$$

To be short, we write $a_1 = \lambda_c \Lambda_{TP}$ for the constants of the Gaussian kernel:

$$s(x) = \; e^{-(\frac{1}{2}\left(\frac{x}{a_1}\right)^2)} \tag{52}$$

Be L the length of the profile, n the number of data points, and λ_c the cut off wave length as nesting index, then the spatial frequencies f are given by

$$f = l f_0 \qquad \text{with} \qquad l = 0, \ldots, n-1 \tag{53}$$

with $f_0 = \frac{2\pi}{L}$.

$$F_\nu = \int_{-\infty}^{\infty} x^\nu \, e^{-(\frac{1}{2}\left(\frac{x}{a_1}\right)^2)} \, e^{-ilf_0 x} \mathrm{d}x; \tag{54}$$

with $d_1 = a_1 f_0$

$$F_0(x) = \sqrt{2\pi}\, a_1 \, e^{-\frac{1}{2}\,(d_1 l)^2}$$

$$F_1(x) = -\, i \sqrt{2\pi} a_1 \; a_1 \; d_1 \, l \, e^{-\frac{1}{2}\,(d_1 l)^2}$$

$$F_2(x) = \sqrt{2\pi}\, a_1 \, a_1^2 \, \left(1 \, - \, (d_1 l)^2\right) \, e^{-\frac{1}{2}\,(d_1 l)^2}$$

$$F_3(x) = i\, \sqrt{2\pi}\, a_1 \, a_1^3 \, \left((d_1 l)^3 \, - 3\,(d_1 l)\right) \, e^{-\frac{1}{2}\,(d_1 l)^2}$$

$$F_4(x) = \sqrt{2\pi}\, a_1 \, a_1^4 \, \left((d_1 l)^4 \, - \, 6\,(d_1 l)^2 + 3\right) \, e^{-\frac{1}{2}\,(d_1 l)^2}$$

and with

$$d = \lambda_c \, \Lambda_{TP} \, \frac{2\pi}{L}$$

$$F_{0,l} = e^{-\frac{1}{2}\,(d\,l)^2}\,; \qquad F_{1,l} = -\,\mathrm{i}\,(\lambda_c \Lambda_{TP})\,(d\,l)\, e^{-\frac{1}{2}\,(d\,l)^2}\,; \qquad F_{2,l} = (\lambda_c \Lambda_{TP})^2 \, \left(1 \, - \, (d\,l)^2\right) \, e^{-\frac{1}{2}\,(d\,l)^2}\,;$$

$$F_{3,l} = \mathrm{i}\,(\lambda_c \Lambda_{TP})^3 \, \left((d\,l)^3 \, - 3\,(d\,l)\right) \, e^{-\frac{1}{2}\,(d\,l)^2}\,; \qquad F_{4,l} = (\lambda_c \Lambda_{TP})^4 \, \left((d\,l)^4 \, - \, 6\,(d\,l)^2 + 3\right) \, e^{-\frac{1}{2}\,(d\,l)^2}$$

As robust spline in ISO 16610-32 a version with L1-norm is proposed. Recalling Eq. (12) for the tension parameter $\beta = 0$

$$\mathcal{E} = \sum_{i=1}^{n-1} (z(x_i) \; - \; w_i)^2 \; + \; r \int_0^L \left(\frac{\partial^2}{\partial x^2} s(x) \right)^2 dx \rightarrow \min_{w_i} \tag{55}$$

it could as well be modified to a robust version either by introducing an appropriate weighting $\Psi(z(x_i) \; - \; w_i)$ for an M-estimation

$$\mathcal{E} = \sum_{i=1}^{n-1} \Psi \; (z(x_i) \; - \; w_i)^2 \; + \; r \int_0^L \left(\frac{\partial^2}{\partial x^2} s(x) \right)^2 dx \rightarrow \min_{w_i} \tag{56}$$

or by choosing L1-norm either

$$\mathcal{E} = \sum_{i=1}^{n-1} |z(x_i) \; - \; w_i| \; + \; r \int_0^L \left| \frac{\partial^2}{\partial x^2} s(x) \right| dx \rightarrow \min_{w_i} \tag{57}$$

which causes problems since the absolute value is not continous or

$$\mathcal{E} = \sum_{i=1}^{n-1} c \, |z(x_i) \; - \; w_i| \; + \; r \int_0^L \left(\frac{\partial^2}{\partial x^2} s(x) \right)^2 dx \rightarrow \min_{w_i} \tag{58}$$

requiring an appropriate factor c to have both terms of the same physical dimension, here to carry the dimension of a length. Seewig suggests in Leach *Characterisation of Areal Surface Texture* [15], chapter 4.2, following definition of c interpreting roughness profiles being a superposition of sinusoidal waves of various wavelengths (with random amplitudes and phases):

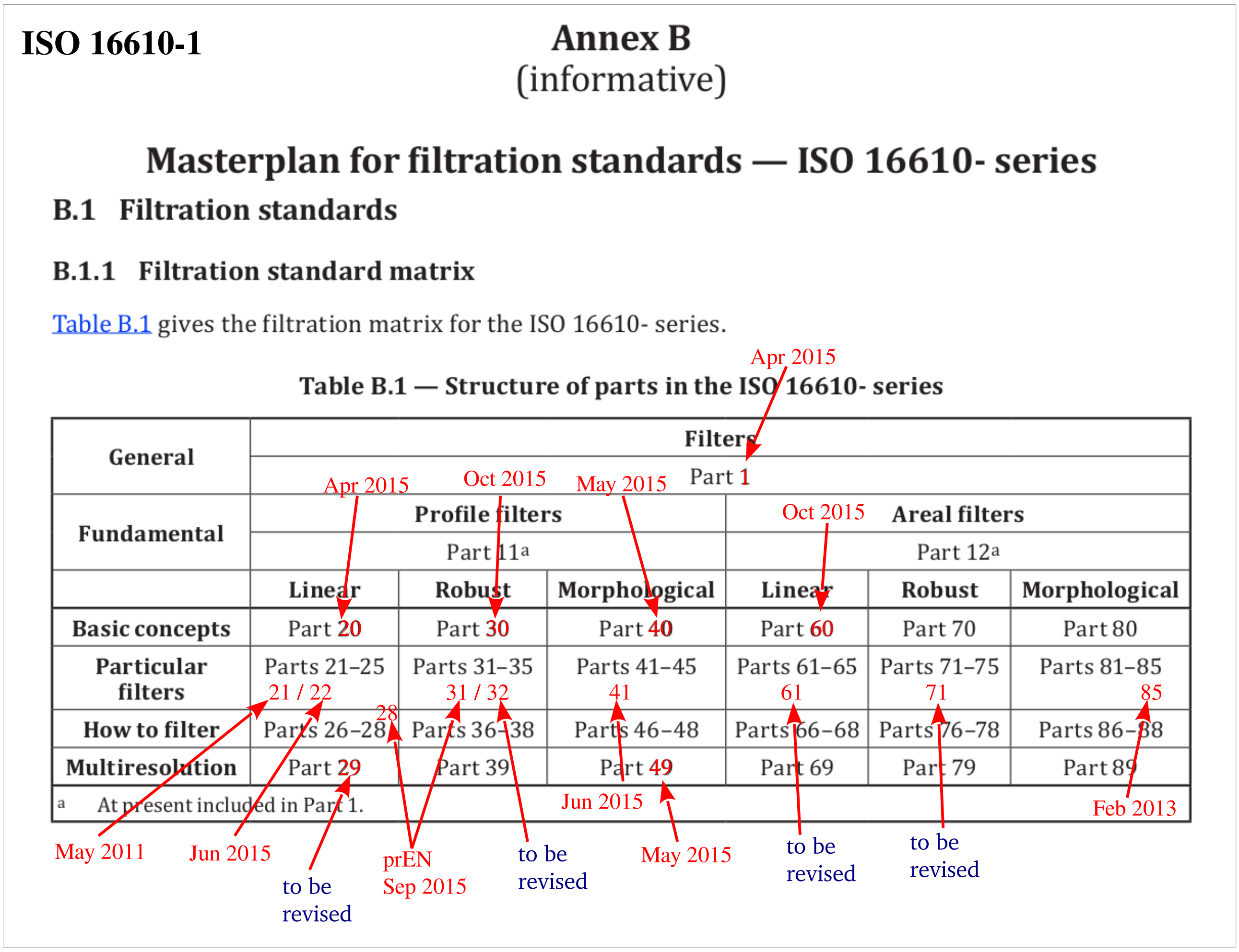
$$c \; = \; \frac{\pi}{2} \, \frac{1}{n} \sum_{k=1}^{n} |z(x_k) \; - \; w_k| \tag{59}$$

With tension term and analogously to the filter equation (15) we have

$$\left(\mathbf{V} \; + \; \beta \alpha^2 \mathbf{P} \; + \; (1-\beta) \alpha^4 \mathbf{Q} \right) \mathbf{z}^{\mathtt{Ws}} = \mathbf{V} \mathbf{z}^{\mathtt{P}} \tag{60}$$

with the column vectors for the waviness $\mathbf{z}^{\mathtt{W}} := (w_1, \ldots, w_n)^T$ respectively the robustly estimated waviness $\mathbf{z}^{\mathtt{Ws}} := (w_1, \ldots, w_n)^T$ and the primary profile $\mathbf{z}^{\mathtt{P}} := (z(x_1), \ldots, z(x_n))^T$:

$$\mathbf{V} = \begin{pmatrix} \frac{c}{|z(x_1)-w_1|} & 0 & \ldots & \ldots & 0 \\ 0 & \ddots & 0 & & \vdots \\ \vdots & 0 & \ddots & 0 & \vdots \\ \vdots & & & \ddots & 0 \\ 0 & \ldots & \ldots & 0 & \frac{c}{|z(x_n)-w_n|} \end{pmatrix} \tag{61}$$



ISO 16610-1

Annex B
(informative)

Masterplan for filtration standards — ISO 16610- series

B.1 Filtration standards

B.1.1 Filtration standard matrix

Table B.1 gives the filtration matrix for the ISO 16610- series.

Table B.1 — Structure of parts in the ISO 16610- series

General	**Filters**					
	Part 1					
Fundamental	**Profile filters**			**Areal filters**		
	Part 11[a]			Part 12[a]		
	Linear	**Robust**	**Morphological**	**Linear**	**Robust**	**Morphological**
Basic concepts	Part 20	Part 30	Part 40	Part 60	Part 70	Part 80
Particular filters	Parts 21–25	Parts 31–35	Parts 41–45	Parts 61–65	Parts 71–75	Parts 81–85
How to filter	Parts 26–28	Parts 36–38	Parts 46–48	Parts 66–68	Parts 76–78	Parts 86–88
Multiresolution	Part 29	Part 39	Part 49	Part 69	Part 79	Part 89

[a] At present included in Part 1.

Figure 18: Survey on standards on filtration.

As for the robust regression Gaussian, here again, we have an implicite equation that is solved iteratively using the waviness values w_k of the linear filter as start values included into the factor c and the matrix $\mathbf{V}$ and then for each step the predicessor waviness.

5 The ISO Toolbox

Basic concepts of **linear filters** are given for profiles in ISO 16610-20 and for areal data in ISO 16610-60. As specific filters, those with [Regressian]-Gaussian (16610-21 / -28 for profiles and 16610-61 for areal data) are standardized and that with smoothing spline in 16610-22 for profiles.

Basics concepts of **robust filters** are given for profiles in ISO 16610-30 and for areal data in ISO 16610-70. As specific filters, those with Regressian-Gaussian (16610-31 for profiles and 16610-71 for areal data) are standardized and that with smoothing spline in 16610-32 for profiles.

The definitions of the basic operations for **mathematical morphology** are given in ISO 16610-40. Specific suggestions for using a circular disk and a horizontal line segment are given in ISO 16610-41. A segmentation procedure for areal topography maps is given in ISO 16610-85.

Filtration chains for **hierarchically sieving** of different sizes are specified is ISO 16610-29 (using discrete wavelet transforms) and ISO 16610-49 (using morphological filters).

Fig. 18 displays the master plan of the ISO 16610 toolbox with the availability state by September 2015.

References

[1] D. Terzopoulos. Regularization of inverse visual problems involving discontinuities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-8(4), 1986. `doi:10.1109/TPAMI.1986.4767807`.

[2] M. Unser, A. Aldroubi, and M. Eden. Recursive regularization filters: Design, properties, and applications. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(3), 1991. `doi:10.1109/34.75514`.

[3] C. de Boor. On calculating with B-splines. *J. Approx. Theory*, 6:50–62, 1972.

[4] B. P. Flannery, W. H. Press, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 2. edition, 1992-2002.

[5] M. Krystek. Transfer functions of discrete spline filters. In *Advanced Mathematical Tools in Metrology III*, number 45 in Series on Advances in Mathematics for Applied Sciences, pages 203–210, 1997. `doi:10.1142/9789814530293`.

[6] Jörg Seewig. *Praxisgerechte Signalverarbeitung zur Trennung der Gestaltabweichungen technischer Oberflächen*. Shaker, 2000.

[7] J. Seewig. Linear and robust gaussian regression filters. *Journal of Physics: Conference Series*, 13(1):254, 2005. `doi:10.1088/1742-6596/13/1/059`.

[8] M. Unser, A. Aldroubi, and M. Eden. Polynomial spline signal approximations: filter design and asymptotic equivalence with shannon's sampling theorem. *IEEE Trans. Information*, 38(1), 1992. `doi:10.1109/18.108253`.

[9] M. Unser, A. Aldroubi, and M. Eden. A family of polynomial spline wavelet transforms. *Signal Processing*, 30(2), 1993. `doi:10.1016/0165-1684(93)90144-Y`.

[10] M. Unser. Ten good reasons for using spline wavelets. In *Wavelet Applications in Signal and Image Processing V*, number 422-431 in Proc. SPIE, 1997. `doi:10.1117/12.292801`.

[11] P. Maragos and R. W. Schafer. Morphological filters - part 1: Their set-theoretic analysis and relations to linear shift-invariant filters. *IEEE Trans. Acoust, Speech, Signal Processing*, 35:1153–1169, 1987.

[12] H. J. A. M. Heijmans. Mathematical morphology: A modern approach in image processing based on algebra and geometry. *SIAM Review*, 37(1):1–36, 1995.

[13] D. J. Keller and F. S. Franke. Envelope reconstruction of probe microscope images. *Surf. Sci.*, 294:409–419, 1993.

[14] J.S. Villarrubia. Algorithms for scanned probe microscope image simulation, surface reconstruction, and tip estimation. *J. Res. Natl. Inst. Stand. Technol*, 102(4):425–454, 1997. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.42.4414.

[15] Richard Leach (Ed). *Characterisation of Areal Surface Texture*. Springer, 2013. `doi:10.1007/978-3-642-36458-7`.

[16] G. Fernández, S. Periaswamy, and W. Sweldens. Liftpack: A software package for wavelet transforms using lifting. In *Wavelet Applications in Signal and Image Processing IV*, number 396 in Proc. SPIE, 1996. `doi:10.1117/12.255250`.

Annex

In order to interpolate between signal values of neighboring knots, often polynomials are chosen, such as Neville polynomial as it is the case for Fernandez-Periaswamy-Sweldens lifting wavelets used in Annex A of ISO 16610-29:

The interpolated value for some position x_{interp} between the knots of given values $(z_\nu, z_{\nu+1}, \ldots, z_{\nu+p})$ be z_{interp}. The coefficients of the polynomial $(\beta_0, \beta_1, \ldots, \beta_p)$ are to be estimated from the $p+1$ signal values $(z_\nu, z_{\nu+1}, \ldots, z_{\nu+p})$. In ISO 16610-29 cubic polynomials are chosen, i.e. $p = 3$ and for knots not directly at the border, the mid position is chosen for interpolation. Furthermore, the polynomial degree is called p in ISO 16610-31 but $\tilde{N}$ in ISO 16610-29, because Fernandez, Periaswamy and Sweldens called it $\tilde{N}$, [16].

Once the coefficients $(\beta_0, \beta_1, \ldots, \beta_p)$ are estimated, the interpolated signal would be

$$z_{\mathrm{interp}} = \sum_{j=0}^{p} \beta_j \, x_{\mathrm{interp}}^j \tag{62}$$

and to obtain the $(p + 1)$ coefficients, we solve the $(p + 1) \times (p + 1)$ linear equation system (this is

Eq (5) in the liftpack paper [16]).

$$z_{\nu+k} = \sum_{j=0}^{p} \beta_j x_{\nu+k}^j \qquad k = 0, \ldots p \tag{63}$$

i.e.

$$\begin{pmatrix} z_\nu \\ \vdots \\ z_{\nu+k} \\ \vdots \\ z_{\nu+p} \end{pmatrix} = \begin{pmatrix} x_{\nu+0}^0 & \ldots & x_{\nu+0}^j & \ldots & x_{\nu+0}^p \\ \vdots & & \vdots & & \vdots \\ x_{\nu+k}^0 & \ldots & x_{\nu+k}^j & \ldots & x_{\nu+k}^p \\ \vdots & & \vdots & & \vdots \\ x_{\nu+p}^0 & \ldots & x_{\nu+p}^j & \ldots & x_{\nu+p}^p \end{pmatrix} \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_j \\ \vdots \\ \beta_p \end{pmatrix} \tag{64}$$

inserting Eq. (64) into Eq. (62)

$$z_{\text{interp}} \;=\; \begin{pmatrix} x_{\text{interp}}^0 & \ldots & x_{\text{interp}}^j & \ldots & x_{\text{interp}}^p \end{pmatrix} \begin{pmatrix} \beta_0 \\ \vdots \\ \beta_j \\ \vdots \\ \beta_p \end{pmatrix}$$

delivers

$$z_{\text{interp}} \;=\; \begin{pmatrix} x_{\text{interp}}^0 & \ldots & x_{\text{interp}}^j & \ldots & x_{\text{interp}}^p \end{pmatrix} \begin{pmatrix} & \vdots & \\ \ldots & x_{\nu+k}^j & \ldots \\ & \vdots & \end{pmatrix}^{-1} \begin{pmatrix} z_\nu \\ \vdots \\ z_{\nu+k} \\ \vdots \\ z_{\nu+p} \end{pmatrix} \tag{65}$$

The filter coefficients $l_0, \ldots, l_p$ to interpolate at position x_{interp}

$$z_{\text{interp}} \;=\; \vec{l}^{\,T} \begin{pmatrix} z_\nu \\ \vdots \\ z_{\nu+k} \\ \vdots \\ z_{\nu+p} \end{pmatrix} \tag{66}$$

with $\vec{l}^{\,T} \;=\; (l_0,\, \ldots,\, l_p)$ are the Neville's polynomial coefficients and as well the Sweldens' wavelet for the predict step to calculate the residual $d_{\text{odd}} = z_{\text{signal}}(x_{\text{odd}}) - z_{\text{interp}}(x_{\text{odd}})$ with

$$z_{\text{interp}}(x_{\text{odd}}) = \vec{l}^{\,T} \begin{pmatrix} z_{\text{even}-2*(p+1)/2} \\ \vdots \\ z_{\text{even}-2=\text{odd}-1} \\ z_{\text{even}=\text{odd}+1} \\ z_{\text{even}+2=\text{odd}+3} \\ \\ \vdots \\ z_{\text{even}+2*(p+1)/2} \end{pmatrix} \tag{67}$$

and furthermore for the update step to calculate the lifting residuals for smoothing $z_{\text{smooth}}(x_{\text{even}}) = z_{\text{signal}}(x_{\text{even}}) + d_{\text{smooth}}$

$$d_{\text{smooth}} = \vec{l}^{\,T} \begin{pmatrix} d_{\text{odd}-2*(p+1)/2} \\ \vdots \\ d_{\text{odd}-2} \\ d_{\text{odd}} \\ d_{\text{odd}+2} \\ \\ \vdots \\ d_{\text{odd}+2*(p+1)/2} \end{pmatrix} \tag{68}$$

The coefficients can be obtained generally for aequidistantly sampled signals, as the sampling interval cancels when the fraction is reduced.

$$\vec{l}^{\,T} = \begin{pmatrix} x^0_{\text{interp}} & \dots & x^j_{\text{interp}} & \dots & x^p_{\text{interp}} \end{pmatrix} \begin{pmatrix} & \vdots & \\ \dots & x^j_{\nu+k} & \dots \\ & \vdots & \end{pmatrix}^{-1} \tag{69}$$

where the $x_{\nu+k}$ are those at the even indexed positions, and x_{interp} the odd indexed one in the middle.

ISO 16610-29 proposes to use cubic polynomials, i.e. $p = 3$. Let's set the x-value for the middle one to be interpolated $x_{\text{interp}} = 0$ and those left and right of it at $(-3, -1, 1, 3)$ not confusing x-positions with indices. Rewriting Eq. 69 specifically, we obtain

$$\vec{l}^{\,T} = \begin{pmatrix} 0^0 & 0^1 & 0^2 & 0^3 \end{pmatrix} \begin{pmatrix} (-3)^0 & (-3)^1 & (-3)^2 & (-3)^3 \\ (-1)^0 & (-1)^1 & (-1)^2 & (-1)^3 \\ 1^0 & 1^1 & 1^2 & 1^3 \\ 3^0 & 3^1 & 3^2 & 3^3 \end{pmatrix}^{-1} \tag{70}$$

i.e.

$$\vec{l}^{\,T} \;=\; (1\,0\,0\,0)\begin{pmatrix} 1 & -3 & 9 & -27 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & 1 & 1 \\ 1 & 3 & 9 & 27 \end{pmatrix}^{-1} \tag{71}$$

i.e.

$$\vec{l}^{\,T} \;=\; (1\,0\,0\,0)\begin{pmatrix} -\frac{1}{16} & \frac{9}{16} & \frac{9}{16} & -\frac{1}{16} \\ \frac{1}{48} & -\frac{9}{16} & \frac{9}{16} & -\frac{1}{48} \\ \frac{1}{16} & -\frac{1}{16} & -\frac{1}{16} & \frac{1}{16} \\ -\frac{1}{48} & \frac{1}{16} & -\frac{1}{16} & \frac{1}{48} \end{pmatrix} \;=\; \left(-\frac{1}{16}\;\frac{9}{16}\;\frac{9}{16}\;\;-\frac{1}{16}\right) \tag{72}$$

If the polynomial shall be a regression polynomial not just going exactly through the $p+1$ (resp. $\tilde{N}+1$) but been fitted to $N > p+1$ signal values via LMS estimation, instead of a Neville polynomial interpolation, it is called Savitzky-Golay filter. The coefficients of this filter are estimated by minimizing the sum of squares of the residuals meaning that there are small finite differences between the signal values z_ν and the polynomial w_ν. The Neville polynomial, however, takes the values of the signal at the positions $x_{\nu+k}$.

The LMS optimization problem for estimating Savitzky-Golay filter coefficients is

$$\min_{\vec{\beta}_\nu}\left\{\sum_{k=0}^{N}(z_{\nu+k}-w_{\nu+k})^2\right\} \tag{73}$$

with $w_{\nu+k}$ for the approximation by an regression polynomial

$$w_{\nu+k} = \sum_{j=0}^{p}\;\beta_{\nu,j}\;(k\,\Delta x)^j \tag{74}$$

$$\frac{\partial}{\partial\beta_{\nu,m}}\left\{\sum_{k=0}^{N}(z_{\nu+k}-w_{\nu+k})^2\right\} \;=\; 0 \tag{75}$$

i.e.

$$2\sum_{k=0}^{N}(z_{\nu+k}-w_{\nu+k})\,\frac{\partial w_{\nu+k}}{\partial\beta_{\nu,m}} \;=\; 0 \tag{76}$$

i.e.

$$\sum_{k=0}^{N}(z_{\nu+k}\,\frac{\partial w_{\nu+k}}{\partial\beta_{\nu,m}}) \;=\; \sum_{k=0}^{N}(w_{\nu+k}\,\frac{\partial w_{\nu+k}}{\partial\beta_{\nu,m}}) \tag{77}$$

with Eq. 74 and its derivative

$$\frac{\partial w_{\nu+k}}{\partial\beta_{\nu,m}} \;=\; \sum_{j=0}^{p}\;\delta_{j,m}\;(k\,\Delta x)^j \;=\; (k\,\Delta x)^m \tag{78}$$

we obtain

$$\sum_{k=0}^{N} z_{\nu+k}\,(k\,\Delta x)^m \;=\; \sum_{k=0}^{N}\sum_{j=0}^{p}\;\beta_{\nu,j}\;(k\,\Delta x)^j\,(k\,\Delta x)^m \tag{79}$$

rearranging it

$$\sum_{j=0}^{p}\left(\sum_{k=0}^{N}\,(k\,\Delta x)^j\,(k\,\Delta x)^m\right)\,\beta_{\nu,j} \;=\; \sum_{k=0}^{N}(k\,\Delta x)^m\,z_{\nu+k} \tag{80}$$

and writing it in matrix notation

$$\begin{pmatrix} & \vdots & \\ \ldots & \sum_{k=0}^{N}\,(k\,\Delta x)^j\,(k\,\Delta x)^m & \ldots \\ & \vdots & \end{pmatrix}\begin{pmatrix} \vdots \\ \beta_{\nu,j} \\ \vdots \end{pmatrix} \;=\; \begin{pmatrix} \vdots \\ \sum_{k=0}^{N}(k\,\Delta x)^m\,z_{\nu+k} \\ \vdots \end{pmatrix} \tag{81}$$

For the smoothed interpolated signal $w(x)$, i.e. for $w(x) = \sum_{j=0}^{p}\;\beta_{\nu,j}\;x^j$ we use

$$\begin{pmatrix} \vdots \\ \beta_{\nu,j} \\ \vdots \end{pmatrix} \;=\; \begin{pmatrix} & \vdots & \\ \ldots & \sum_{k=0}^{N}\,(k\,\Delta x)^j\,(k\,\Delta x)^m & \ldots \\ & \vdots & \end{pmatrix}^{-1}\begin{pmatrix} \vdots \\ \sum_{k=0}^{N}(k\,\Delta x)^m\,z_{\nu+k} \\ \vdots \end{pmatrix} \tag{82}$$

and choose $k = N/2$ for the position of which the smoothed signal is to be estimated. A symmetric filtering around k is obtained for N being odd: $N = 2n + 1$. Performing a shift of indices in Eq. 83 delivers

$$\begin{pmatrix} \vdots \\ \beta_{\nu,j} \\ \vdots \end{pmatrix} \;=\; \begin{pmatrix} & \vdots & \\ \ldots & \sum_{k=-n}^{n}\,(k\,\Delta x)^j\,(k\,\Delta x)^m & \ldots \\ & \vdots & \end{pmatrix}^{-1}\begin{pmatrix} \vdots \\ \sum_{k=-n}^{n}(k\,\Delta x)^m\,z_{\nu+k} \\ \vdots \end{pmatrix} \tag{83}$$

such that the desired mid position x is position at z_ν and $x = 0 \cdot \Delta x$ and the smoothed signal will be

$$w_\nu \;=\; \sum_{j=0}^{p}\;\beta_{\nu,j}\;0^j \;=\; (0^0\;0^1\;\ldots\;0^p)\begin{pmatrix} \vdots \\ \beta_{\nu,j} \\ \vdots \end{pmatrix} \tag{84}$$

i.e. with $(0^0\;0^1\;\ldots\;0^p)$ being the $(p+1)$-dimensional unity vector $(1\;0\;\ldots\;0)$ we obtain

$$w_\nu \;=\; (1\;0\;\ldots\;0)\begin{pmatrix} & \vdots & \\ \ldots & \sum_{k=-n}^{n}\,(k\,\Delta x)^j\,(k\,\Delta x)^m & \ldots \\ & \vdots & \end{pmatrix}^{-1}\begin{pmatrix} \vdots \\ \sum_{k=-n}^{n}(k\,\Delta x)^m\,z_{\nu+k} \\ \vdots \end{pmatrix} \tag{85}$$

for the final filter equation.

The Savitzky-Golay filter (or regressian polynomial filter) is moment preserving to the p-th moment.

It is combined with the Gaussian filter by including the Gaussian kernel as weights. Here as well it is the mid position where the smoothed signal is desired. Furthermore with indices chosen for a symmetric filtering around k is chosen for N being odd $N = 2n + 1$ as above:

$$\min_{\vec{\beta}_\nu} \left\{ \sum_{k=-n}^{n} s_k \left(z_{\nu+k} - w_{\nu+k} \right)^2 \right\} \tag{86}$$

with

$$s_k = \frac{1}{\alpha\lambda} \, e^{-\pi \left(\frac{k \, \Delta x}{\alpha\lambda} \right)^2} \tag{87}$$

the Savitzky-Golay filter modified by a weight function as Gaussian kernel is referred to as Regression-Gaussian filter.

$$\sum_{k=-n}^{n} s_k \, z_{\nu+k} \frac{\partial w_{\nu+k}}{\partial \beta_{\nu,m}} = \sum_{k=-n}^{n} s_k \, w_{\nu+k} \frac{\partial w_{\nu+k}}{\partial \beta_{\nu,m}} \tag{88}$$

i.e.

$$\begin{pmatrix} & \vdots & \\ \ldots & \sum_{k=-n}^{n} s_k \; (k \, \Delta x)^j \, (k \, \Delta x)^m & \ldots \\ & \vdots & \end{pmatrix} \begin{pmatrix} \vdots \\ \beta_{\nu,j} \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \sum_{k=-n}^{n} s_k \, (k \, \Delta x)^m \, z_{\nu+k} \\ \vdots \end{pmatrix} \tag{89}$$

$$w_\nu = (1 \; 0 \; \ldots \; 0) \begin{pmatrix} & \vdots & \\ \ldots & \sum_{k=-n}^{n} s_k \, (k \, \Delta x)^j \, (k \, \Delta x)^m & \ldots \\ & \vdots & \end{pmatrix}^{-1} \begin{pmatrix} \vdots \\ \sum_{k=-n}^{n} s_k \, (k \, \Delta x)^m \, z_{\nu+k} \\ \vdots \end{pmatrix} \tag{90}$$

In ISO 16610-28 it is used to minimize end effects by preserving the mean, i.e. the first moment with $p = 1$. In ISO 16610-31 it is used for shape retainment, preserving the width of peaks, i.e. second moment with $p = 2$.