Computing Derivatives of Noisy Signals Using Orthogonal Functions Expansions.

Adi Ditkowski [*†],
Abhinav Bhandari[‡], Brian W. Sheldon[§]

16 Nov. 2008

Abstract

In many applications noisy signals are measured. These signals has to be filtered and, sometimes, their derivative have to be computed.

In this paper a method for filtering the signals and computing the derivatives is presented. This method is based on expansion onto transformed Legender polynomials.

Numerical examples demonstrates the efficacy of the method as well as the theoretical estimates.

1 Introduction

In many applications, a noisy signals are sampled, and have to be processed. Few of the things that have to be done are filtering the noise and computing the derivatives of the signal.

An example for this application is the evolution of stress during the growth of films. The evolution of stress during the growth of patterned, electrodeposited Ni films was measured using a multibeam opitical stress sensor (MOSS) [1]. In this technique, the space between adjacent laser spots is used to monitor bending of the substrate-film system. This spot spacing is directly related to the curvature, which in turn provides a precise in situ measurement of the force exerted on the top of the substrate, due to the stress in the growing film. It is traditional to report this type of data as the product of stress and thickness (i.e., force), as shown in Figure 1. In materials where the film stress does not change appreciably

[*]This research was supported by the ISRAEL SCIENCE FOUNDATION (grant No. 1364/04).
[†]School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel
[‡]Department of Engineering, Brown University, Providence RI, 02912
[§]Department of Engineering, Brown University, Providence RI, 02912

after it is deposited, the derivative of this data gives the so-called instantaneous stress at the top of the film (after dividing the derivative by the growth rate of the film). A plot of instantaneous stress for a patterned Ni film is show in Figure 2(b). Of particular interest here is the peak in the stress. The physical cause of this peak is unclear, and identifying the responsible mechanism(s) is an active research interest [1]. To accomplish this, it is essential that accurate values be obtained for both the maximum stress and the time where this peak occurs.

In this particular example, 150832, not equally spaced, measurements were taken. a plot of this signal is given in Figure 1.

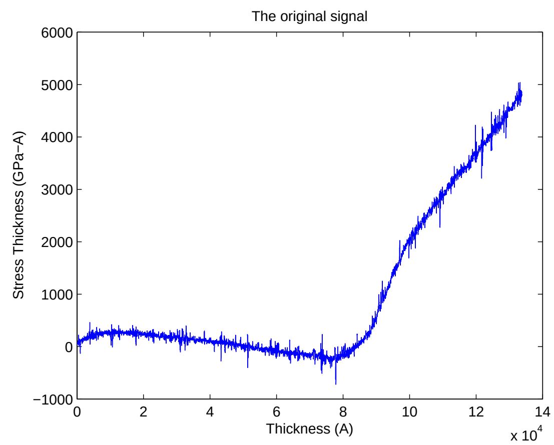


Figure 1: The measured signal.

Here, and in the rest of the paper, the term signal is defined as the measured data. The 'clean signal' is the pure physical phenomena, which would have been measured with an infinite, or at least machine, precision and the noise in the difference between these two.

Attempt to use standard finite difference scheme in order to compute the derivative of the signal fails, since the magnitude of the derivative is much larger then the one of the clean signal. This phenomenon can be clearly illustrated in the following example. Let,

$$f(t) = u(t) + \epsilon \sin(\omega t) \ . \tag{1.1}$$

Here $f(t)$ represents the sampled signal, $u(t)$, represents the 'clean' signal and $\epsilon \sin(\omega t)$, the noise. It is assumed that the derivatives of $u(t)$ are of order 1, $\epsilon \ll 1$ and $\omega \gg 1$. The derivative of (1.1) is

$$f'(t) = u'(t) + \epsilon\omega \cos(\omega t) \ . \tag{1.2}$$

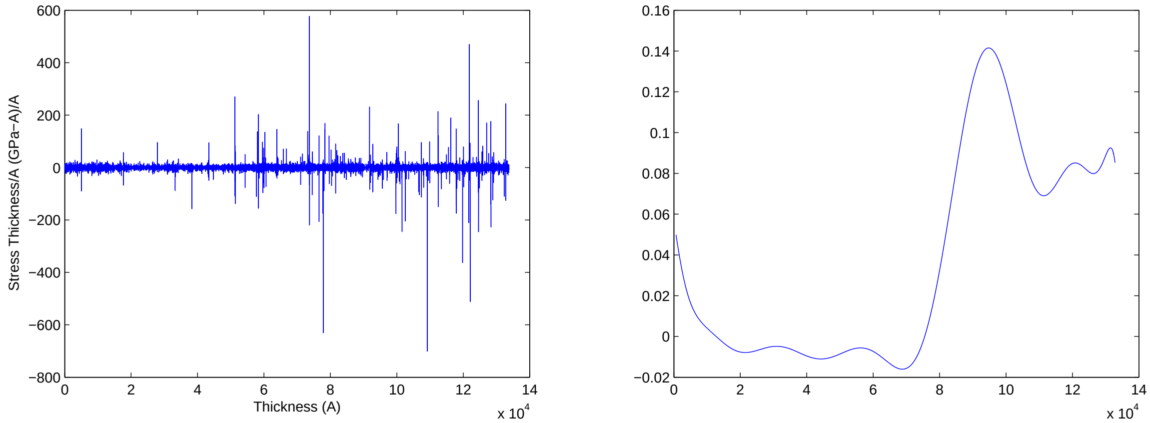


Figure 2: Left, (a): the derivative of the signal presented in figure 1. computed by finite difference method. Right, (b): the derivative of the same computed by modified Legendre polynomial expansion.

If $\epsilon\,\omega \gg 1$ then $u'(t)$ is just a small perturbation to the derivative of the noise. In Figure 2. the derivative of the signal plotted in Figure 1. computed by standard finite difference and by the modified Legendre polynomial expansion method, presented below, are shown.

It is not clear how to distinguish between the noise and the 'clean signal'. Here, we shall make the following qualitative assumptions:

1. The 'clean' signal is a slowly varying C^∞ function.
2. The noise is either concentrate at 'high modes' or at least evenly spreads over all modes, s.a. 'white noise'.
3. The signal is sampled using large number of points, ξ_j, $j = 0 \ldots N$, with respect to the 'clean' signal and $\max h_j = \max(\xi_{j+1} - \xi_j) \ll \xi_N - \xi_0$.

The terms 'modes' are the eigenfunctions in the expansion by orthogonal functions.

The goal of this paper is to present a simple, fairly fast and easy to implement, algorithm (less then 100 lines of MATLAB code) to filter the noise and compute an approximation to the first derivative.

There are several approaches for filtering and computing the derivatives of a noisy signals; some of them are using PDE, approximating by global polynomials using least-squares, moving least-squares and projection onto an orthogonal subspace.

Diffusion equations are widely used for denoising. The idea behind this approach is that high frequencies are rapidly attenuated by the diffusion process. In order to control the attenuation of high frequencies, without loosing important feature of the signal, or picture in image processing context, sophisticated, often nonlinear diffusion PDE are tailored to the specific problem. For more information, see, for example [2]. some of the difficulties with this approach is that we are bounded by the dynamics of the PDE, and, since this dynamics has to be followed with a reasonable accuracy, the time marching is slow, even when implicit methods are used.

Approximating the signal by global polynomials using least-squares is widely used for low order polynomials, especially, linear fitting of data. The standard least-squares fitting to higher order polynomials process, for example by using the MATLAB command `polyfit`, is unstable. There are two solutions for this situation, either to approximate the signal by low order polynomials, locally, for example by using the moving least-squares method, or using global orthogonal approximations.

The moving least-squares is a method of reconstructing continuous functions from a set of unorganized point samples via the calculation of a weighted least-squares measure biased towards the region around the point at which the reconstructed value is requested, see [10]. Typically, a gaussian is used as a weight function. This is, conceptually, a simple method to implement, however, as will be demonstrated in section 3, it is inaccurate and slow.

In the projection onto orthogonal subspace, the signal is projected onto a set of orthogonal functions, or modes. The 'clean signal' is then approximated by a linear combination of few of these modes. This method will be discussed in details in section 2. In particular we propose to use a modified, or mapped orthogonal polynomials, rather then the standard orthogonal polynomials. This bases approximates the derivative with higher accuracy. In principle, it is also possible to approximate higher derivatives, however, the accuracy is much lower.

It should be noted that there is an overlap between the methods. For example; using the standard heat equation, with Neumann boundary condition in the domain $x \in [0, \pi]$ for time t, is exactly as using expansion into $\cos(n\,x)$ function and multiply each mode by $\exp(-n^2 t)$.

In section 2. we present the way to utilize expansion by orthogonal polynomials, and their modifications, for this task. Numerical example will be presented in Section 3. A simplified algorithm for applying the method as well as implementation remarks are presented in section 4.

2 Projection onto orthogonal functions.

In this section we shall describe different possibilities of using projections onto orthogonal functions.

The idea behind this approach is, according to assumption 2, the noise spectrum is 'evenly' spreads over all modes, while the spectrum of the 'clean signal' is concentrated in the lower modes. With a proper choice of function base, only few modes are required to accurately approximate the 'clean signal'. The expected error in the coefficients is small since the noise is evenly distributed over a large number of modes. Qualitatively; let us denote the noise by U_{noise}, and its expansion into a set of N orthogonal modes $\phi_j,\ j = 1 \ldots N$

$$U_{noise} = \sum_{j=1}^{N} c_j \, \phi_j \quad , \tag{2.1}$$

where

$$c_j = \langle \phi_j, U_{noise} \rangle \quad . \tag{2.2}$$

The inner product $\langle \cdot, \cdot \rangle$ is such that

$$\langle \phi_j, \phi_k \rangle = \delta_{j,k} \quad . \tag{2.3}$$

The energy of the noise is

$$E_{noise} = \langle U_{noise}, U_{noise} \rangle \tag{2.4}$$

and using the Parseval equality

$$E_{noise} = \sum_{j=1}^{N} |c_j|^2 \ . \tag{2.5}$$

Since, according to assumption 2, the noise is evenly spread over all modes, the 'average' magnitude $|c_j|$ is

$$|c_j| \approx \sqrt{\frac{E_{noise}}{N}} \ . \tag{2.6}$$

If the clean signal u can be accurately approximated by K modes, i.e.

$$||u - \sum_{j=1}^{K} a_j \, \phi_j|| < \epsilon \ , \tag{2.7}$$

Then the error in the modes a_j is $c_j \approx \sqrt{\frac{E_n}{N}}$. The conclusion from (2.6) is that as many modes as possible should be used for the approximation, and the decay rate of the error in the first modes is only of order $N^{1/2}$. On the other hand K should be kept as small, i.e. as fast convergence as possible.

This approach is closely related to spectral methods, see for example [3], [4] and [6]. There are, however, few fundamental differences.

1. The data is given at specific points, ξ_j, $j = 0 \ldots N$, not at some chosen collocation points.

2. The noise is high; the noise to signal ratio can be as large as 10%.

3. There is a large number of data points.

As a result the standard practice used in spectral methods cannot be applied here. However, since there is a large number of points and the expected accuracy is low, low order integration methods, such as the trapezoid rule, could be used.

The, the basic algorithm for recovering the 'clean signal' is:

1. Select the function base.

2. Expand the signal in this function base.

3. Truncate the series, and use it as the approximation for the 'clean signal'.

The number of terms which should be taken is about the number of modes with the coefficients which are significantly larger then the noise. The rest of the modes are truncated, either directly, or by using an exponential filter. The assumption that the spectrum of the noise is 'evenly' spread over all modes, may not hold for all applications. In this case, the point of truncation can be determined by a plot of the amplitudes of the modes, as in Figure 6.

In order to evaluate the different methods we shall look at the following synthetic example. Let the 'clean signal', u be

$$\begin{aligned} u \;=\; & \frac{63}{256} + \frac{63x^2}{512} + \frac{189x}{512} - \frac{105\cos(\pi x)}{256\pi^2} - \frac{15\cos(2\pi x)}{256\pi^2} - \frac{5\cos(3\pi x)}{512\pi^2} \\ & - \frac{5\cos(4\pi x)}{4096\pi^2} - \frac{\cos(5\pi x)}{12800\pi^2} + \frac{105\sin(\pi x)}{512\pi} + \frac{15\sin(2\pi x)}{256\pi} \\ & + \frac{15\sin(3\pi x)}{1024\pi} + \frac{5\sin(4\pi x)}{2048\pi} + \frac{\sin(5\pi x)}{5120\pi} - \frac{36883}{102400\pi^2} \end{aligned} \tag{2.8}$$

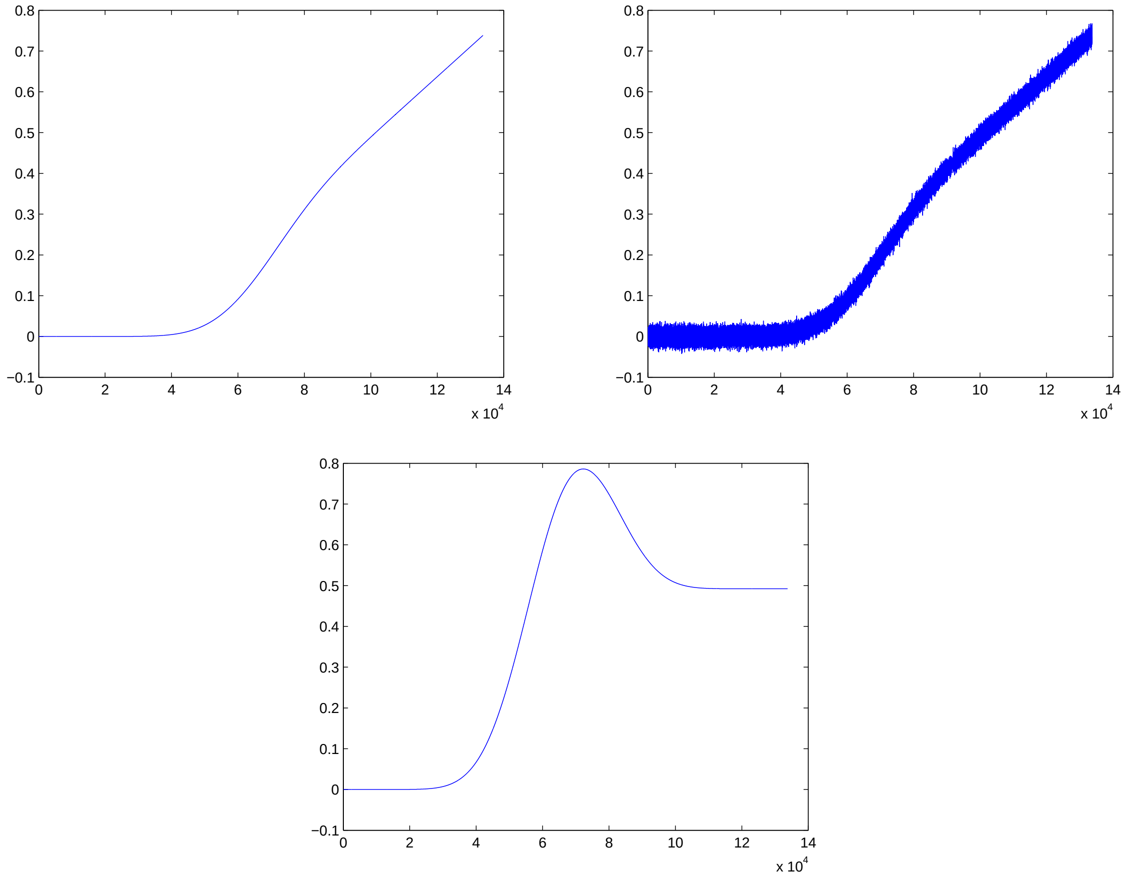


Figure 3: Top left, (a): the clean signal. Top right, (b): the noisy signal. Bottom, (c): the derivative of the clean signal.

where

$$x = 2\frac{\xi - \xi_0}{\xi_n - \xi_0} - 1 \tag{2.9}$$

The noise that was added is a pseudorandom number with a normal distribution with mean 0 and standard deviation 1/100, at each sampling point. The noise was generated by the MATLAB command `(1/100)*randn`. The sampling points, ξ_j, were taken from the signal presented in Figure 1. The plot of this test signal is presented in Figure 3.

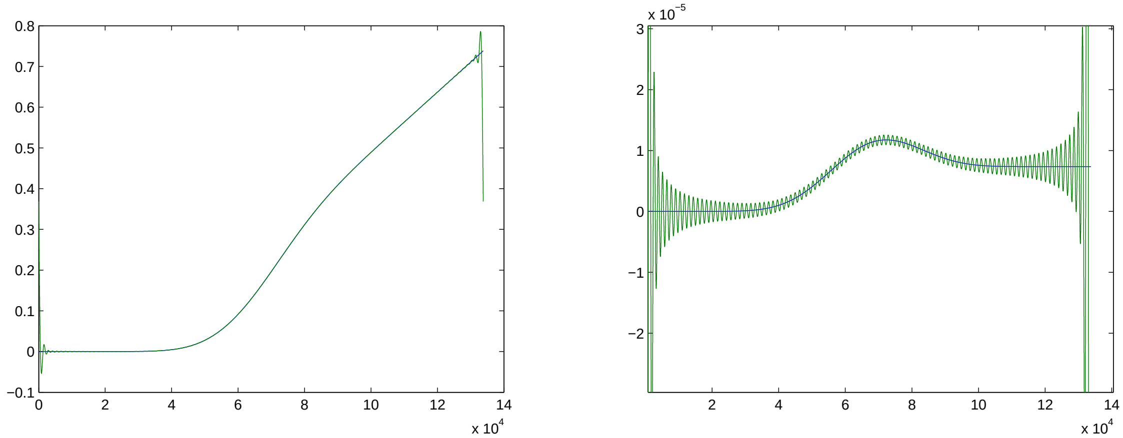


Figure 4: Expansion using trigonometric polynomial e^{ijx}. Left, (a): the approximation of the clean signal. Right, (b): the approximation of the clean signal derivative.

2.1 Projection onto trigonometric polynomials.

Expending in trigonometric polynomials,

$$f(x) = \sum_{j=-N/2}^{N/2} a_j \, \frac{1}{\sqrt{2\,\pi}} e^{ijx} \tag{2.10}$$

is widespread and highly efficient. It is also well known that if $f^{(k)}(x),\ k = 0 \ldots p-1$ is 2π-periodic and $f^{(p)}(x)$ is a piecewise C^1-function. Then

$$|a_j| \leq \frac{\text{constant}}{|j|^{p+1} + 1}\ . \tag{2.11}$$

See, for example [4]. If $f(x)$ and all it's derivative were periodic, then $|a_j|$ would have decay exponentially. However, in our case, $f(x)$ is not periodic, thus, there is no L_∞ convergence of the series, and the series cannot be differentiated, term by term. In Figure 4. the signal were recovered using about 180 modes. The Gibbs phenomenon is clearly shown and large oscillations contaminated the whole domain.

Better approximations can be achieved using cosine transform. Since the boundary condition for this Sturm-Liouville problem is the vanishing of the derivatives in both ends, there is a pointwise convergence. As can be seen in Figure 5. there is a good approximation of the signal. However the approximation of the derivative near the boundary breaks down as the periodic extension of the derivative is not continuous.

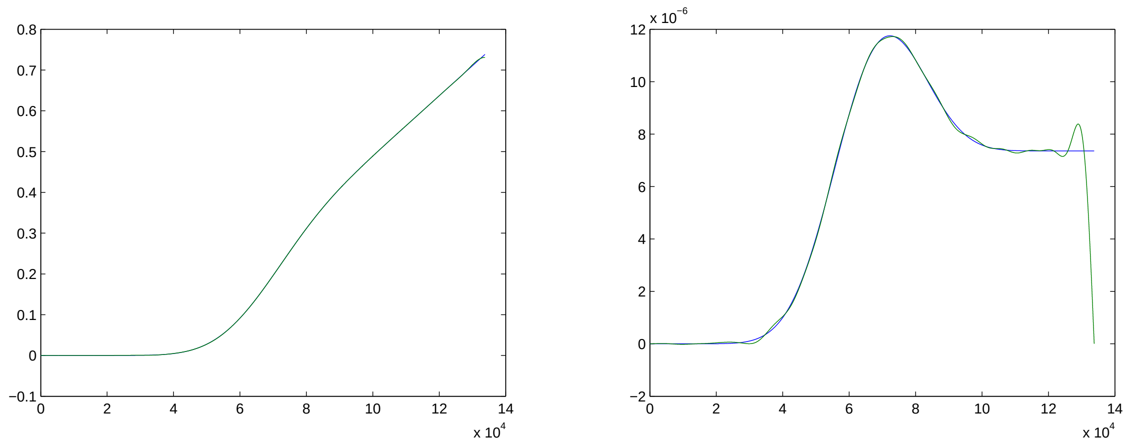


Figure 5: Expansion using $\cos(j\,x)$. Left, (a): the approximation of the clean signal. Right, (b): the approximation of the clean signal derivative.

Though highly sophisticated methods had been developed in order to eliminate the Gibbs phenomenon by Gottlieb and Shu, see for example [4] and [5], a simpler approach is to avoid this situation altogether.

2.2 Projection onto Legendre polynomials.

Expanding of C^∞ functions in orthogonal polynomials, which are the eigenfunctions of a singular Sturm-Liouville problem, is highly efficient, as the coefficients decay exponentially. Orthogonal polynomials are widely used in spectral methods, see for example [4] and [6]. Here we had chosen to use Legendre, instead of Chebyshev polynomials, since the weight function for the Chebyshev polynomials, $1/\sqrt{1-x^2}$ is singular near $x = \pm 1$. In the common practice of spectral methods, it doesn't cause any difficulty, since the integration, with the weight function, is implemented using a proper distribution of the collocation nodes. Here, we do not have the freedom to choose our computational point, as explained in the beginning of this section, and the integration, with a singular weight function over a dense grid generates large errors.

As can be seen from Figure 6, the first 10-15 coefficients indeed decay exponentially, the amplitude of the rest of the modes is, more or less, constant. Therefore, the recovered signal is constructed from these first modes. The 'noise level' is assumed to be the amplitude of the rest of the modes. Since it is also assumed the same level of noise exists also in the first ten modes, we estimate the error by the 'average' amplitude of the rest of the modes, see

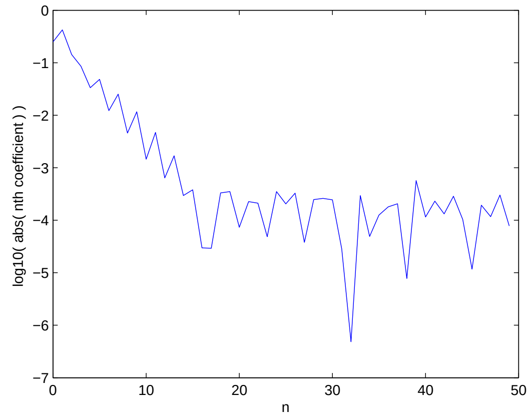


Figure 6: Expansion using Legendre polynomials; $\log_{10}$ of the absolute value of the amplitude of the modes.

(2.6).

The results here are significantly better then the previous cases, however, as can be seen in 7c and d the error in the derivative near the edges (-1 and 1 in the transformed variable x) is still large. The reason for it is that the distance between zeroes of nth Legendre polynomials, P_n, are $O(1/n^2)$, near ± 1. This estimate can be derived form a general property of Jacobi polynomials with $-1/2 \leq \alpha, \beta \leq 1/2$. Let us denote by $\xi_k^{(n)}$ the kth zero of the nth Jacobi polynomial, then

$$1 \leq -\cos \frac{k + (\alpha + \beta - 1)/2}{n + (\alpha + \beta + 1)/2}\pi \leq \xi_k^{(n)} \leq -\cos \frac{k}{n + (\alpha + \beta + 1)/2}\pi \leq 1 \ . \tag{2.12}$$

In particular, for the Legendre polynomials, where $\alpha = \beta = 0$:

$$1 \leq -\cos \frac{k - 1/2}{n + 1/2}\pi \leq \xi_k^{(n)} \leq -\cos \frac{k}{n + 1/2}\pi \leq 1 \ . \tag{2.13}$$

See [7] for proofs and further results.

Therefore the derivative near ± 1 are of the order of n^2. Thus the error in the amplitude of the mode is multiplied by the same factor. The conclusion from this observation is that is order to reduce the error, the distance between zeroes should be of order n.

2.3 The proposed method: projection onto modified Legendre polynomials.

Using mapping for redistribution of the zeros of Chebyshev polynomials was introduced by Kosloff and Tal-Ezer in 1993, [8], and further investigated by Don and Solomonoff in 1995,

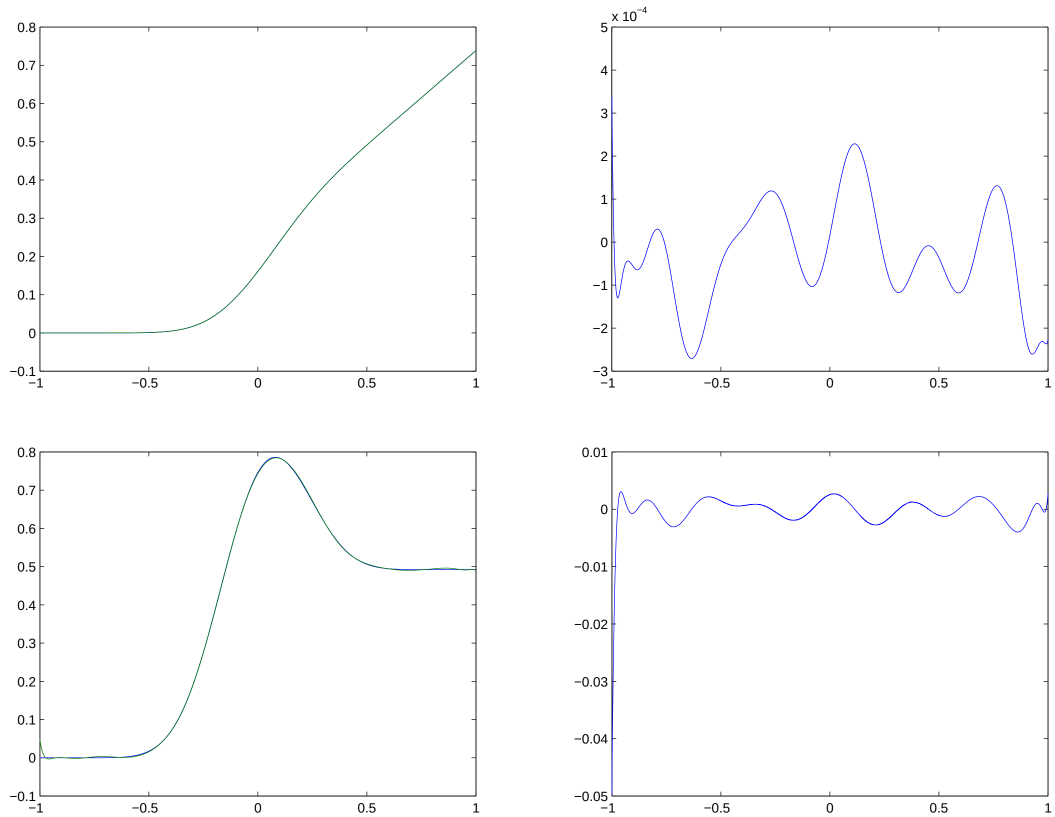


Figure 7: Expansion using Legendre polynomials. Top left: (a): the clean and the recovered signals. Top right, (b): the clean minus the recovered signals. Bottom left, (c): the derivatives of the clean and the recovered signals with respect to the transformed variable x. Bottom right, (d): the differences between the derivatives of the clean and the recovered signals with respect to x.

[9]. These transformations is useful for increasing the allowed time steps by resizing the eigenvalues of the differentiation operator, and to reduce the effects of roundoff errors. Here we use the function proposed by Kosloff and Tal-Ezer, namely

$$x = \frac{\arcsin(\alpha y)}{\arcsin(\alpha)} \quad 0 < \alpha < 1 \ . \tag{2.14}$$

Recalling that the Legendre differential equation is

$$\frac{d}{dx}\left((1-x^2)\frac{d}{dx}P_n(x) \right) + n(n+1)P_n(x) = 0 \ . \tag{2.15}$$

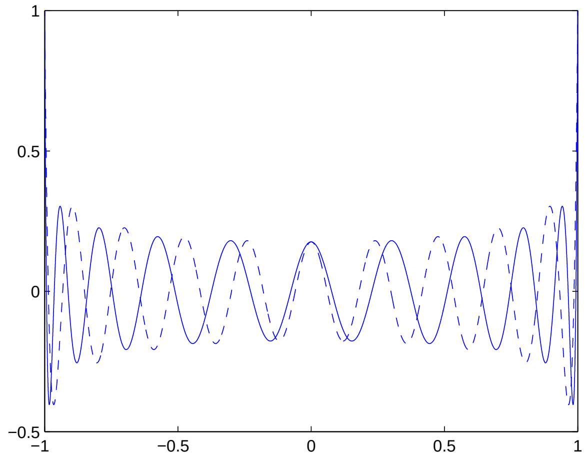


Figure 8: Demonstration of the mapping; Legendre polynomial P_{20}, solid, vs. modified, or thransformed polynomial, dashed.

After substituting the transformation (2.14) into (2.15) one gets

$$\frac{d}{dy}\left[\left(1-\left(\frac{\arcsin(\alpha y)}{\arcsin(\alpha)}\right)^2\right)\frac{\arcsin(\alpha)}{\alpha}\sqrt{1-y^2\alpha^2}\,\frac{d}{dy}\hat{P}_n(y)\right] + \frac{\alpha}{\arcsin(\alpha)\sqrt{1-y^2\alpha^2}}\,n(n+1)\hat{P}_n(y) = 0\,. \tag{2.16}$$

This is still a singular Sturm-Liouville problem, with a non-singular weight function

$$\frac{\alpha}{\arcsin(\alpha)\sqrt{1-y^2\alpha^2}} \tag{2.17}$$

and

$$\hat{P}_n(y) = P_n(x(y)) = P_n\left(\frac{\arcsin(\alpha y)}{\arcsin(\alpha)}\right)\,. \tag{2.18}$$

The orthogonality relation for the $\hat{P}_n(y)$ is:

$$\int_{-1}^{1}\hat{P}_n(y)\hat{P}_k(y)\frac{\alpha}{\arcsin(\alpha)\sqrt{1-y^2\alpha^2}}\,dy = \frac{2}{2n+1}\delta_{nk}\,. \tag{2.19}$$

Therefore, if

$$u(y) = \sum_n c_n\,\hat{P}_n(y)\,, \tag{2.20}$$

then

$$c_n = \frac{2n+1}{2}\int_{-1}^{1}\hat{P}_n(y)\,u(y)\,\frac{\alpha}{\arcsin(\alpha)\sqrt{1-y^2\alpha^2}}\,dy\,. \tag{2.21}$$

In order to find the coefficients c_n the integral (2.21) is evaluated numerically.

It is also worth noting that the $\hat{P}_n(y)$'s satisfy the recurrence formula

$$(n+1)\,\hat{P}_{(n+1)}(y) \;=\; (2n+1)\frac{\arcsin(\alpha y)}{\arcsin(\alpha)}\hat{P}_n(y) \;-\; n\,\hat{P}_{(n-1)}(y)\;. \tag{2.22}$$

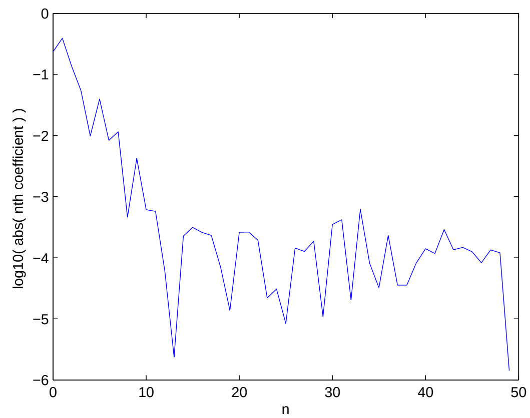


Figure 9: Expansion using modified Legendre polynomials. $\log_{10}$ of the absolute value of the amplitude of the modes.

Though this transformation was tailored for the Chebyshev polynomials it also does a stisfying eork here. The reason for it can be seen from (2.12) for Chebyshev polynomials, $\alpha = \beta = -1/2$, and from (2.13), that the distributions of the zeros are not much different, at least for large n. In Figure 8, it can be seen, that the zeroes of the modified Legendre polynomials are much more evenly distributed (in this example $\alpha = 0.925$).

As can be seen from Figure 9, also here, the first 10-12 coefficients indeed decay exponentially, the amplitude of the rest of the modes is, more or less, constant.

By comparing 7c and d to 10c and d that the error of the derivative near the boundary was reduced ny an order of magnitude.

3 Numerical Examples.

In this section we shall numerically demonstrate the $1/\sqrt{n}$ decay of the error, compare the proposed method to moving least-squares and present an oscillatory example.

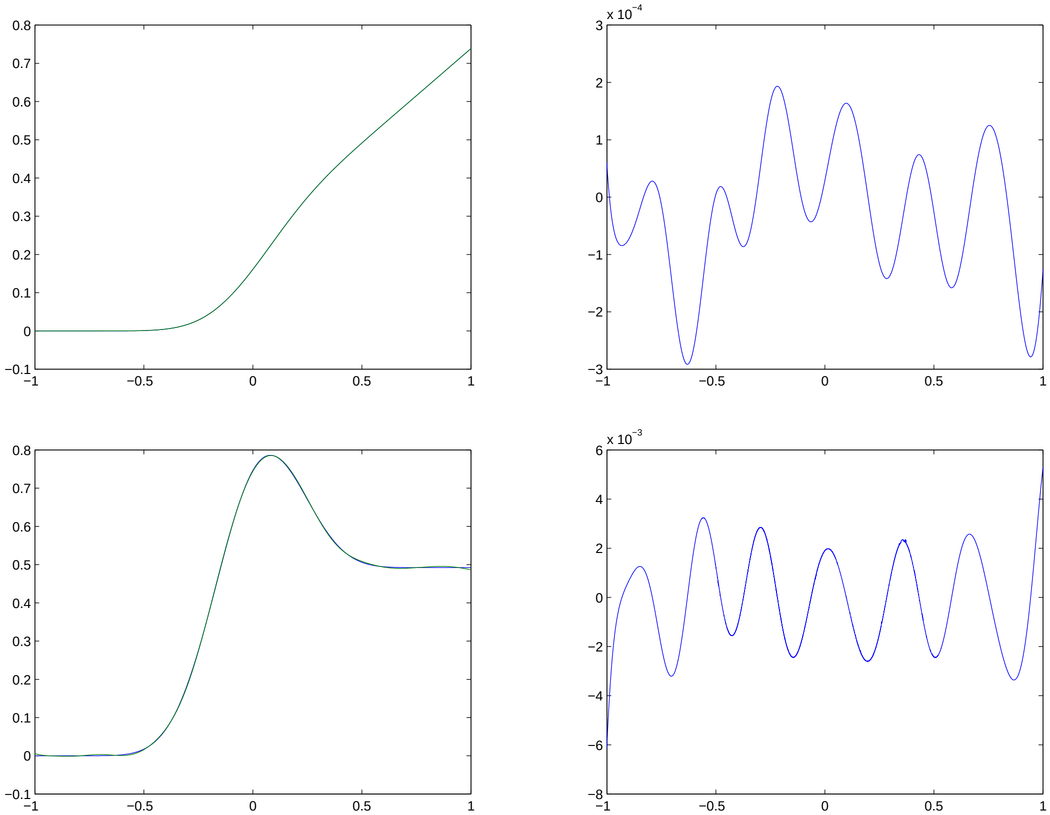


Figure 10: Expansion using modified Legendre polynomials. Top left: (a): the clean and the recovered signals. Top right, (b): the clean minus the recovered signals. Bottom left, (c): the derivatives of the clean and the recovered signals with respect to the transformed variable x. Bottom right, (d): the differences between the derivatives of the clean signal and the recovered signal with respect to x.

3.1 Errors dependence on the number of points.

In the precious section, an estimate of the dependence of the error on the number of sampling points was derived. According to (2.6), the error should decay as $1/\sqrt{n}$. In this section we demonstrate this result. We took the example from section 2.3 and decreased the number of points by taking every, fourth, ninth, ... , hundredth point. In Figure 11 the $\log_{10}(L_2 \text{ of; the error})$ vs. $\log_{10}(N)$ and the linear fit of the data, as calculated by the MATLAB command `polyfit`, for both the approximation to the 'clean signal' and its derivative. As can be seen, there is a good agreement with the theoretical estimate.

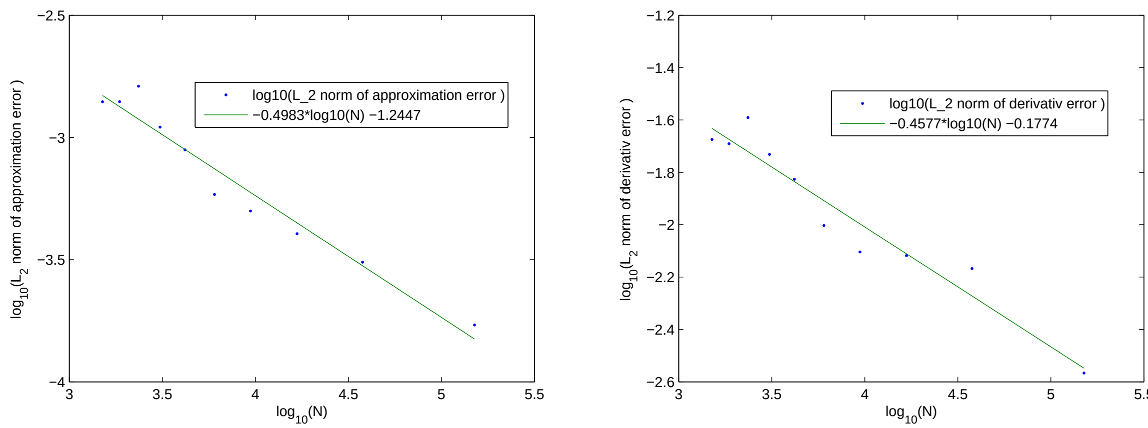


Figure 11: Errors dependence on the number of points. Expansion using modified Legendre polynomials. Left: (a) $\log_{10}(L_2$ of; the error) vs. $\log_{10}(N)$. Right (b): $\log_{10}(L_2$ of; the derivative error) vs. $\log_{10}(N)$.

3.2 Moving least-squares.

The moving least-squares is a method of reconstructing continuous functions from a set of unorganized point samples via the calculation of a weighted least-squares measure biased towards the region around the point at which the reconstructed value is requested, see [10]. Typically, a gaussian in the form:

$$e^{\frac{(x-x_j)^2}{\sigma^2}} \tag{3.1}$$

is used as a weight function.

In our examples we ran the example from the pervious section with a second order polynomial fitting near each point x_j. The results of the approximations are presented in Figures 12 and 13. For approximating the clean signal, the moving least square method good results and the error decay reciprocally as the square root of the gaussian 'width', σ. The convergence rate of the derivative is much larger, however the actual errors are very large. In this particular example, comparable results were obtain using our proposed method and by using moving least-squares with $\sigma = 0.1$ this is not surprising, since the effective width of the gaussian is of the same order then the whole domain. There are, however two fundamental differences; the fact that here only low (second) order polynomials were used in the neighborhood of each x_j, makes accurate representation of moderately oscillating functions, impossible. This will be demonstrated in the next example. The other

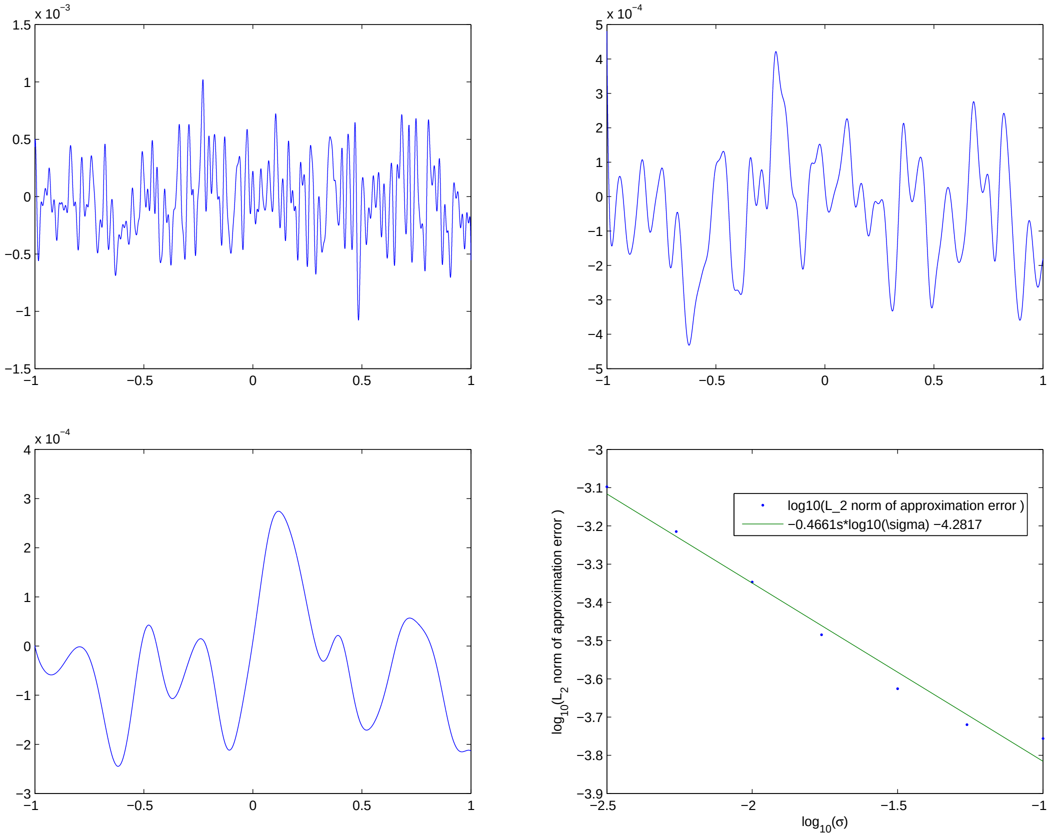


Figure 12: Moving least-squares. Approximation error for different σ^2 Top left: (a): $\sigma^2 = 0.0001$. Top right, (b): $\sigma^2 = 0.001$. Bottom left, (c): $\sigma^2 = 0.01$. Bottom right, (d): $\log_{10}$ error vs. $\log_{10}\sigma$.

difference is that on a Pentium 4, 2.39GHz PC, running a MATLAB ver. 7.2 code, it took 18-21 seconds to run our method and 7 hours to run the moving least-squares code, with $\sigma = 0.1$.

3.3 Oscillatory example.

The previous example was smooth and non-oscillatory, thus it could be accurately approximately by 10-15 Legendre, or modified Legendre modes. In this example we would like to recover

$$u = \frac{\cos(30\,x)}{1+x^2}\,, \tag{3.2}$$

and it's derivative.

u and it's approximation using moving least-squares method, with $\sigma^2 = 0.01$ are shown

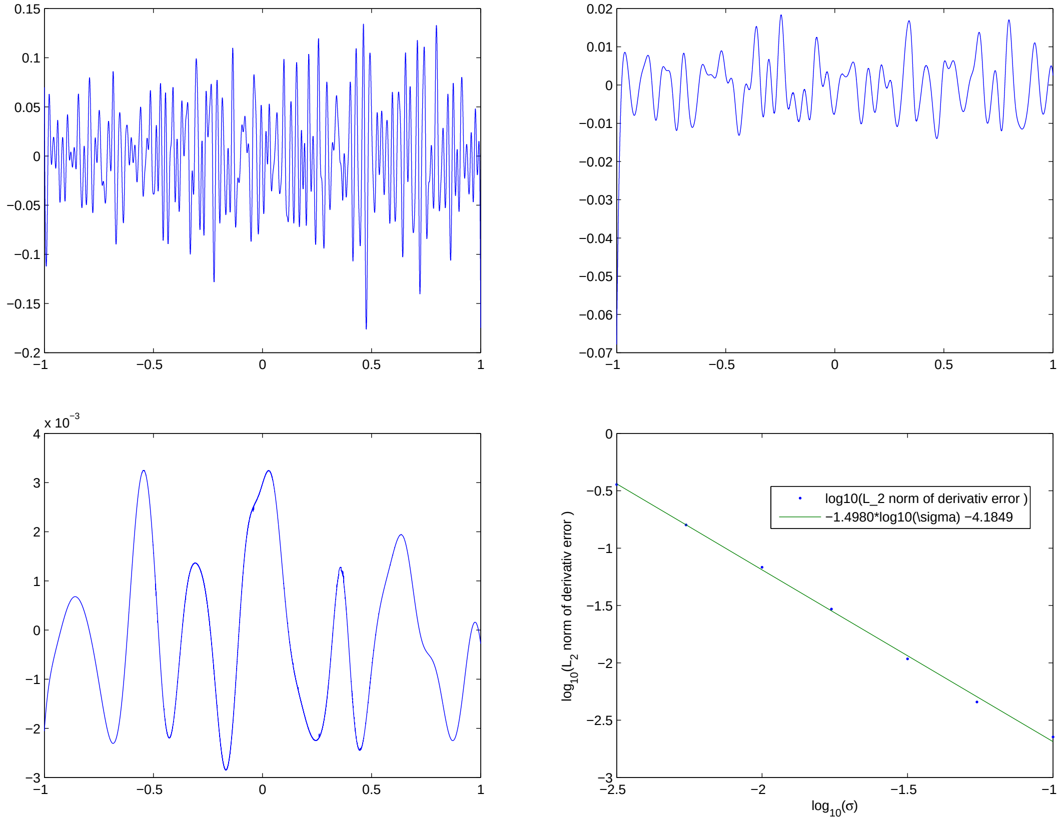
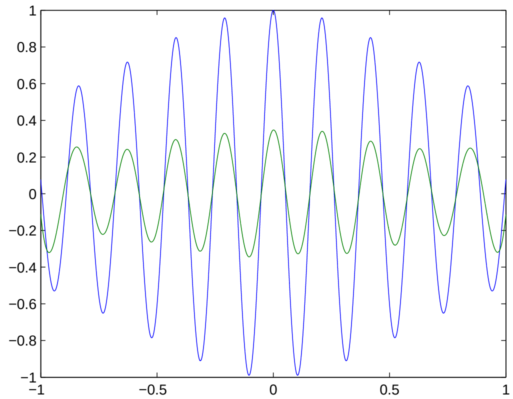


Figure 13: Moving least-squares. Derivative error for different σ^2 Top left: (a): $\sigma^2 = 0.0001$. Top right, (b): $\sigma^2 = 0.001$. Bottom left, (c): $\sigma^2 = 0.01$. Bottom right, (d): $\log_{10}$ error vs. $\log_{10}\sigma$.

in 14a. Clearly this approximation fails.

A plot of $\log_{10}(\text{abs}(n\text{th mode}))$ vs. n is presented in Figure 14b. Unlike the previous case, here about 30 modes are above the nose level. In this example, u is an even function, therefore all the odd modes should be zero. It can be seen here that the noise level, is the same in these low modes.

It can be seen from Figure 15 that even though many more modes are needed for accurately approximate $\frac{\cos(30\,x)}{1+x^2}$ then (2.8), the relative errors are comparable to the previous example.



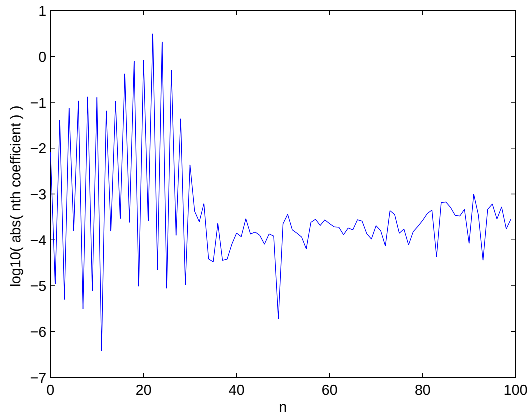


Figure 14: Oscillatory example. Left: (a) The function u and it's approximation using moving least-squares. Right (b): Expansion using modified Legendre polynomials. $\log_{10}$ of the absolute value of the amplitude of the modes.

4 Implementation.

In this section a simplified algorithm for approximating the clean signal using the modified Legendre method is given. The remarks are listed after the algorithm.

```
Read data
            % the data is in the form of a table (x_j, u_j), j=0..N


y_j = 2*(x_j - x_0)/(x_N - x_0)-1
            % map the input point from [x_0 ... x_0] to [-1, 1]


Compute the wights w_j for the numerical integration
            % Remark 1


Compute the wight function
      hat_w_j = alpha/( arcsin(alpha) sqrt(1-alpha^2 y_j^2) )
            % Remark 2


Compute hat_P_0(y) and hat_P_1(y)
            % Remark 3
```

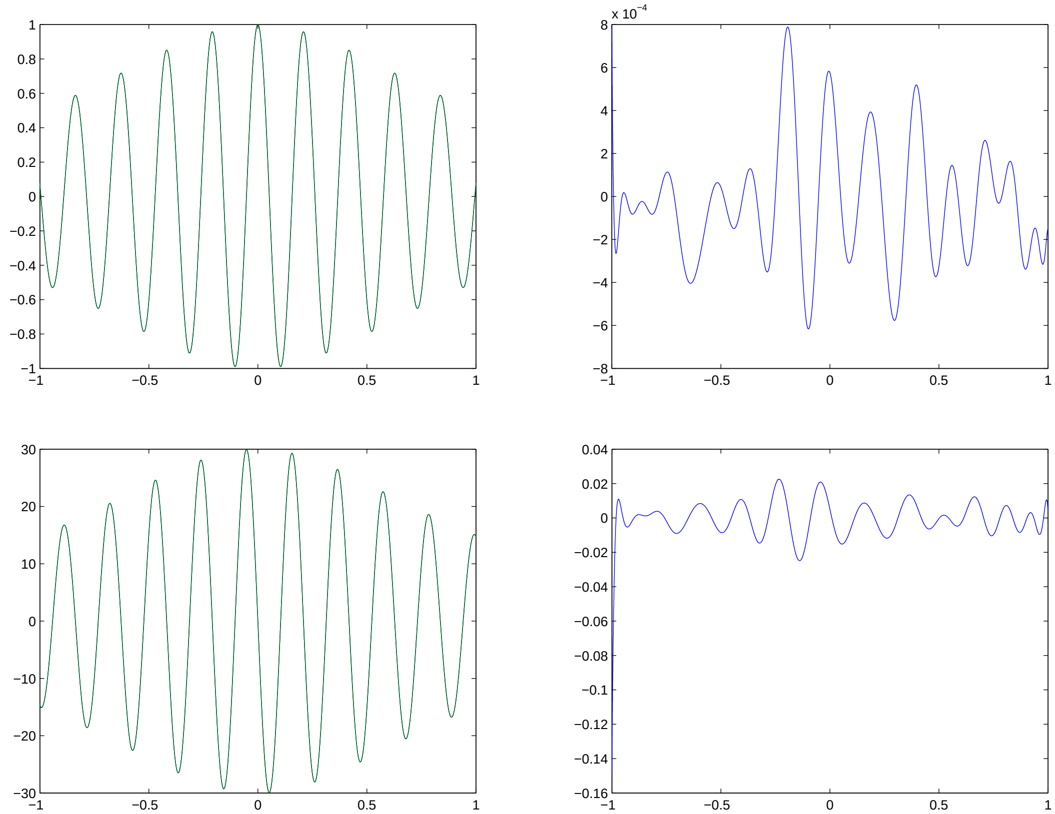


Figure 15: Oscillatory example. Expansion using modified Legendre polynomials. Top left: (a): the clean and the recovered signals. Top right, (b): the clean minus the recovered signals. Bottom left, (c): the derivatives of the clean and the recovered signals with respect to the transformed variable x. Bottom right, (d): the differences between the derivatives of the clean and the recovered signals with respect to x.

```
Compute c_0 = sum_j  w_j hat_w_j u_j hat_P_0(y_j)
        c_1 = sum_j  w_j hat_w_j u_j hat_P_1(y_j)
            % Remark 4

From n = 2 to number_of_computed_modes
            % Remark 5

    Compute hat_P_n(y) =
         ( (2*n-1)*( arcsin(alpha*y)/arcsin(alpha) )*hat_P_(n-1)(y)
            - (n-1)*hat_P_(n-2)(y) )/n
```

```
          % Remark 6

    Compute c_n = sum_j  w_j hat_w_j u_j hat_P_n(y_j)
          % Remark 4

    Compute hat_c_(n) = hat_c_(n) * exp( -(n/cutoff)^(2*s) )
          % Remark 7

end

Compute recovered_u(y) = sum_n hat_c_(n+1) * hat_P_n(y)
          % Remark 8
```

Remarks:

1. In the numerical examples presented in this paper the composite trapezoid rule was used.
2. See (2.17).
3. See (2.19).
4. See (2.21).
5. The `number_of_computed_modes` should be larger then the 'noise level', but it is expected to be small with respect to N.
6. This is the forward recurrence formula, (2.22). It is inaccurate for large `number_of_computed_modes`. However for small `number_of_computed_modes`, the error can be tolerated.
7. Applying the exponential filter for smoothly truncating the series.
8. The derivative of the recovered signal can be computed directly by a numerical scheme, which is simpler then summing up the derivative of $\hat{P}(y)$. Note that the recovered signal is given in the transformed variable y rather then the original variable x.

It should be noted that this is a simplified algorithm, rather then a practical one. Many details, such as the fact that only three $\hat{P}(y)$ are needed to be stored at each time, therefore the computation of the recovered signal at the last step, should actually be done in the main loop, were omitted for the sake of clarity.

5 Conclusions.

In this paper, a method for filtering a noisy signal and to approximate the 'clean' signal was presented. This method is based on projection of the signal onto an orthogonal set of modified Legendre polynomials, i.e. Legendre polynomial on transformed variable in which the the zeros of the polynomials are almost evenly distributed. Using this transformation causes the maximum of the derivatives to be of order n, rather then n^2. This reduces the effect of the errors in the coefficients.

Numerical examples demonstrates the efficacy of the method as well as the theoretical estimates.

Acknowledgments. We would like to thank Prof. David Levin from the School of Mathematical Sciences at Tel Aviv University for the useful discussions.

References

[1] A. Bhandari, B. W. Sheldon and S. J. Hearne *Competition between tensile and compressive stress creation during constrained thin film island coalescence*, J. of App. Phys. 101, 033528 (2007), 157-469.

[2] J. Weickert, *Anisotropic Diffision in Image Processing*, (European Consortium for Mathematics in Industry), Stuttgart : Teubner, 1998.

[3] D. Gottlieb and S.A. Orzag, *Numerical Analysis of Spectral Mehods: Theory and Applications.* CBMS-NSF **26**. Philadelpia: SIAM, (1977).

[4] J. S. Hesthaven, S. Gottlieb and D. Gottlieb, *Spectral Methods fo Time-Dependent Problems*, Canbridge monographs on applied an computational mathematics, Canbridge university perss, 2007.

[5] D. Gottlieb and C.W. Shu, *On the Gibbs Phenomenon and its Resolution*, SIAM Review, **39** (1997), 644-668.

[6] D. Funaro, *Polynomial Approximatio of Differential Equations*, Lecture Notes in Physics, **8**. Berlin: Springer-Verlag, 1992.

[7] , G. Szegö, *Orthogonal Polynomial*, Colloquium Publications, **23**. American Mathematical Society, Providence RI, 1939.

[8] D. Kosloff, H. Tal-Ezer, *A Modified Chebyshev Pseudospectral Method Whith an* $O(N^{-1})$ *Time Step Reduction*, J. of Comp. Phys. 104 (1993), 157-469.

[9] W. S. Don, A. Solomonoff, *Accuracy Enhancement for Higher Derivatives Using Chebyshev Collocation and a Mapping Technique*, SIAM J. Sci. Comput. 18 (1997), No. 4, 1040–1055.

[10] W. Holger, *Scattered data approximation*, Canbridge monographs on applied an computational mathematics, Canbridge university perss, 2005.