Journal of Approximation Theory and Applied Mathematics

Notes on Nonparametric Regression with Wavelets

M. Schuchmann and M. Rasguljajew from the Darmstadt University of Applied Sciences

Abstract

In this article we describe a nonparametric regression using a wavelet basis. There exist different approaches for a regression based on wavelets. For the regression which we use in our article we must calculate coefficients over an integral but for further regressions with the same number of points we can use the same coefficients. We describe the regression for points in R^2 and R^3 and we use a help function, which is constant on an area around the points in contrast to other approaches where the regression function is shifted.

Regression for Functions $f: R \longrightarrow R$

We have got n points $(x_1,y_1), (x_2,y_2), ..., (x_n,y_n)$ for example from measurements. We assume, that there exists a causal relationship between the x_i and the y_i, like

(1) $$y_i = f(x_i) + e_i \ .$$

The e_i represents the error, for example the measurements error. Like in the classical statistics y_i can be a realisation of a random variable, so that we have the theoretical model

$$Y_i = f(x_i) + E_i \ .$$

e_i is a random variable with mean 0 and variance σ^2. With additional assumptions we can assume, that the e_i are independent identically $N(0,\sigma^2)$ distributed, but that's not necessary for our method. The function f is often unknown in praxis, but in the classical regression analysis the type is known. Here even the type of the function f can be unknown.

If we apply a continuous approximation, where we knew the function f, we get an orthogonal projection form f on V_j with

$$f_j(x) = \sum_k f_k^j \phi_{j,k}(x) ,$$

$$\text{with} \quad f_k^j = \int_{-\infty}^{\infty} \phi_{j,k}(x) f(x) dx \ .$$

For easier notation we assume, that the variable x has values out of the interval [0, 1] and $x_i = i/n$, with $i = 1,2,..,n$. With the resolution j we could adjust, how many or how small the details are, that we want take in account. Because f is now unknown, we define a function $\tilde{f}$ which has around the point $x = x_i$ the constant function value y_i, so

$$\tilde{f}(x) = y_i \text{ for } \frac{i-1}{n} + \frac{1}{2n} \le x < \frac{i}{n} + \frac{1}{2n} .$$

So we define:

$$\tilde{f}(x) = \begin{cases} y_i & ; \frac{2(i-1)+1}{2n} \le x < \frac{2i+1}{2n} \\ 0 & ; else \end{cases}$$

We define the help function $\tilde{f}$ so, that the function is constant around the measurement point x_i. There exists definitions, where the help function is constant from on $[x_i, x_{i+1}]$, but here the graph of regression function is shifted to the right.

For easier notation we assume that the scaling function is real-valued. Now we calculate a best approximation $\tilde{f}_j$ of $\tilde{f}$ in V_j and we use this function as a regression function of the points (x_i, y_i). So we get an approximation of the coeffitions f_k^j with:

$$\tilde{f}_k^j = \int_{-\infty}^{\infty} \phi_{j,k}(s)\tilde{f}(s)ds = \sum_{i=1}^{n} \int_{\frac{2(i-1)+1}{2n}}^{\frac{2i+1}{2n}} \phi_{j,k}(s)\tilde{f}(s)ds = \sum_{i=1}^{n} y_i \int_{\frac{2(i-1)+1}{n}}^{\frac{2i+1}{2n}} \phi_{j,k}(s)ds$$

So we get $\tilde{f}_j$:

$$\tilde{f}_j(x) = \sum_{k=-\infty}^{\infty} \tilde{f}_k^j \cdot \phi_{j,k}(x) = \sum_{k=-\infty}^{\infty} \sum_{i=1}^{n} y_i \underbrace{\int_{\frac{2(i-1)+1}{2n}}^{\frac{2i+1}{2n}} \phi_{j,k}(s)ds}_{:=a^j{}_{i,k}} \cdot \phi_{j,k}(x)$$

$$= \sum_{i=1}^{n} y_i \sum_{k=-\infty}^{\infty} a^j{}_{i,k} \cdot \phi_{j,k}(x)$$

If we use the same number of measurement points n, we don't need do calculate the integral above twice, we can use the same coefficients $a^j{}_{i,k}$.

In practice we don't need the whole summation area $\mathbb{Z}$ for the index k, because we have (here [0, 1]) a compact interval and either the scaling function ϕ has compact support or it will vanish for big or small arguments.

Example 1:

We simulated measurement points like in formula (1) and used a normal distributed error with mean $\mu = 0$ and standard deviation $\sigma = 0.01$. We used the function f:

$$f(t) = \begin{cases} \sin(2\pi \cdot t) & \text{if } 0 \le t < 1/2 \\ \sin(4\pi \cdot t) & \text{if } 1/2 \le t < 1 \\ 0 & \text{else} \end{cases}$$

We set $n = 20$. In the graph we see, that f is not differentiable an the point $x = ½$:

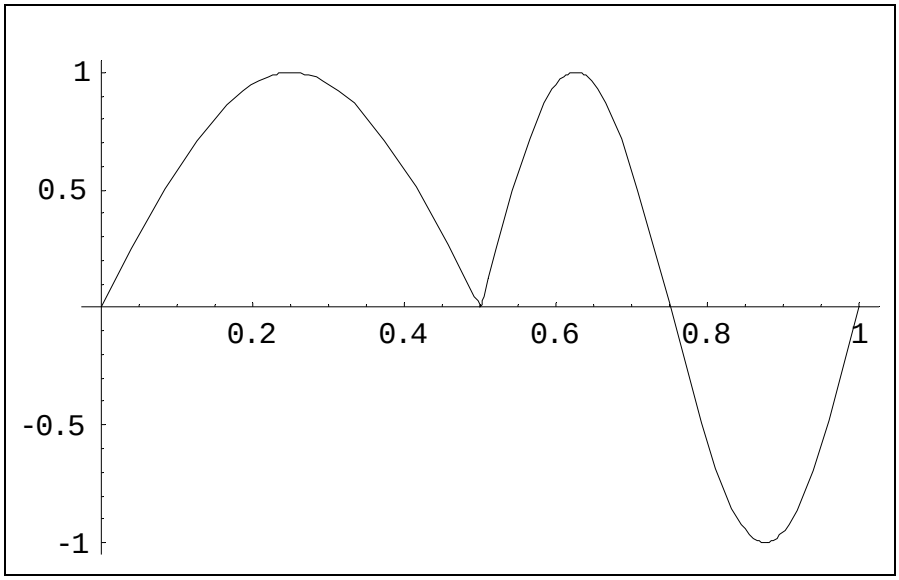

Figure 1: Graph of f

Here is the plot of the points:

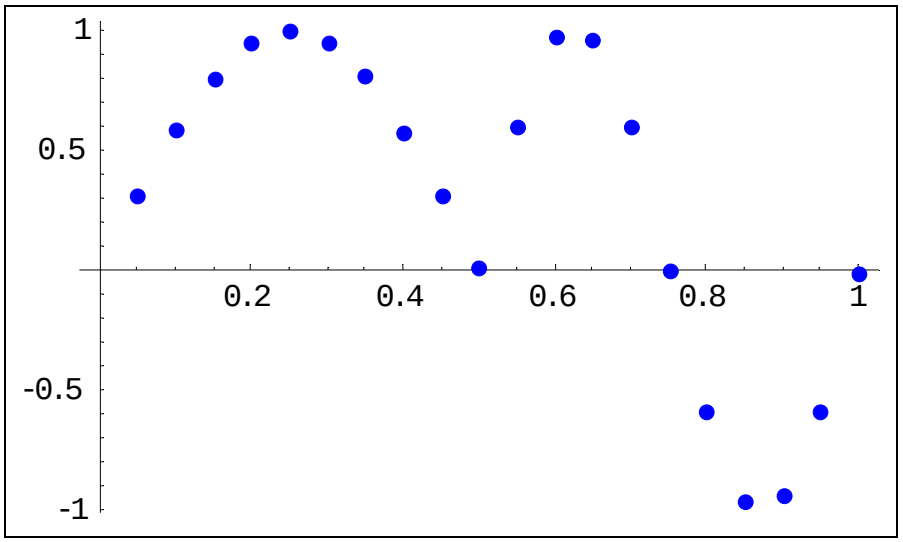

Figure 2: Simulated measurement points

Using the Haar wavelet und setting $j = 4$, $k = -16,\ldots,16$, we get the following regression function:

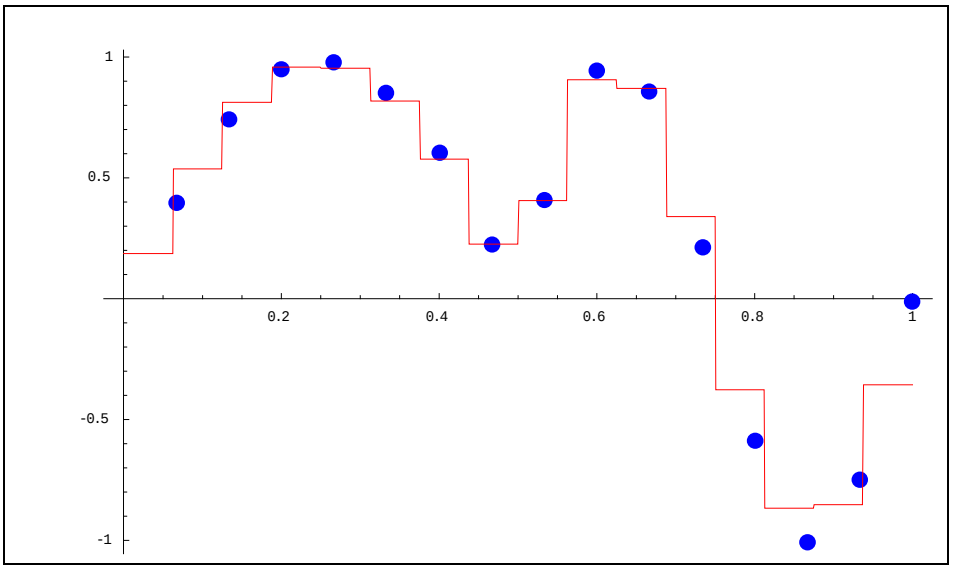

Figure 3: Graph of the regression function using the Haar wavelet

With the Shannon wavelet we get a useful regression function:

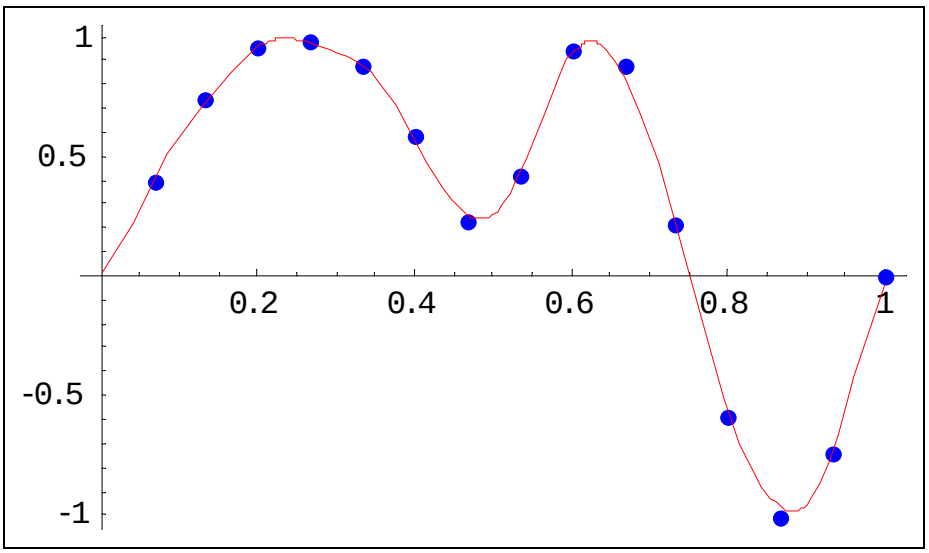

Figure 4: Graph of the regression function using the Shannon wavelet

Remarks 1:

1) The nonparametric regression with the Shannon wavelet can be applied under http://www.statistikcloud.at/htdocs-eng/Wavelet-Regression-v3 (for small examples).

2) A method for discrete approximation, which causes less effort, is the direct application of the Least Squares Method. So we would solve

$$\min Q(a)$$

$$\text{with } Q(a) = \sum_{i=1}^{n} (y_i - f_j(t_i))^2$$

and

$$f_j(t) = \sum_{k=k_{min}}^{k_{max}} a_k \phi_{j,k}(t).$$

This is a quadratic problem, because f_j is linear in a.

Regression for Functions $f: R^2 \longrightarrow R$

We assume that we got points $(s_i, t_l, y_{i,l})$ and

(2) $y_{i,l} = f(s_i, t_l) + e_{i,l}$ with $i = 1,2,...,n_1$, $l = 1,2,...,n_2$ and $(s_i, t_l) \in G$

with

$$f: R^2 \rightarrow R.$$

Here are $e_{i,l}$ the errors, which are for example realisations of independent identically $N(0, \sigma^2)$ random variables $E_{i,l}$. We assume $s_i \neq s_j$ and $t_i \neq t_j$ for $i \neq j$.

Now we choose a partition $(Z_{i,l})$ from G:

(1) $\overline{\bigcup_{i,l} Z_{i,l}} = G$ and $Z_{i,l} \cap Z_{i',l'} = \{\}$ with $i \neq i'$ and $l \neq l'$.

(2) $(s_i, t_l) \in Z_{i,l}$

We construct a function $\tilde{f}$, which has on the area $Z_{i,l}$ the constant function value $y_{i,l}$:

$$\tilde{f}(s,t) = y_{i,l} \text{ if } (s,t) \in Z_{i,l},$$
$$\tilde{f}(s,t) = 0 \text{ else.}$$

So $\tilde{f}$ looks like:

$$\tilde{f}(s,t) = \sum_{i=1}^{n_1} \sum_{l=1}^{n_2} y_{i,l} \cdot 1_{Z_{i,l}}(s,t) \ ,$$

with $1_A(s,t) = 1$ if $(s,t) \in A$ and $1_A(s,t) = 0$ else.

The regression function $\tilde{f}_j$ we get analogous to the two dimensional case:

$$\tilde{f}_j(s,t) = 2^j \sum_{k_1,k_2} \tilde{f}^j_{k_1,k_2} \cdot \phi(2^j s - k_1, 2^j t - k_2),$$

with

$$\tilde{f}^j_{k_1,k_2} = 2^j \cdot \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \tilde{f}(s,t) \cdot \phi(2^j s - k_1, 2^j t - k_2) \, ds \, dt$$

$$= 2^j \cdot \sum_{i=1}^{n_1} \sum_{l=1}^{n_2} y_{i,l} \cdot \underbrace{\iint_{Z_{i,l}} \phi(2^j s - k_1, 2^j t - k_2) ds dt}_{:= a^j{}_{i,k_1,k_2}} \ .$$

Using the same partition ($Z_{i,l}$), we need to evaluate the integral above only once if we save the coefficients $a^{j}{}_{i,k_1,k_2}$.

If the support of ϕ is compact, we must only take the following $k_1,k_2 \in Z$ in account:

$$\{(k_1, k_2) \mid supp\,\phi(2^j s - k_1, 2^j t - k_2) \cap G \neq \{\}\}\ .$$

If the support of ϕ is not compact, we use only the following $k_1,k_2 \in Z$ in the summation above:

$$\{(k_1, k_2) \mid |\phi(2^j s - k_1, 2^j t - k_2)| > \varepsilon \text{ for } (s,t) \in G\}$$

With a useful $\varepsilon > 0$. In many examples we saw, that the method is relatively insensitive and even with scaling functions without compact support we need not many basis coefficients for a good regression. Using an equidistant grid, we get the s_i and t_l with

$$s_i = s_1 + (i\text{-}1)h_s\ , \text{for } 1 \le i \le n_1$$

and

$$t_l = t_1 + (l\text{-}1)h_t\ , \text{for } 1 \le l \le n_2\ ,$$

with

$$h_s = \frac{s_{n_1} - s_1}{n_1 - 1} \text{ and } h_t = \frac{t_{n_2} - t_1}{n_2 - 1}.$$

Here you see the grid an $Z_{i,l}$ in a graph:

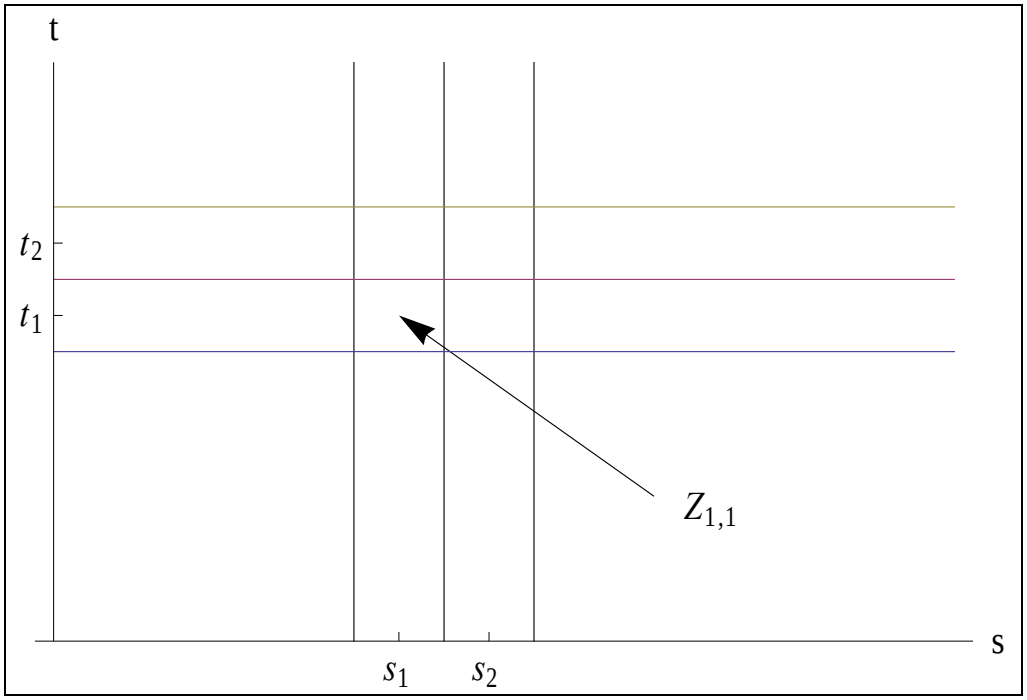

Figure 5: Scheme for the grid

Now the $Z_{i,l}$ are:

$$Z_{i,l} = \left[s_i - h_s/2, s_i + h_s/2\right) \times \left[t_l - h_t/2, t_l + h_t/2\right)$$

The formula for the coefficients $\widetilde{f}^{\,j}_{k_1,k_2}$ has then the following form:

$$\widetilde{f}^{\,j}_{k_1,k_2} = 2^j \cdot \sum_{i=1}^{n_1}\sum_{l=1}^{n_2} y_{i,l} \cdot \int_{t_l-h_t/2}^{t_l+h_t/2}\int_{s_i-h_s/2}^{s_i+h_s/2} \phi(2^j s - k_1, 2^j t - k_2)\, ds\, dt$$

Example 2:

We generated points by simulation measurements (formula (2)) and used the function

$$f(s,t) = e^{-s^2-t^2}.$$

We generated 100 function values over the area $[-3, 3]^2$ at equidistant points. The error was chosen normal distributed with mean 0 and the standard deviation 0.001.

We set:

$$n_1 = n_2 = 10;$$
$$s_1 = t_1 = -3;$$
$$s_{n_1} = t_{n_2} = 3.$$

Here you see the points $(s_i, t_l, y_{i,l})$ together with the graph of f:

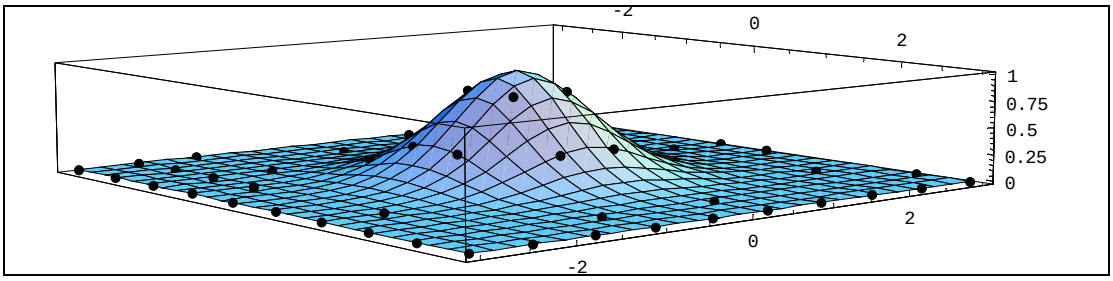

Figure 6: Graph from f and the regression points

We use the Daubechies wavelet of order 7. Here is the graph of the one dimensional scaling function:

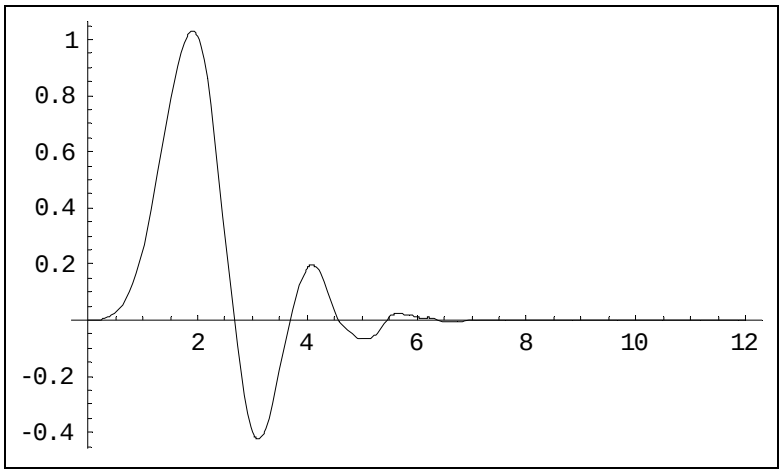

Figure 7: Graph of the scaling function from the Daubechies wavelet

And here is the graph of the scaling function with two arguments:

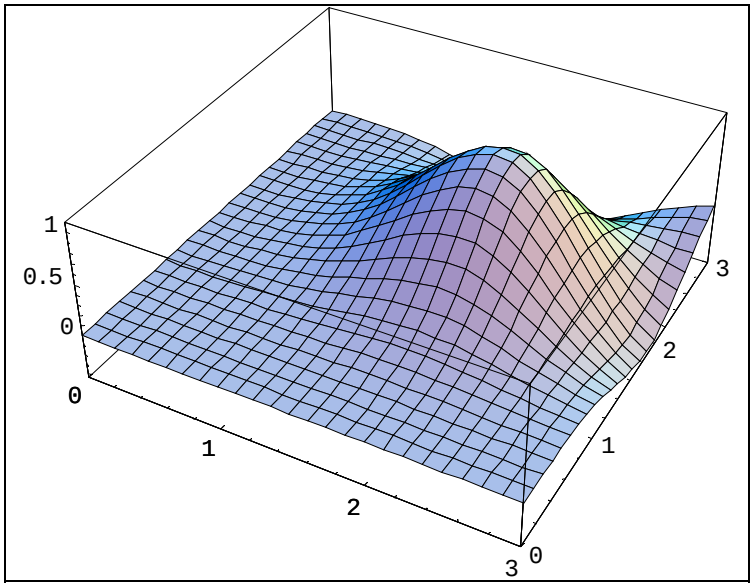

Figure 8: Graph of the scaling function with two arguments from the Daubechies wavelet

Here is the graph of the approximation function at resolution $j = 0$:

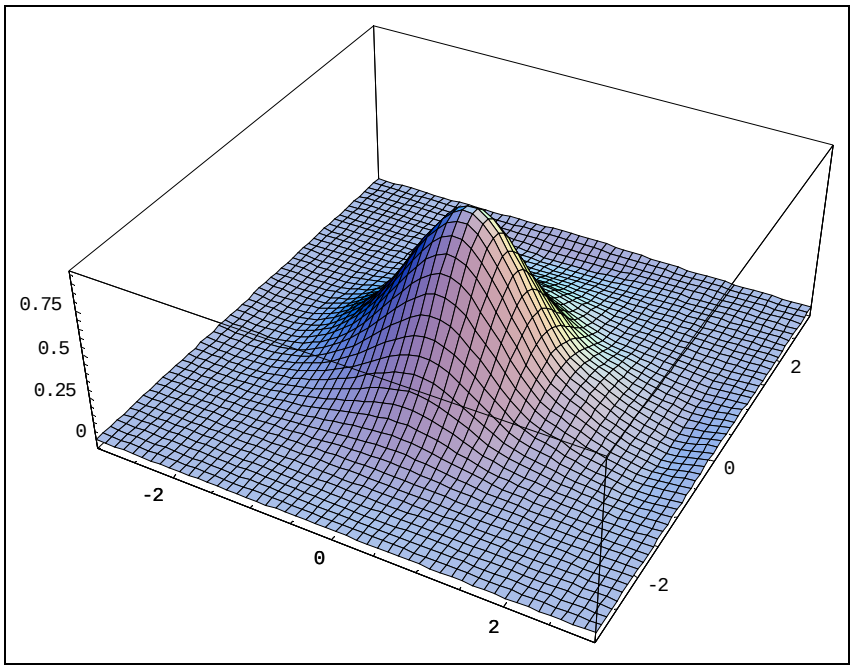

Figure 9: Graph of the approximation function for $j = 0$

The graph of the approximation function at resolution $j = 1$:

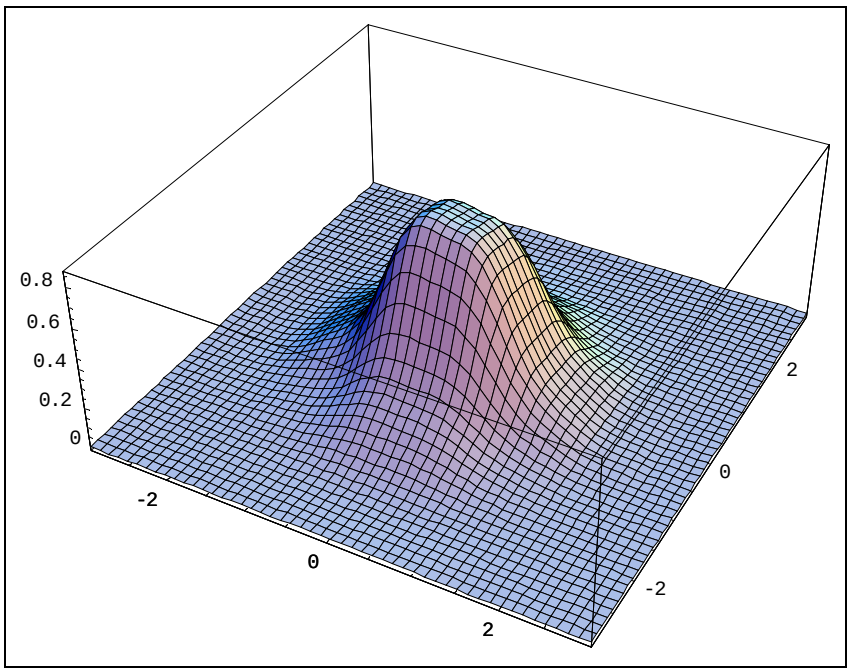

Figure 10: Graph of the approximation function for $j = 1$

The graph of the details d_0 ($= f_1 - f_0$):

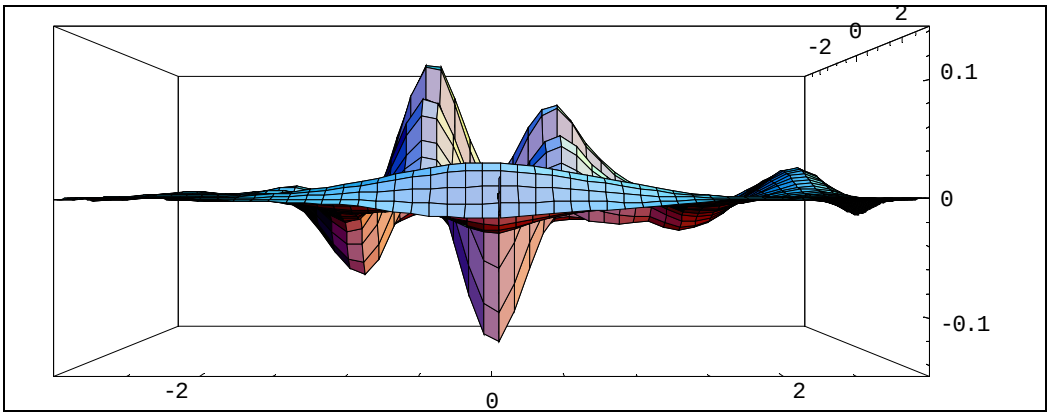

Figure 11: Graph of the Details for $j = 0$

The indices k_1 and k_2 were chosen from $-11 \cdot 2^j$ to $3 \cdot 2^j$ .We calculate the mean quadratic error at both resolutions:

$$\frac{1}{n_1 \cdot n_2} \sum_{i}^{n_1} \sum_{l}^{n_2} (y_{i,l} - f_0(s_i, t_l))^2 = 0.00043196 ,$$

$$\frac{1}{n_1 \cdot n_2} \sum_{i}^{n_1} \sum_{l}^{n_2} (y_{i,l} - f_1(s_i, t_l))^2 = 0.000261559 .$$

References

[1] R. T. Ogden. *Essential Wavelets for Statistical Applications and Data Analysis.* Boston, Berlin, Basel: Birkhäuser, (1997).

[2] M. Schuchmann. *Approximation and Collocation with Wavelets. Approximations and Numerical Solving of ODEs, PDEs and IEs.* Osnabrück, DAV, (2012).