Smoothing of well rates in subsurface hydrocarbon reservoir simulators

Ognyan Kounchev, Michail Todorov, Daniela Georgieva,
Nikola Simeonov, Vassil Kolev

1. Introduction

A common problem in reservoir simulators is the history matching problem, where a number of wells are operated at a prescribed flow rate, measured by the operator. The data provides input to a simulator which then has to match various other measured quantities, such as pressure drop at wells, movement of saturation fronts, water break-out and other. A common problem is that the input data is very rough and if input directly would cause considerable numerical difficulties, such as excessive Newton iterations to converge or excessively small time-steps.

2. Posing the Problem

A typical input for a well is a flow rate, specified at discrete time instances, which is positive at every instance. The goal is to replace the "rough" flow rate with a smoother function, which retains two properties of the original:

- It remains positive at every instance;
- The integral over the entire time range is preserved.

Different smoothing scenarios are expected to be seen.

2.1. First Scenario: Approximation by Smoothing Splines and Newton-Raphson Method

Replace the data function $f(t)$ by a smoothing spline $S_f \in C^2$ with restrictions

$$\int_0^T f(t)dt = \int_0^T S_f dt \qquad \text{and} \qquad S_f > 0$$

for $0 \leq t \leq T$.

Let us assume that the data sites $\{t_j\}_{j=1}^N$ with $t_1 < t_2 < \cdots < t_N$ are given with some data $f_j \geq 0$, which are assumed to be the values of a function $f(t)$, i.e.

$$f(t_j) = f_j \qquad \text{for } j = 1, 2, \ldots, N.$$

The problem is to "smoothen" those data f_j, which represent a very abrupt jump, by finding a function $g(t)$, for which the following conditions hold

$$\int\limits_{t_1}^{t_N} L_f(t)\, dt = \int\limits_{t_1}^{t_N} g(t)\, dt$$

$$g(t) \geq 0 \qquad \text{for } t_1 \leq t \leq t_N;$$

here the function $L_f(t)$ is the linear interpolating spline, which satisfies

$$L_f(t_j) = f_j \qquad \text{for } j = 1, 2, \ldots, N.$$

For solving this problem we propose to use approximation (smoothing) cubic splines $S_f(t)$, which by definition belong to $C^2(t_1, t_N)$ [1, 2] (i.e., have two continuous derivatives), having a parameter λ, which provides a trade off between the "goodness of interpolation to the data f_j" and coarseness of the graph of the spline function $S_f(t)$. Such a spline S_f is defined as the **unique solution** of the following problem:

$$\min_{S_f} \left(\lambda \sum_{j=1}^{N} w_j \left(S(t_j) - f_j\right)^2 + (1-\lambda) \int\limits_{t_1}^{t_N} \varphi(t) \left|S''(t)\right|^2 dt \right). \tag{1}$$

Here the parameter λ, the so-called smoothing parameter is given. We assume also: given weights $w_j \geq 0$, which show how good we wish to have the size of $|S(t_j) - f_j|$ for every j, and also the function $\varphi(t) \geq 0$ in the interval $[t_1, t_N]$, which shows the "roughness" of the function $S_f(t)$ at every particular point t. We will not use this large freedom but we will choose $\varphi(t) = 1$.

However we will use essentially the weights w_j in order to satisfy the condition

$$C := \int\limits_{t_1}^{t_N} S_f(t)\, dt = \int\limits_{t_1}^{t_N} L_f(t)\, dt. \tag{2}$$

2.2. Second Scenario: Replace the initial piece-wise linear "rough" curve by another piece-wise linear less "rough" curve

The new curve must be subject to the above restrictions. This is possible to implement and to get a simple explicit relationship to some new average value (see Figure 1):

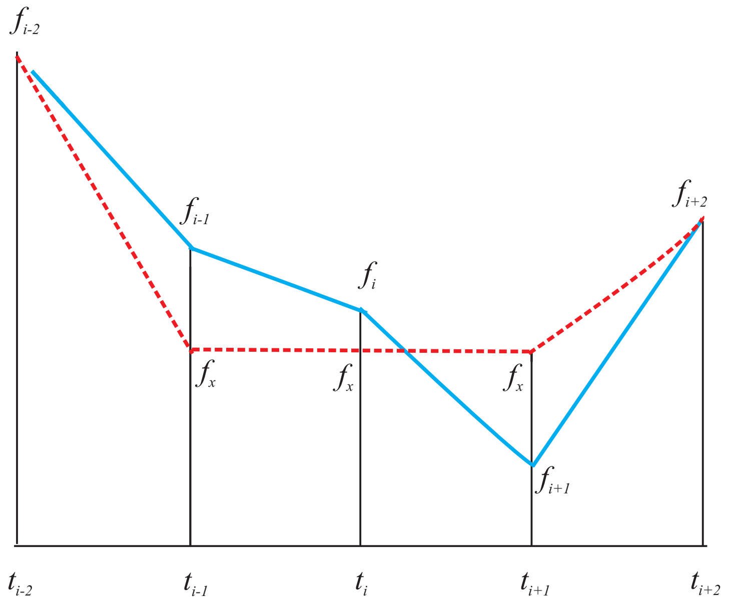

Figure 1: Graph sketch of the moving average. The areas under the solid line (four trapezoidals) and dashed line (two lateral trapezoidals and two congruent rectangles) are equal.

$$f_x = \frac{f_{i-1}(h_{i-1}+h_i)+f_i(h_i+h_{i+1})+f_{i+1}(h_{i+1}+h_{i+2})}{h_{i-1}+2(h_i+h_{i+1})+h_{i+2}} \tag{3}$$

where $h_i = t_i - t_{i-1}$, $i = 1, N$ procedure for $i = 1, \ldots, N$ by step 4. Let us note that it is possible to divide to groups of more trapezoidals but then one can lose the general trend of the original empirical curve. Also, obviously $f_x > 0$ for any empirical data. Applying this procedure one cuts the largest deviations of the measurements.

2.3. Other Scenarios: Approximation by Discrete Wavelet Transform

One can approximate any piecewise continuous function by using a pair of orthogonal bases containing scaling and wavelet functions $\varphi(t)$ and $\psi(t)$. The scaling function $\varphi(t) = \sum\limits_n \sqrt{2}h(n)\varphi(2t-n)$, where $h(n)$ is the lowpass filter coefficient estimated with $h(n) = \frac{1}{\sqrt{2}}\langle\varphi(t/2), \varphi(t-k)\rangle$, while the wavelet function $\psi(t) = \sum\limits_n \sqrt{2}g(n)\psi(2t-n)$, where $g(n)$ is the highpass filter coefficient. Both filter coefficients are coupled with the explicit relationship $g(n) = (-1)^n h(1-n)$ (see, for example [3] and [4]). The wavelets expand the signals into separate frequency components, and then one can study each component with a resolution matched to its scale. The discrete wavelet transform (DWT) is a special case of wavelet transform that provides a compact representation of a signal in time and frequency that can be computed efficiently. The decomposition of a given function $f(t)$ for j-level by the above basis functions is:

$$f(t) = \sum_n h(n)\varphi(t-n) + \sum_j \sum_n g_{2^j+n}(n)\psi(2^j t - n) = f_a(t) + \sum_j f_{d_j}(t). \quad (4)$$

The first term $f_a(t)$ is the approximation function, while the second term is a sum of the so-called detail functions. An example of decomposition for level $j = 3$ with the orthogonal 10-taps Daubechies wavelet is shown on Figure 2.

Two approaches are possible to be considered:

a) For *uniform mesh (UM)* on the time segment $[0, T]$ the integral of the function $f(t)$ can be approximated for a given j-level by the discrete wavelet decomposition $f_a(t)$ with error ε_0:

$$\int\limits_0^T f(t) = \int\limits_0^T f_a(t) + \varepsilon_0;$$

b) For *non-uniform mesh (NUM)* on the time segment $[0, T]$ the integral of the function $f(t)$ can be approximated also with $f_a(t)$ but with another error ε_1:

$$\int\limits_0^T f(t) = \int\limits_0^T f_a(t) + \varepsilon_1.$$

Simpson's integration rule [5] for smooth functions is preferable compared to the trapezoidal rule [5, 6] since the error is roughly proportional to 10^{-4} and it does not require a dense mesh to attain a priori desired accuracy. Although the trapezoidal rule is inefficient in general, it can be shockingly efficient for very jagged and periodic functions fast approaching zero. This simplest numerical integration technique can be extraordinarily efficient when it is skilfully applied for getting reliable approximations of empirical data and relationships.

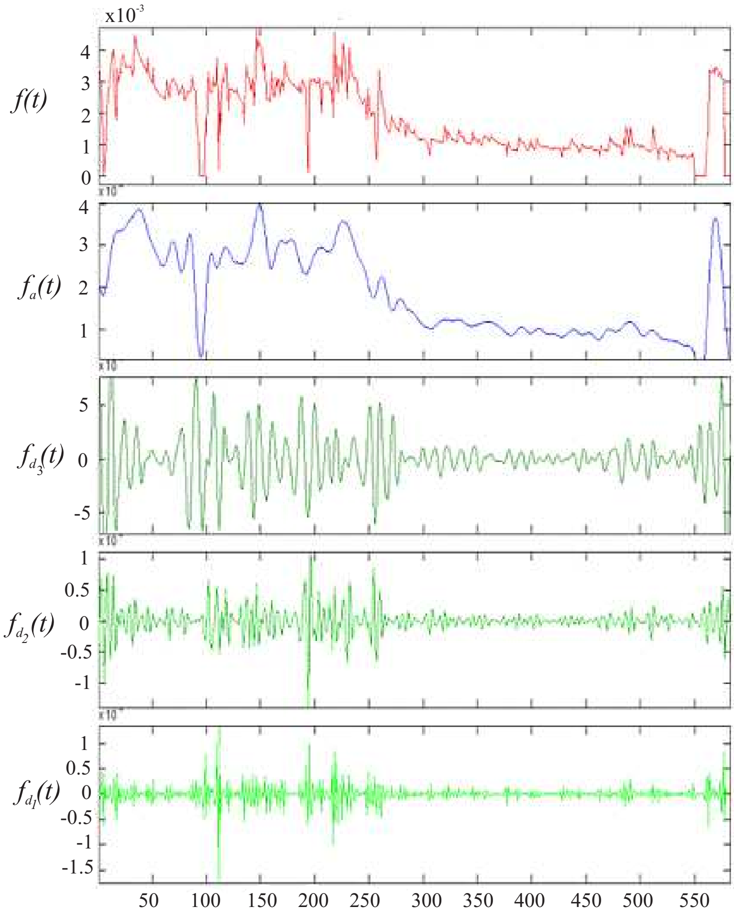

Figure 2: DWT of the customer function for level $j = 3$ with the orthogonal 'db10'

3. Numerical Results

In the particular set of data we have $N = 585$ and the area under the empirical curve is given by the integral

$$C = \int_{t_1}^{t_N} S_f\,(t)\,dt = 1.316303598725274.$$

For solving the system (1) and (2) we need one more free parameter: we

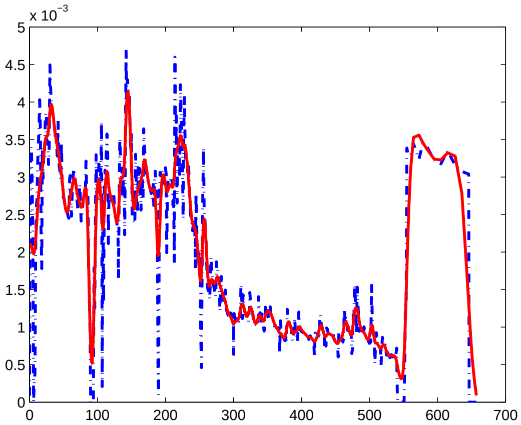

Figure 3: First scenario: original empirical data (dashed line), approximating curve with restriction for smoothing parameter $\lambda = 0.08$ (solid line).

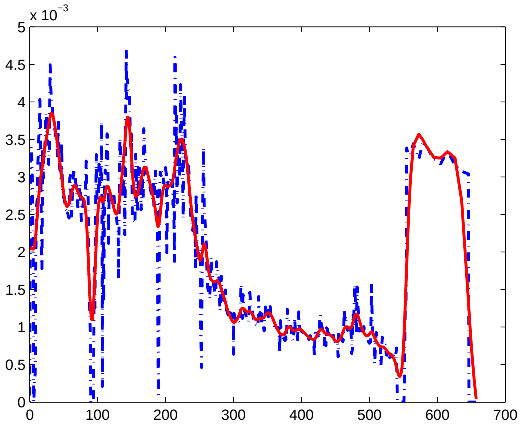

Figure 4: First scenario: original empirical data (dashed line), approximating curve with restriction for smoothing parameter $\lambda = 0.01$ (solid line).

consider the unknown weights (with unknown parameter x):

$$\begin{aligned} &w_1 = \cdots = w_{10} = w_{576} = \cdots = w_{585} = x \\ &w_j = 1 \qquad \text{for } 11 \leq j \leq 575 \end{aligned}$$

and we solve the system by using a Newton-Raphson solver.

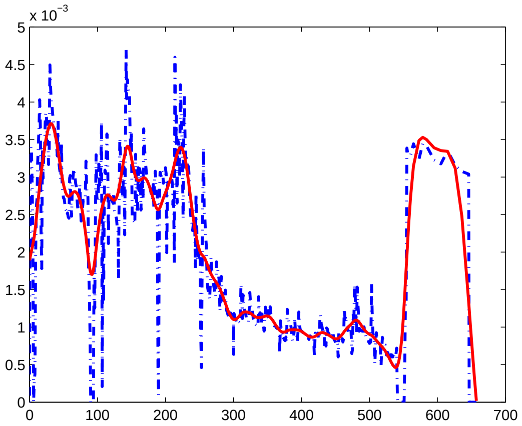

Figure 5: First scenario: original empirical data (dashed line), approximating curve with restriction for smoothing parameter $\lambda = 0.001$ (solid line).

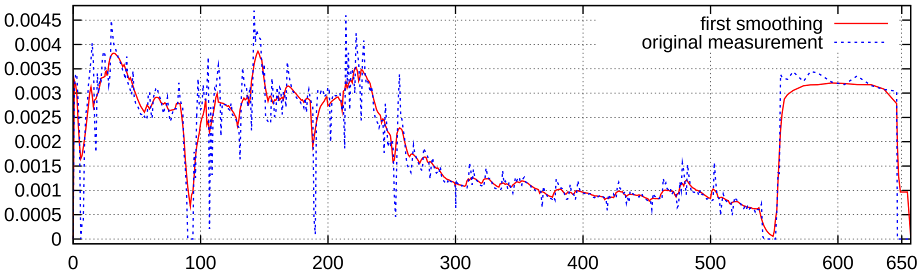

Figure 6: Second scenario: original empirical data (dashed line), piece-wise linear curve with restriction from Figure 1 (solid line).

For different values of the parameter λ we obtain solutions $S_f\left(t\right)$, which satisfy condition (2) with precision 10^{-16}.

To demonstrate how does the procedure work we provide some experimental results with different values of the smoothing parameter λ. It is clearly seen on Figures 3, 4, 5 that the smaller values of the parameter λ smooth more the spline $S_f\left(t\right)$. Also, condition $S_f\left(t\right) \geq 0$ is satisfied. The latter is attained by manipulation of the weights w_j and the function $\varphi\left(t\right)$.

For the first scenario – smoothing splines with restrictions and sequential Newton-Raphson technique we get the results: $OldArea = 1.316303598725274$ and $NewArea = 1.316303598725274$. They are practically identical because the error reaches the machine epsilon, i.e., $Err \sim 10^{-17}$

For the second scenario based on the explicit formula (3) – replacement of one piece-wise linear curve by another one the numerical results cover fully the prediction given by the first scenario of smoothing: $OldArea = 1.31630359872527$, $NewArea = 1.31630359872527$, with $Er \sim 10^{-17}$ (Figure 6).

Since the customer data form a jagged function with fast approaching zero parts the trapezoidal rule for integration in the third scenario is used. The calculated integral value is

$$\int\limits_0^T f(t) = 1.311518426300291 \quad \text{with} \quad T = 585.$$

The minimal-approximation absolute errors for *UM* and *NUM* are tabulated in Tables I and II. Obviously, the accuracy of the approximating integrals depend on the uniform mesh and the levels of DWT. From the level decompositions for *UM* when $j = 1, \ldots, 5$ and *NUM* of $j = 1, 2, 3$ we conclude that the increase of j-level leads to both a decrease of the accuracy, and an increase of the approximation errors ε_0 and ε_1 (see Figures 7 and 8). The higher levels of DWT, however, provide smoother functions, which is the customer preference. The magnitudes of errors of the both methods vary as follows: $\varepsilon_0 \in (10^{-6}, 10^{-4})$, $\varepsilon_1 \in (10^{-4}, 10^{-2})$.

Table I: The minimal approximation errors for *UM*

level	wavelet	$\lvert\varepsilon_0\rvert * 10^{-4}$
1	'sym2'	0.025
2	'db15'	0.890
3	'db2'	3.941
4	'bior3.1'	0.243
5	'db41'	1.105

Table II: The minimal approximation errors for *NUM*

level	wavelet	$\lvert\varepsilon_1\rvert * 10^{-4}$
1	'db42'	0.4668
2	'sym9'	6.9017
3	'db42'	35.44

Conclusion

Which scenario to choose? Actually the developed scenarios are equivalent as a prediction and an order of approximation of the quadratures. Their advantage consists in the fast computer realization provided the output results as input data for the further computer processing and simulations. These procedures are not unique – they can be varied depending on the needs of the user.

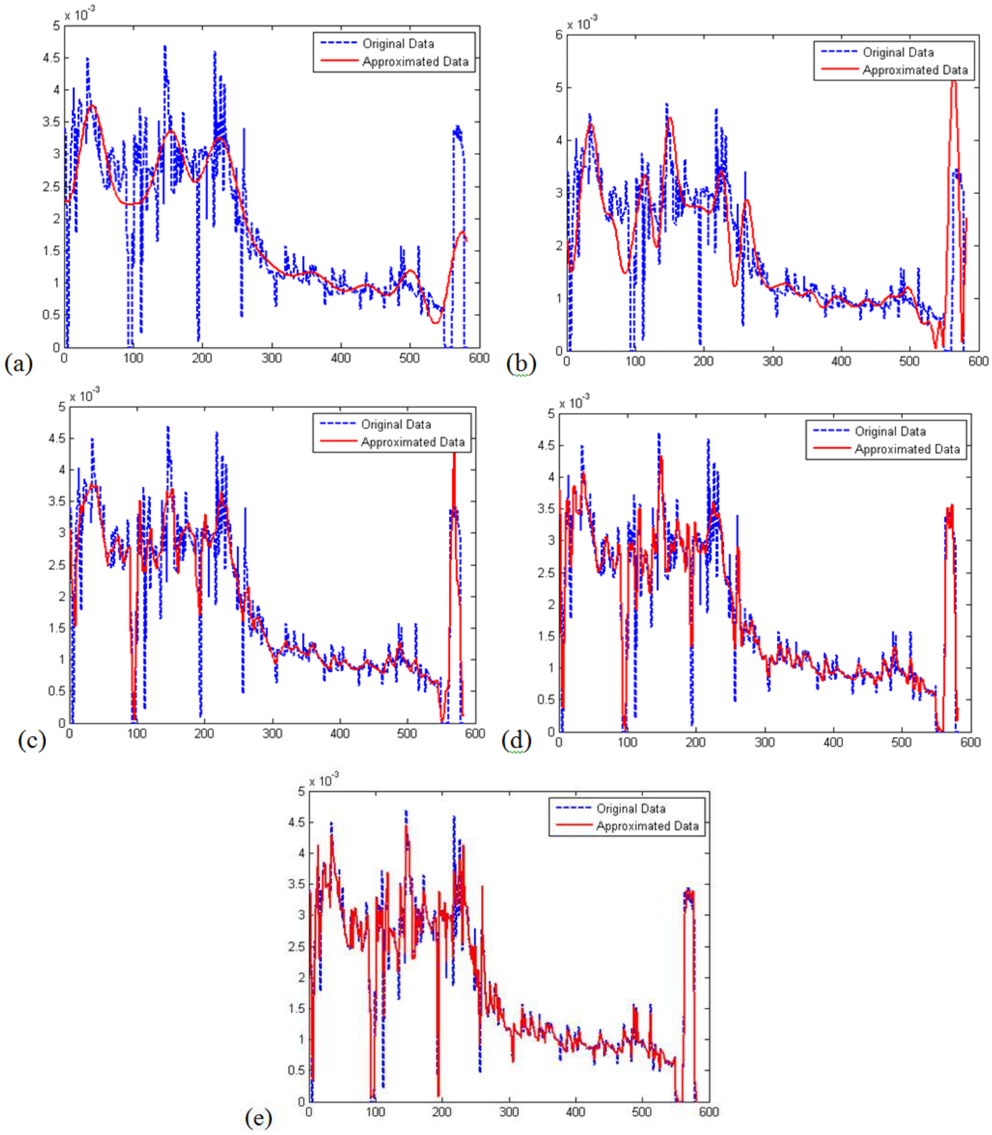

Figure 7: The approximation function for UM with ε_0 for: (a) $j = 5$ with the orthogonal wavelet 'db41'; (b) $j = 4$ with the biorthogonal wavelet 'bior3.1'; (c) $j = 3$ with the orthogonal wavelet 'db2'; (d) $j = 2$ with the orthogonal wavelet 'db15'; (e) $j = 2$ with the symmlet 'sym15'. Empirical data (dashed line), approximation function (solid line)

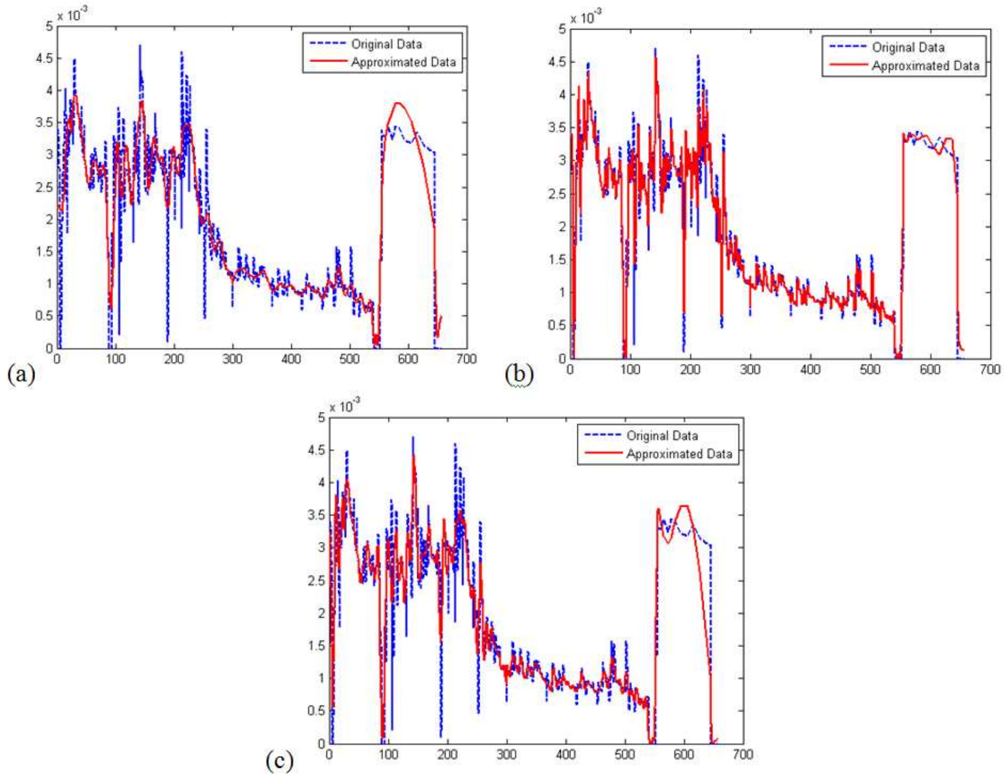

Figure 8: The approximation function for NUM with ε_1 for: (a) $j = 3$ with the orthogonal wavelet 'db42'; (b) $j = 2$ with the symmlet 'sym9'; $j = 1$ with the orthogonal wavelet 'db42'. Empirical data (dashed line), approximation function (solid line)

References

[1] C. De Boor, *A Practical Guide to Splines*, Revised Edn., Springer, 2001, pp. 207–214.

[2] R. E. Smith Jr., J. M. Price and L. M. Howser, "A Smoothing Algorithm Using Cubic Spline Functions", NASA TN D-7397, Feb. 1974, Retrieved May 2011.

[3] K. Shukla, A. Tiwari, Efficient Algorithms for Discrete Wavelet Transform With Applications to Denoising and Fuzzy Inference Systems, Springer Briefs in Computer Science, 2013.

[4] O. Christensen and K. Christensen, Approximation Theory: From Taylor Polynomials to Wavelets, Birkhauser Verlag AG, 2004.

[5] J. Epperson, An Introduction to Numerical Methods and Analysis, Willey, 2007.

[6] A. Gilat and V. Subramaniam, Numerical Methods for Engineers and Scientists: An Introduction with Applications using MATLAB, 3rd edn., Willey, 2013.