Extrapolation and Approximation with a Wavelet Collocation Method for ODEs

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

Abstract

In this article we use a wavelet collocation method for an approximation of the solution of an ODE. We show that an approximation with the Shannon wavelet leads to better approximations than a Daubechies wavelet and we even can use the approximation for an extrapolation.

Introduction

We use the same Method as in "An Approximation on a Compact Interval Calculated with a Wavelet Collocation Method can Lead to Much Better Results than other Methods" described.

In the wavelet theory a scaling function ϕ is used, which belongs to a MSA (multi scale analysis). From the MSA we know, that we can construct an orthonormal basis of a closed subspace V_j, where V_j belongs to a the sequence of subspaces with the following property:

$$\ldots \subset V_{-1} \subset V_0 \subset V_1 \subset \ldots \subset L^2(R),$$

$\{\phi_{j,k}(t)\}_{k\epsilon Z}$ is an orthonormal basis of V_j with $\phi_{j,k}(t) = 2^{j/2}\phi(2^j t - k)$.

We use the following approximation function

$$y_j(t) := \sum_{k=k_{\min}}^{k_{\max}} c_k \cdot \phi_{j,k}(t) \quad \text{, with } \phi \in C^1(R).$$

k_{max} and k_{min} depend on the approximation interval $[t_0, t_{end}]$ (see [7]).

Now we can approximate the solution of an initial value problem $y' = f(y,t)$ and $y(t_0) = y_0$ by minimization of the following function

$$(1) \qquad Q(c) = \sum_{i=1}^{m} \left\| y_j'(t_i) - f(y_j(t_i), t_i) \right\|_2^2 + \left\| y_j(t_0) - y_0 \right\|_2^2 \ .$$

We apply the describe method in an example:

Applying the Method and Assessing an Approximation

Example 1:
We want to approximate the solution of

$$y' = -t\,y,$$
$$y(0) = 1$$

on the interval $I = [t_0, t_{end}] = [0, 3]$.

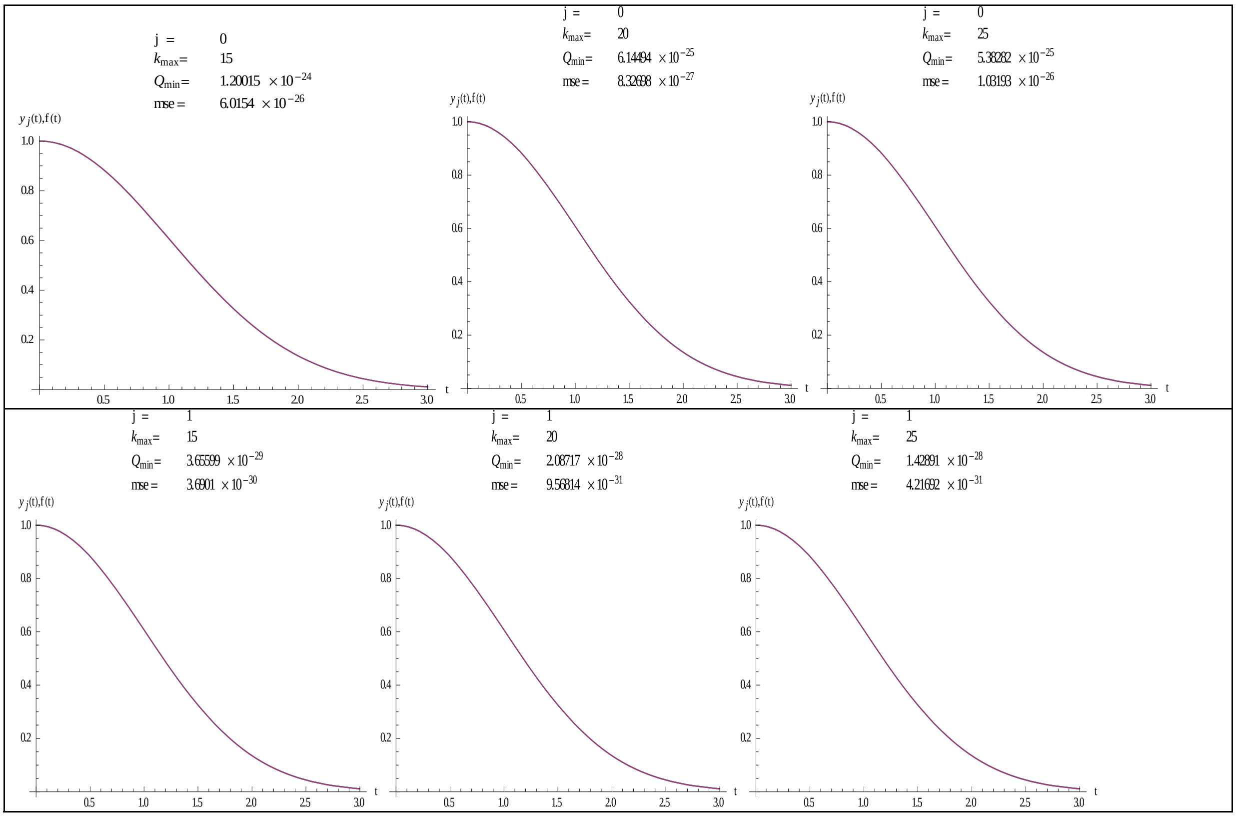
We know minimize Q for $k_{max} = -k_{min} = 15, 20, 15$ with $j = 0, 1, 2$ and $h = (t_{end} - t_0)/(4k_{max}) = 3/(4k_{max})$.

We calculate the mean squared error

(2) $$mse = \frac{1}{101}\sum_{i=0}^{100}\left(y(t_0 + i\cdot h_0) - y_j(t_0 + i\cdot h_0)\right)^2$$

with $h_0 = (t_{end} - t_0)/100 = 3/100$ and $Q_{min} = min\ Q(c) = Q(\hat{c})$. sse is the sum of squared errors with $sse = mse\cdot 101$.

Here we see the results of the graphs from y_j and y.



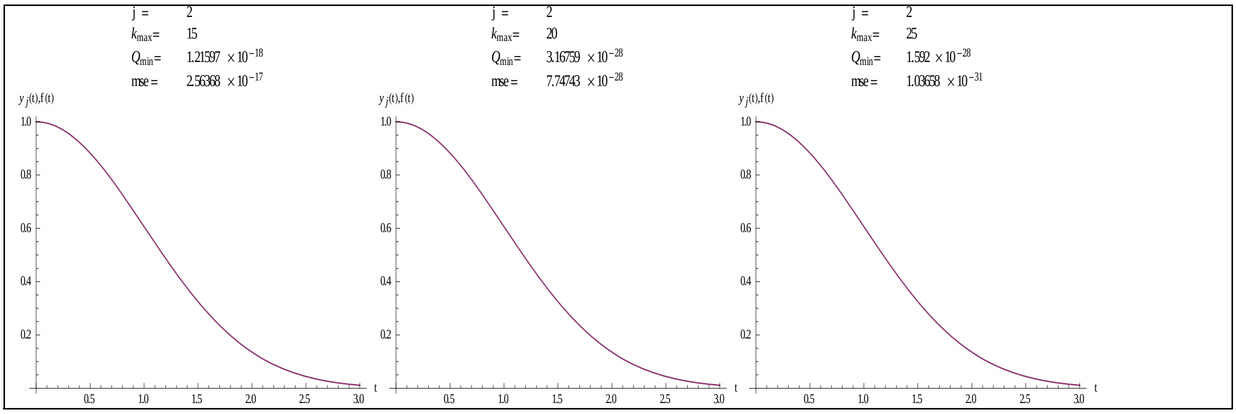


Figure 1. Graphs from y_j and y

As seen above, there is a correlation between Q_{min} and *mse*.

We now apply a linear regression on the points *(-ln(*Q_{min}*), -ln(mse))* (here we have: (55.0254, 53.4888), (55.749, 55.4352), (55.8814, 55.2207), (65.4786, 63.1568), (63.7366, 64.5066), (64.1155, 65.3259), (41.251, 33.5874), (63.3194, 57.8099), (64.0074, 66.7291))

for the different *j* und k_{max}:

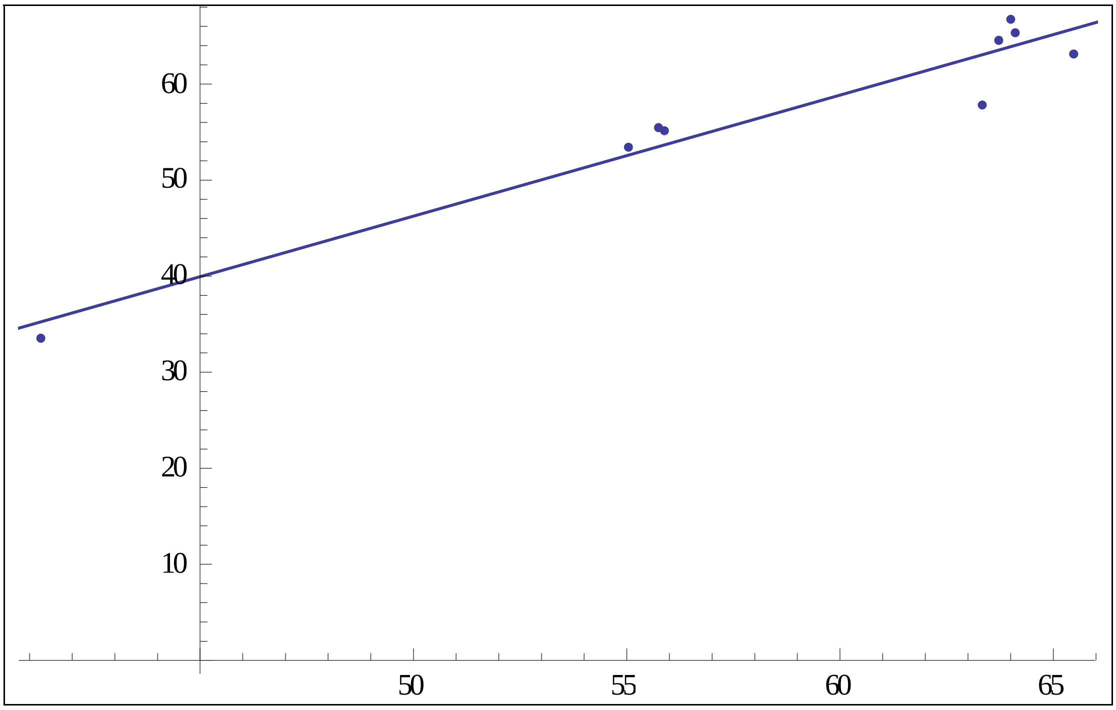


Figure 2. Linear Regression on the points *(-ln(*Q_{min}*), -ln(mse))*

Here is the regression table with a R^2 of 0.934136.

	Estimate	SE	TStat	PValue
1	-16.7718	7.48689	-2.24016	0.0600638
x	1.26041	0.126497	9.96392	0.0000219101

So we can see relativ good with Q_{min}, if an approximation is good. But it can occur that the residuals are very small at the collocation points but not between them. For that reason we later define Q_a to detect this.

Now we see the graph of d for k_{max} = 25 and j = 2 with

(3) $$d(t) = \left\| y_j'(t_i) - f(y_j(t_i), t_i) \right\|_2^2 :$$

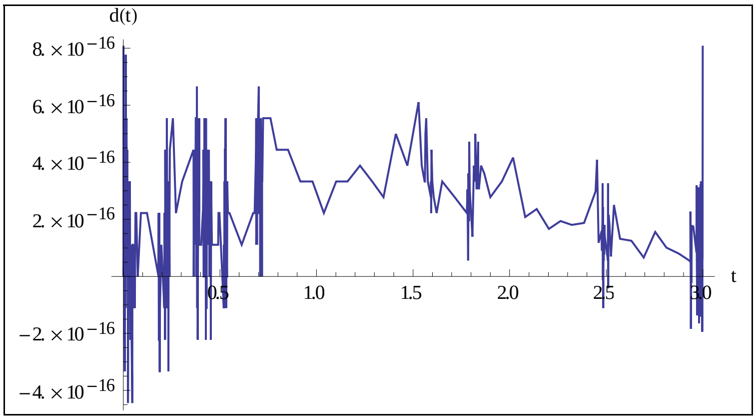


Figure 3. Graph of d for k_{max} = 25 and j = 2

Here is the Graph of *(k, -ln(mse))*:

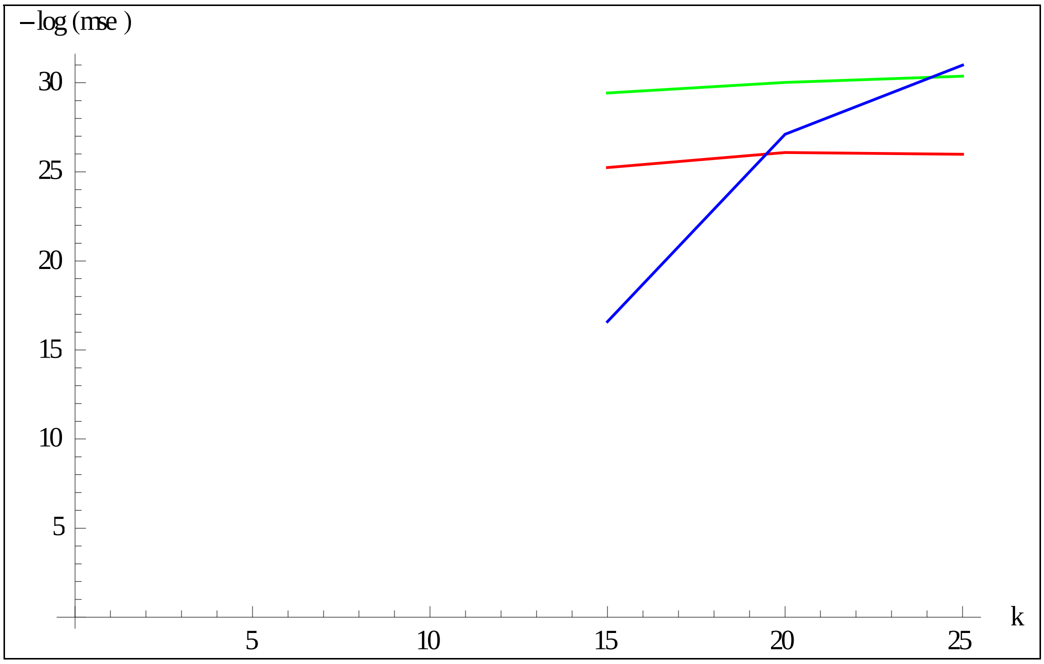


Figure 4. k_{max} vs. *-ln(mse)*

In red we see the points for j = 0 connected with lines, in green for j = 1 and in blue for j = 2. Theoretically the curves must increase because for a greater k_{max} the values of Q_{min} must shrink.

We have seen in this example, that there is a correlation between Q_{min} and *mse*. This relationship can - with an insufficient number of collocation points - no longer exist. Here, however, we can simply use another criterion Q_a where we can simply use the calculated $\hat{c}$ from the minimization:

(4) $$Q_a(\hat{c}) = \sum_{i=1}^{m_a} \left\| y_j'(\tau_i) - f(y_j(\tau_i), \tau_i) \right\|_2^2 + \left\| y_j(t_0) - y_0 \right\|_2^2$$

with $\tau_i = t_0 + i \cdot h/a$. $m_a = a \cdot m$ and an integer $a > 1$. If we use a big a, we should weight Q_a with $1/a$, but in different simulation we got with $a = 2$ good results:

(4a) $$\tilde{Q}_a(\hat{c}) = 1/a \cdot \sum_{i=1}^{m_a} \left\| y_j'(\tau_i) - f(y_j(\tau_i), \tau_i) \right\|_2^2 + \left\| y_j(t_0) - y_0 \right\|_2^2$$

For a good approximation Q_a should be small with any a. If h is too big, than $Q_a >> Q_{min}$.

Example 2:
We now calculate the approximations in example 1 with another h, which is too big $h = (t_{end} - t_0)/(1/2k_{max}) = 6/k_{max}$. Here we see with the following graphs, that Q_{min} can be small but *mse* and *sse* a relativ big (and so the approximation is not good). The worse approximation we can detect with Q_2 or Q_4.

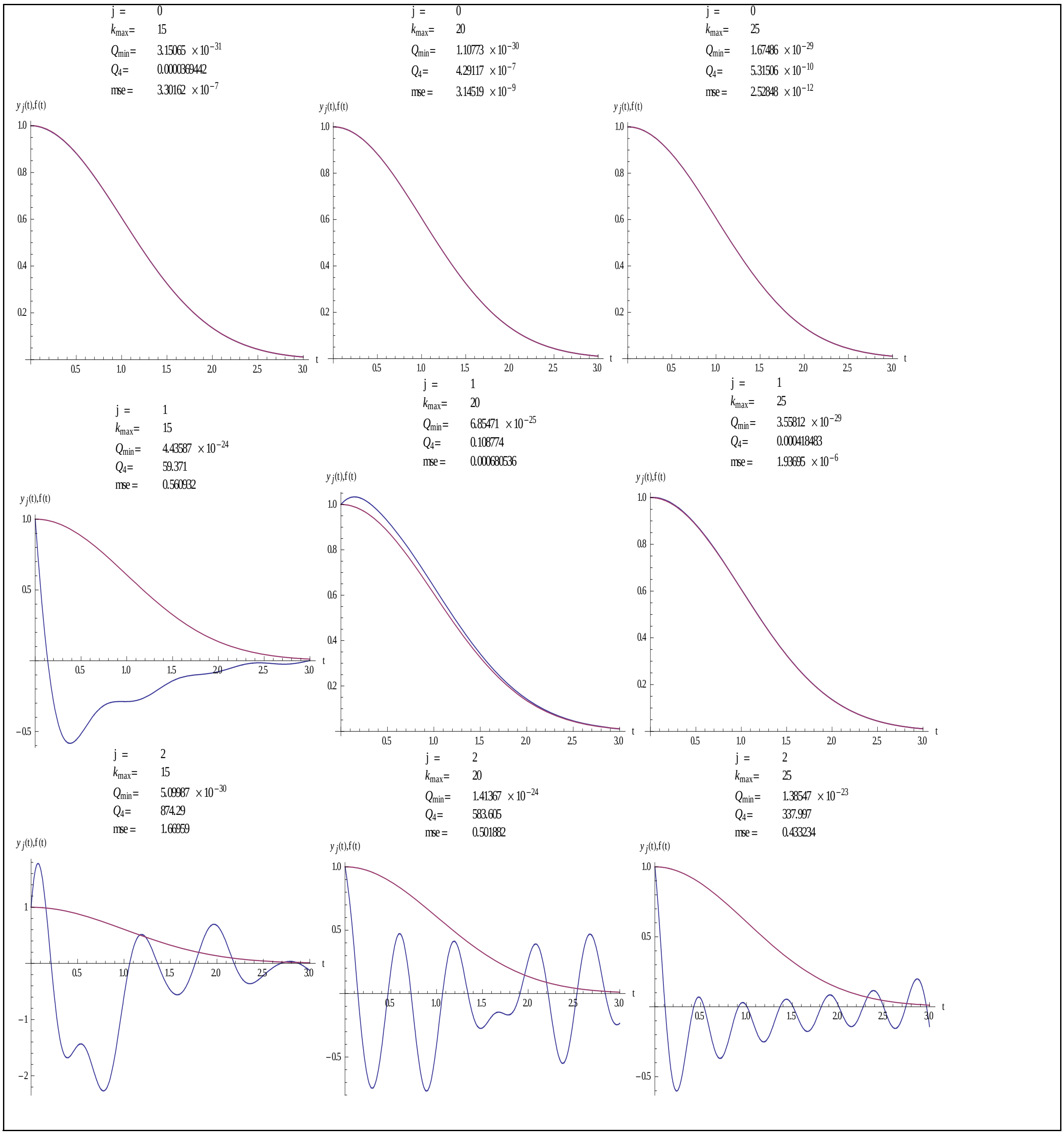


Figure 5. Graphs from y_j and y with a too big h

Now we see three linear regressions with the points *(-ln(Q_{min}), -ln(mse))* the points *(-ln(Q_4), -ln(mse))* and *(-ln(Q_2), -ln(mse))*. Here we can see, that the correlation between Q_{min} and *mse* is because of the too big step size not so high.

Q_{min} vs. mse:

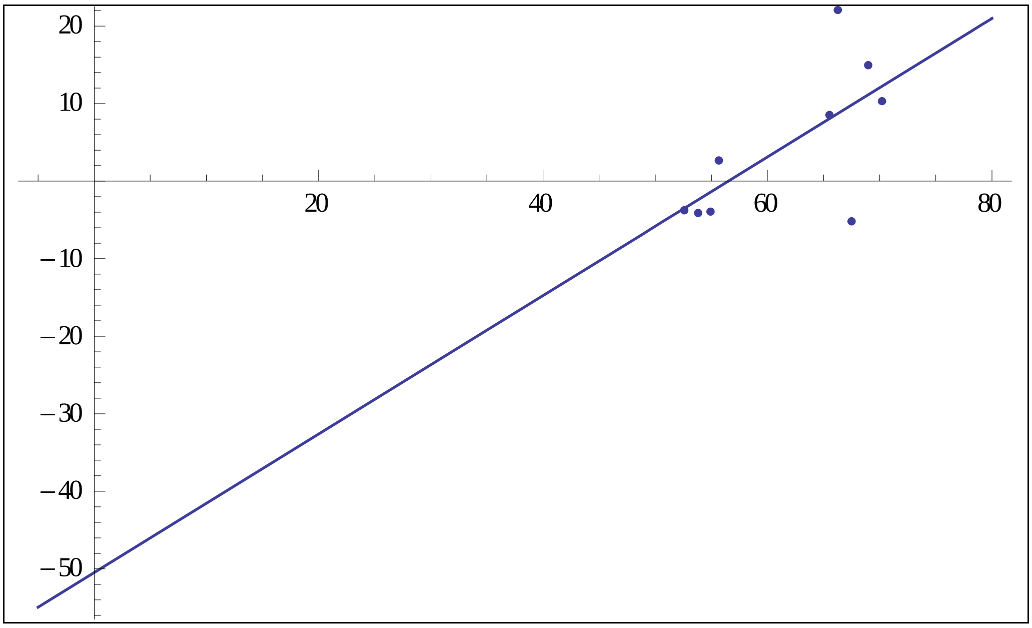


Figure 6. Linear Regression on the points *(-ln(Q_{min}), -ln(mse))*

Here is the regression table with a R^2 of 0.433152.

	Estimate	SE	TStat	PValue
1	-50.4893	23.9804	-2.10544	0.0732805
x	0.893277	0.386234	2.31279	0.0539648

Q_4 vs. mse:

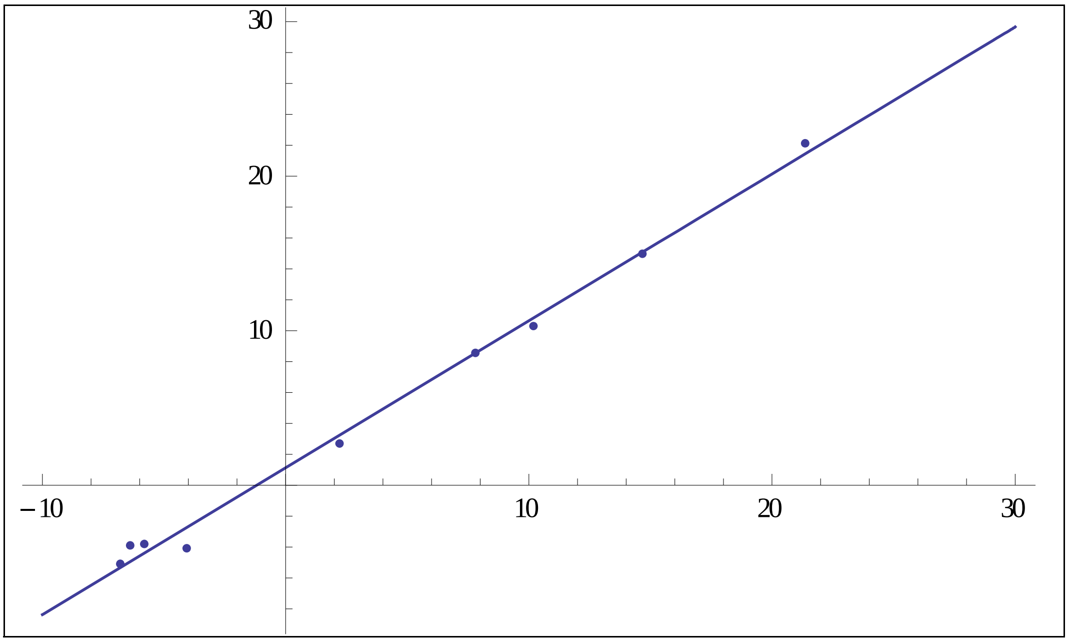


Figure 7. Linear Regression on the points *(-ln(Q_4), -ln(mse))*

Here is the regression table with a R^2 of 0.994715.

	Estimate	SE	TStat	PValue
1	1.13039	0.272975	4.14099	0.00434376
x	0.950635	0.0261912	36.296	3.13045×10^{-9}

Q_2 vs. mse:

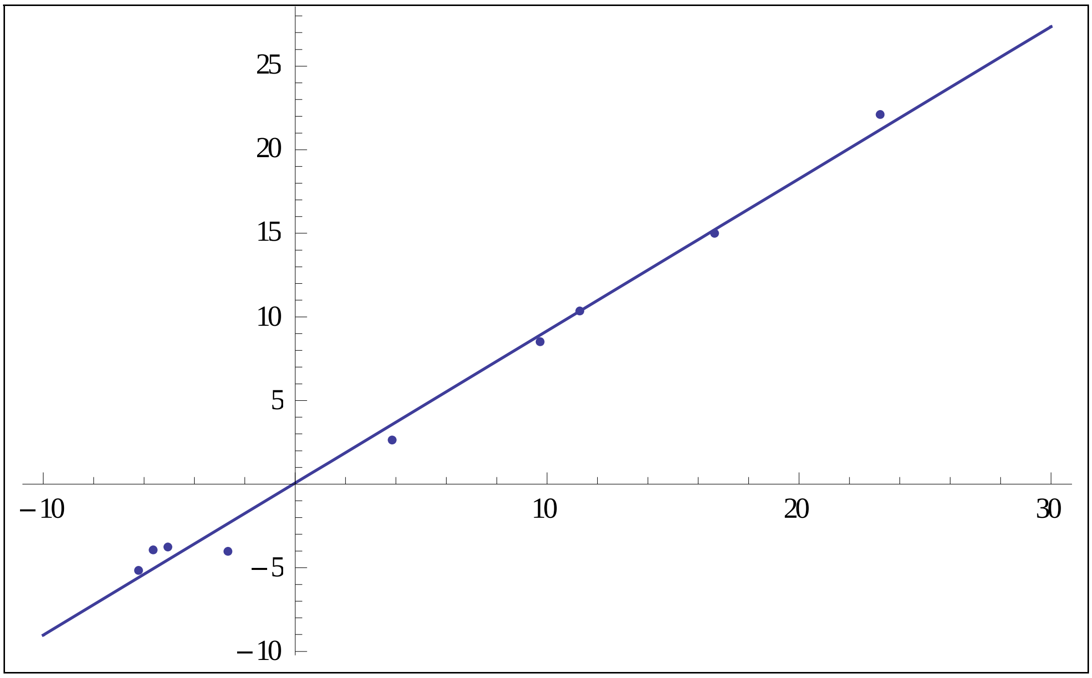


Figure 8. Linear Regression on the points *(-ln(Q_2), -ln(mse))*

Here is the regression table with a R^2 of 0.991411.

	Estimate	SE	TStat	PValue
1	0.0730749	0.362907	0.20136	0.846143
x	0.90963	0.0320016	28.4245	1.71502×10^{-8}

So Q_a is here a good criterion to detect a worse approximation.

For $j = 2$ and $k_{max} = 15$ the approximation was bad. Here was $h = 6/15 = 0.4$. We see with the graph of d (see (3)) that the residuals are only small at the collocation points:

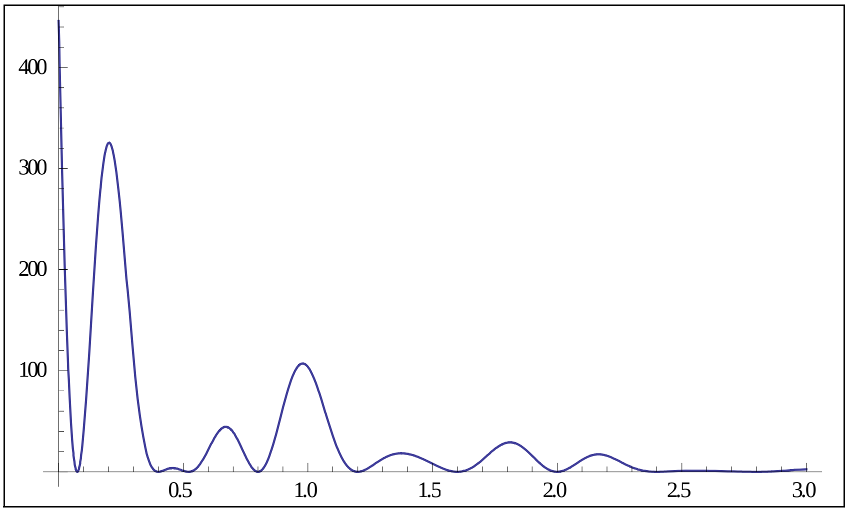


Figure 9. Graph of d for j = 2 and k_{max} = 15

Here we see how $d(t_i) = d(0.4i)$ is very small and between two collocation points d has very big function values.

For j = 0 and k_{max} = 20 the approximation was good. Here h = 6/20 = 0.3. We see the graph for that case:

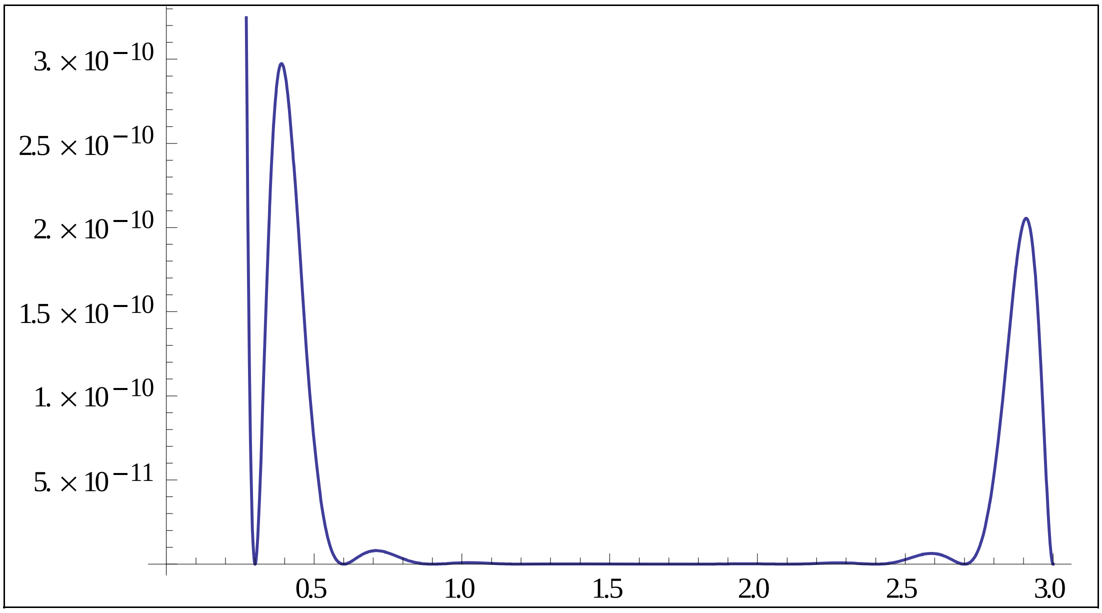


Figure 10. Graph of d for j = 0 and k_{max} = 20

Because we started with the collocation point t_1 in Q we get a relative big value of d at the point t_0.

Here we see the graph with the whole plot range:

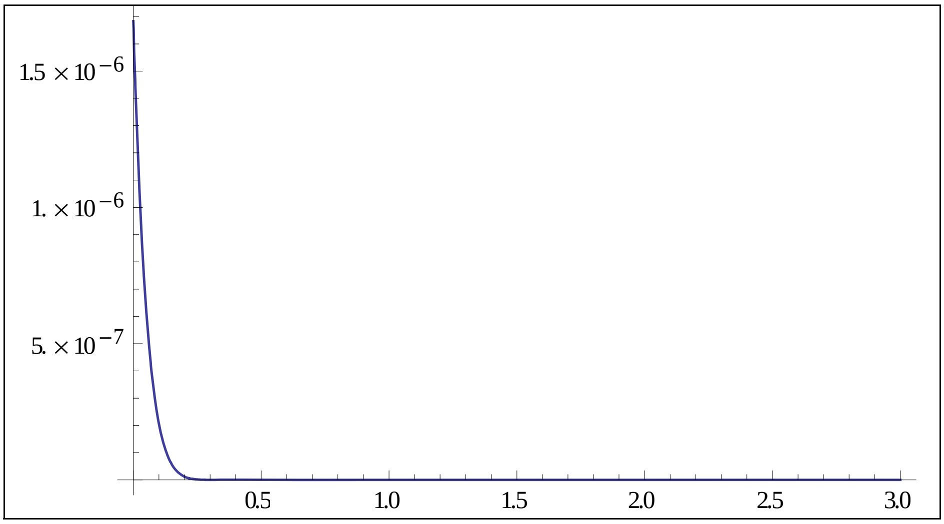


Figure 11. Graph of d for j = 0 and k_{max} = 20, whole plot range

Using the Method for an Extrapolation

The approximation function can be even used for an extrapolation outside the approximation interval $[t_0, t_{end}]$.

We consider the approximations function y_j for j = 0 and k_{max} = 15 from example 1 on the interval $[-2, 5]$:

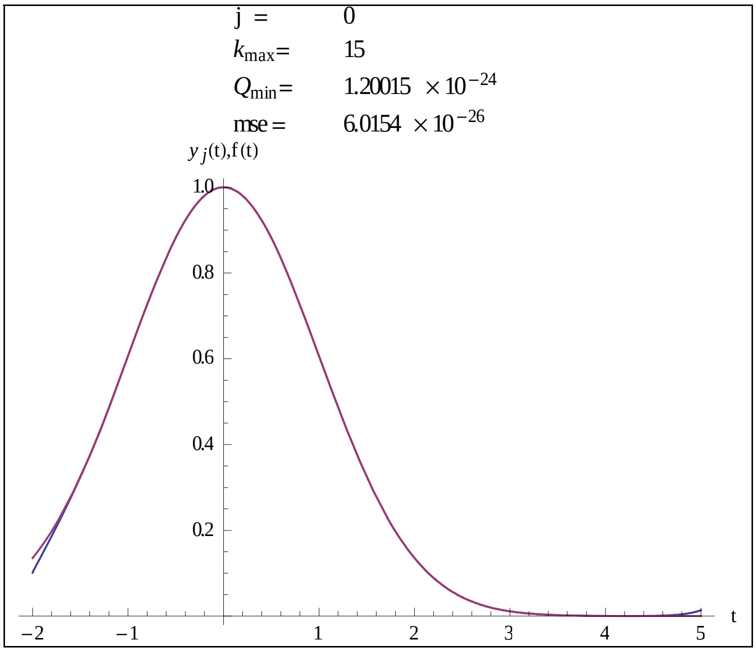


Figure 12. Extrapolation with y_j for j = 0 und k_{max} = 15

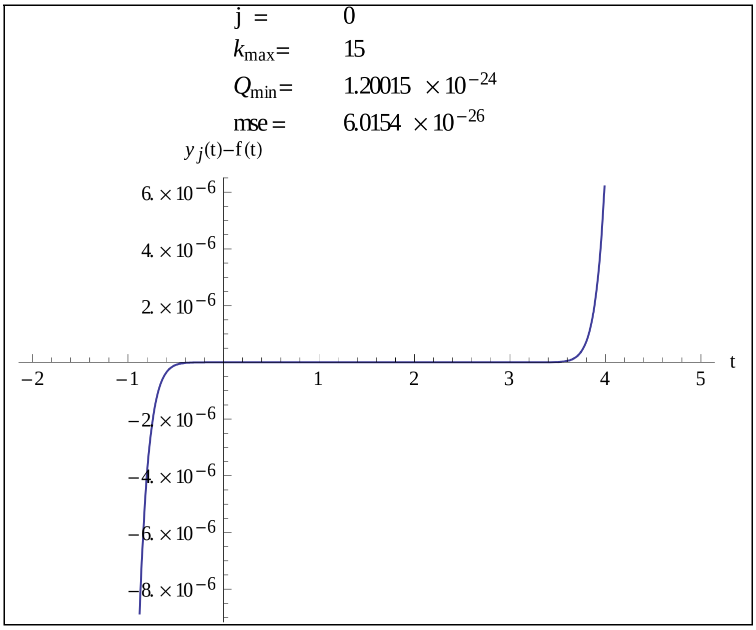


Figure 13. Graph of y_j - y for j = 0 und k_{max} = 15.

If we use in example 1 the Intervall $I = [-1, 1]$ with $h = 2/m$ and $m = 2k_{max}$, then we get the following graph of the approximation function y_j for j = 0 and k_{max} = 15:

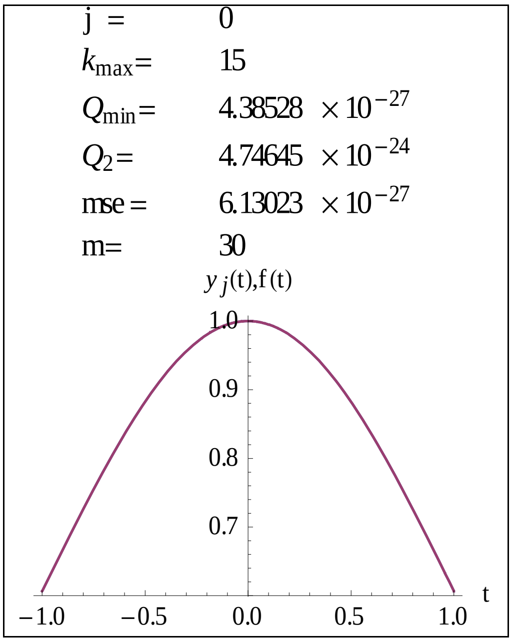


Figure 14. Graphs of y_j and y for j = 0 und k_{max} = 15

Here is the graph of the difference y_j - y:

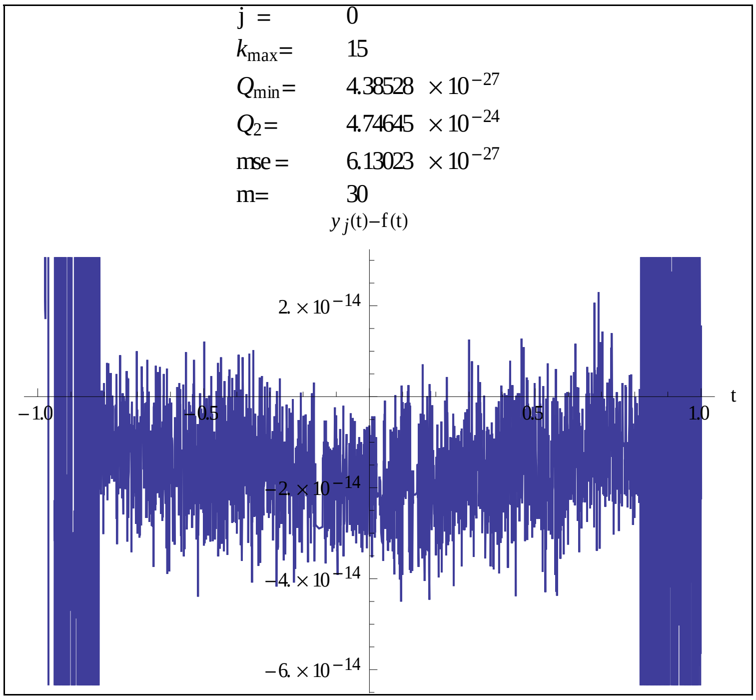


Figure 15. Graph of y_j - y for j = 0 und k_{max} = 15

Now we consider this approximation function on a bigger interval [-2, 2]:

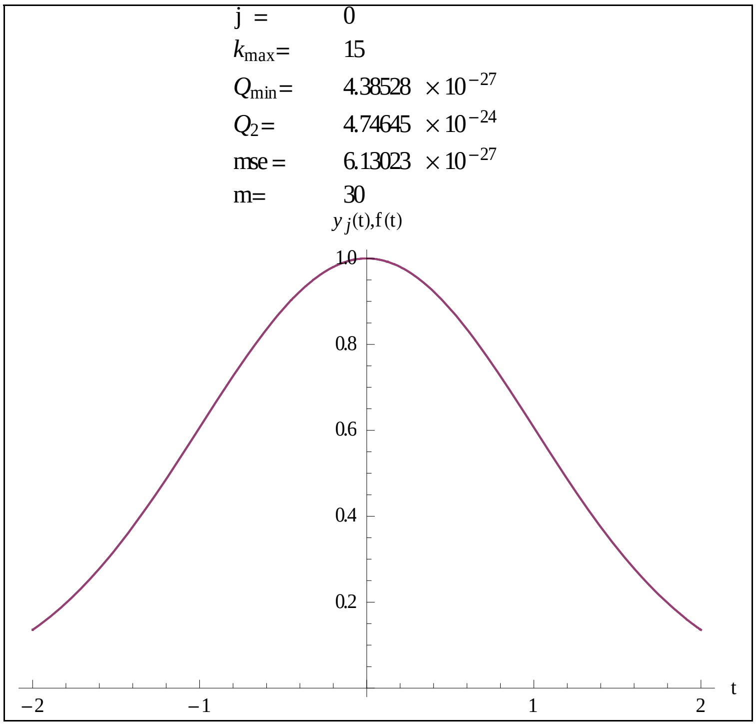


Figure 16. Graphs of y_j and y for j = 0 und k_{max} = 15 on *[-2, 2]*

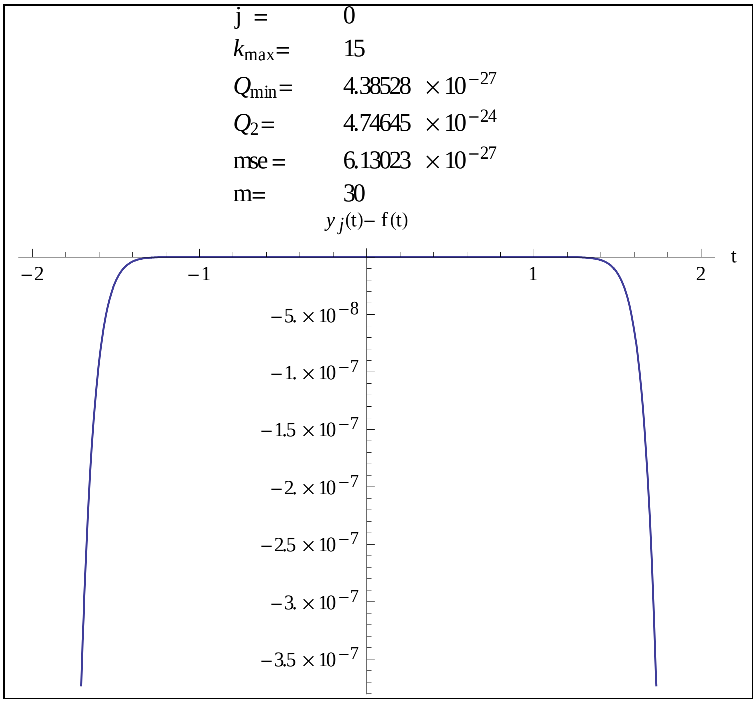


Figure 17. Graph of y_j - y for j = 0 und k_{max} = 15 on *[-2, 2]*

Here is the approximation function on a three times bigger interval than the approximation interval:

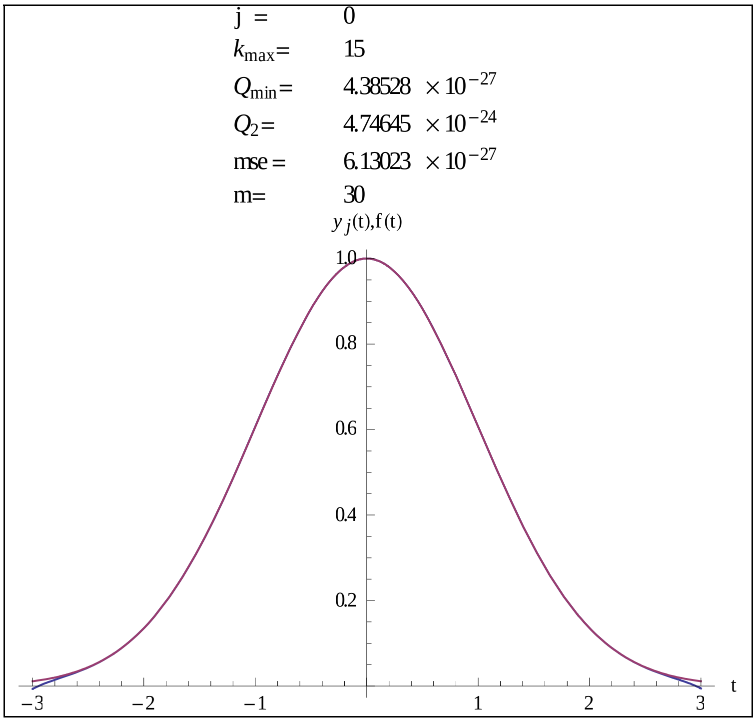


Figure 18. Graphs of y_j and y for j = 0 und k_{max} = 15 on *[-3, 3]*

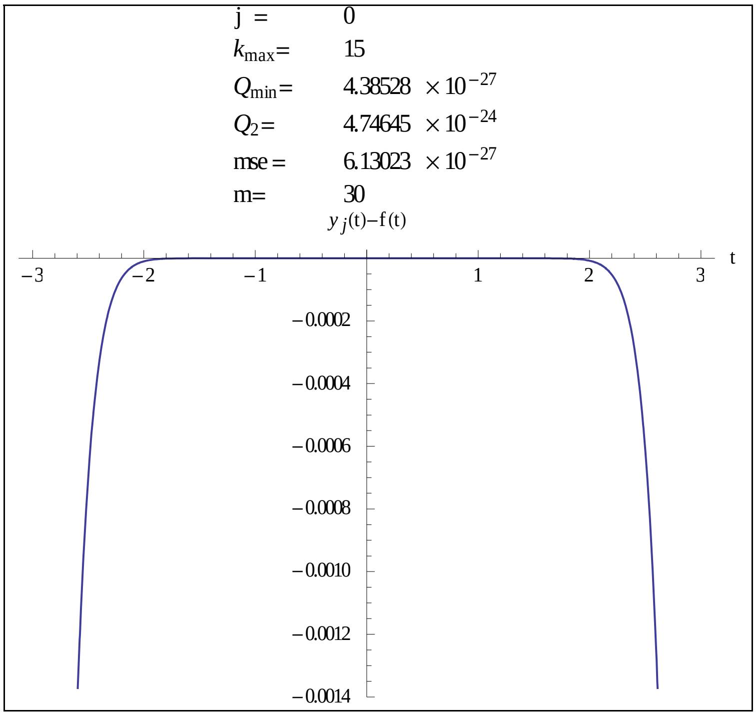


Figure 19. Graph of y_j - y for j = 0 und k_{max} = 15 on *[-3, 3]*

To what extent one can extrapolate the approximation depends on the number of coefficients c_k, that means from k_{max} (and generally from k_{min}, too) and from the from the width of the approximation interval I. Here k_{max} can not be chosen arbitrarily large, since only the coefficients c_k can be determined well in which $|\phi_{j,k}|$ is still sufficiently large or by using wavelets with compact support at all nonzero.

A Comparison of the Shannon Wavelet with the Daubechies Wavelet of Order 8

Example 3:
We solve approximately the problem of example 1 on $I = [-1, 1]$ and we minimize Q and use the collocation points $t_i = i \cdot h$ (with i = 1, 2, …,m, $m = c \cdot k_{max}$), with $h = 2/(c \cdot k_{max})$ and $k_{min} = -k_{max}$, k_{max} = 15, 20, 25, c = 1, 2, 3 and j = 0, 1, 2.

We use the Shannon wavelet and for a comparison the Daubechies wavelet of order 8.

If we use generally a Daubechies wavelet of order g with the approximation interval $I = [t_0, t_{end}]$ we can chose $k_{min}= 2^j\, t_0 - (2g-1)+1$ and $k_{max}=2^j\, t_{end} - 1$, because of the compact support of the Daubechies wavelet (otherwise $\phi_{j,k} = 0$ on I).

In example 3 we have g = 8. Two tables follow for comparing the results:

Daubechies wavelet:

j	m	k_{min}	k_{max}	$Qmin$	Q_2	mse
0	15.	−15	0.	4.35628×10^{-28}	0.00466786	8.43333×10^{-7}
0	30.	−15	0.	1.7124×10^{-8}	1.0587×10^{-6}	2.27548×10^{-11}
1	17.	−16	1.	1.00872×10^{-29}	0.29901	0.0000127997
1	34.	−16	1.	1.46341×10^{-8}	8.28391×10^{-7}	5.72889×10^{-11}
2	21.	−18	3.	6.71512×10^{-30}	246.608	0.141992
2	42.	−18	3.	3.57194×10^{-9}	0.0340748	1.89731×10^{-8}

Shannon:

j	m	k_{min}	k_{max}	Qmin	Q_2	mse
0	15	−15	15	8.23083×10^{-12}	8.38252×10^{-11}	3.9383×10^{-14}
0	30	−15	15	6.25902×10^{-27}	1.41041×10^{-24}	2.56371×10^{-27}
0	20	−20	20	1.23927×10^{-11}	6.93437×10^{-11}	1.85861×10^{-14}
0	40	−20	20	1.45232×10^{-26}	9.48622×10^{-25}	2.32583×10^{-27}
0	25	−25	25	2.2285×10^{-11}	7.84257×10^{-11}	1.78816×10^{-14}
0	50	−25	25	1.20324×10^{-26}	6.9879×10^{-25}	4.64306×10^{-27}
1	15	−15	15	2.99289×10^{-18}	6.02327×10^{-12}	3.72281×10^{-15}
1	30	−15	15	9.91007×10^{-30}	3.01769×10^{-25}	3.27535×10^{-29}
1	20	−20	20	4.97124×10^{-9}	8.27043×10^{-8}	1.56287×10^{-11}
1	40	−20	20	1.92217×10^{-27}	2.78109×10^{-24}	3.97155×10^{-28}
1	25	−25	25	8.75935×10^{-10}	1.37052×10^{-8}	6.55506×10^{-12}
1	50	−25	25	5.66866×10^{-27}	3.01211×10^{-23}	2.03996×10^{-27}
2	15	−15	15	2.17737×10^{-29}	0.000289405	1.76241×10^{-7}
2	30	−15	15	1.69325×10^{-28}	6.41826×10^{-18}	1.0083×10^{-21}
2	20	−20	20	9.6976×10^{-8}	0.0000230687	5.39382×10^{-9}
2	40	−20	20	7.86162×10^{-28}	1.77107×10^{-20}	1.3874×10^{-24}
2	25	−25	25	4.2941×10^{-8}	2.35033×10^{-6}	3.54143×10^{-10}
2	50	−25	25	3.08239×10^{-28}	2.38455×10^{-24}	6.62603×10^{-29}

Here are the Graphs of y_j and y, y_j - y and of y_j and y on a bigger interval (here [-3, 3]) for an extrapolation outside the approximation interval I an at least the graph of d. We start with the Daubechies wavelet:

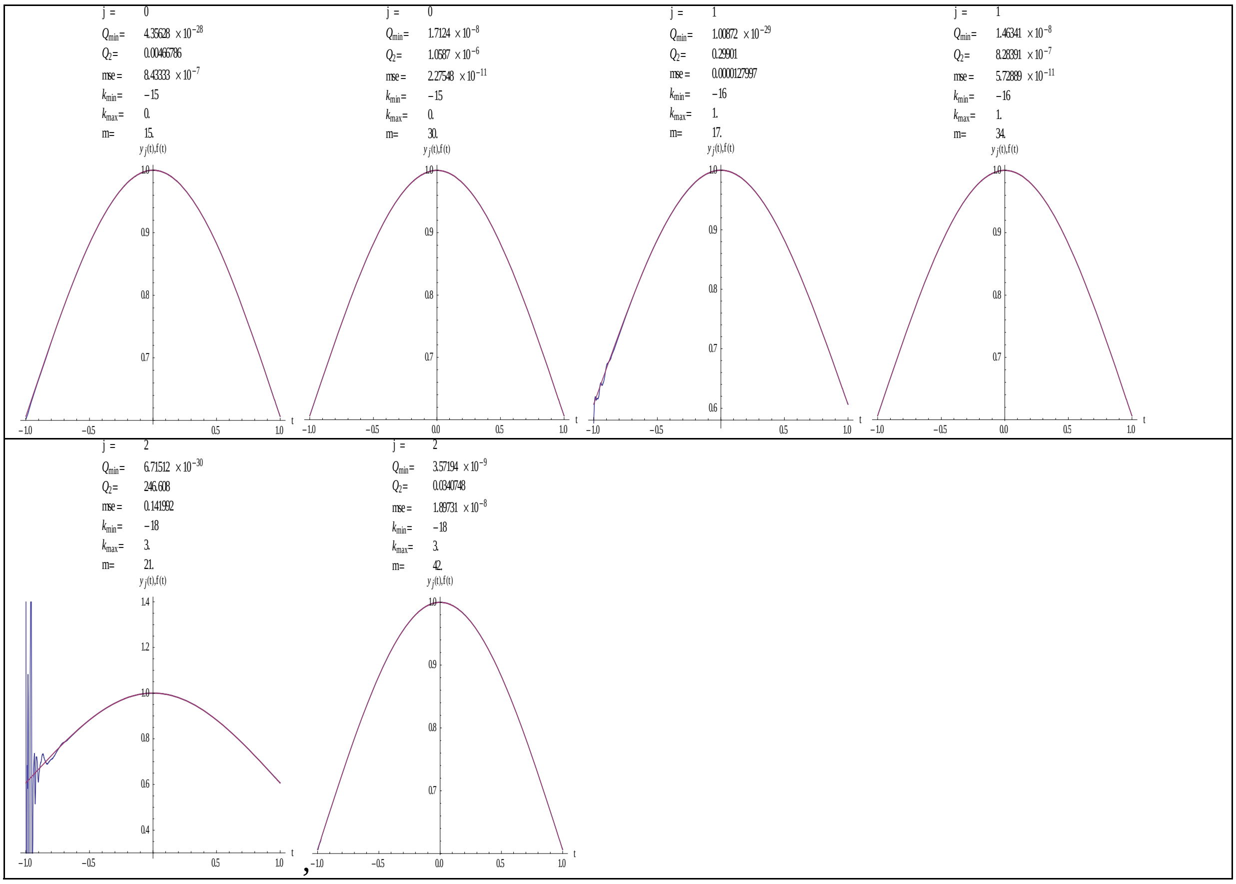


Figure 20. Graphs of y_j and y with the Daubechies wavelet

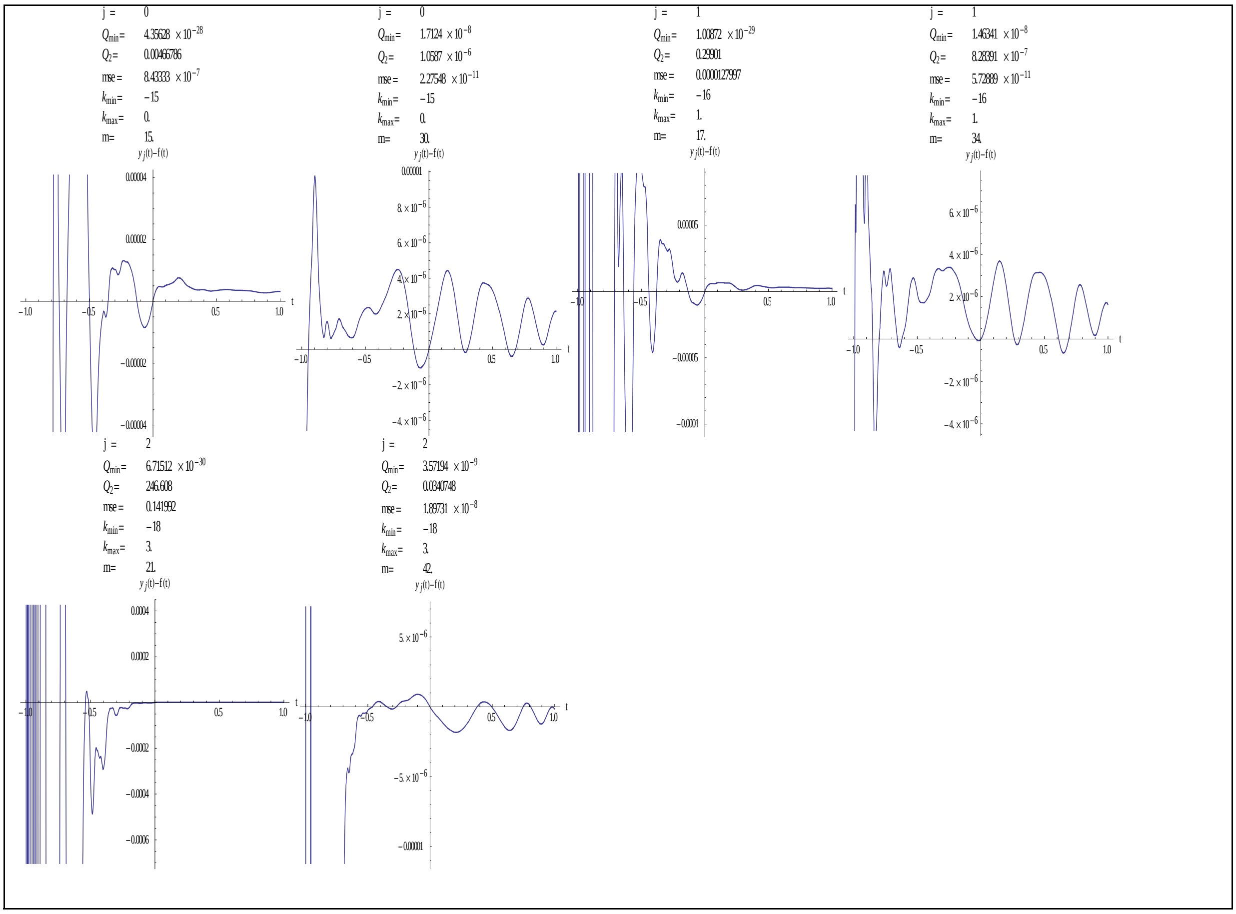


Figure 21. Graphs of y_j - y with the Daubechies wavelet

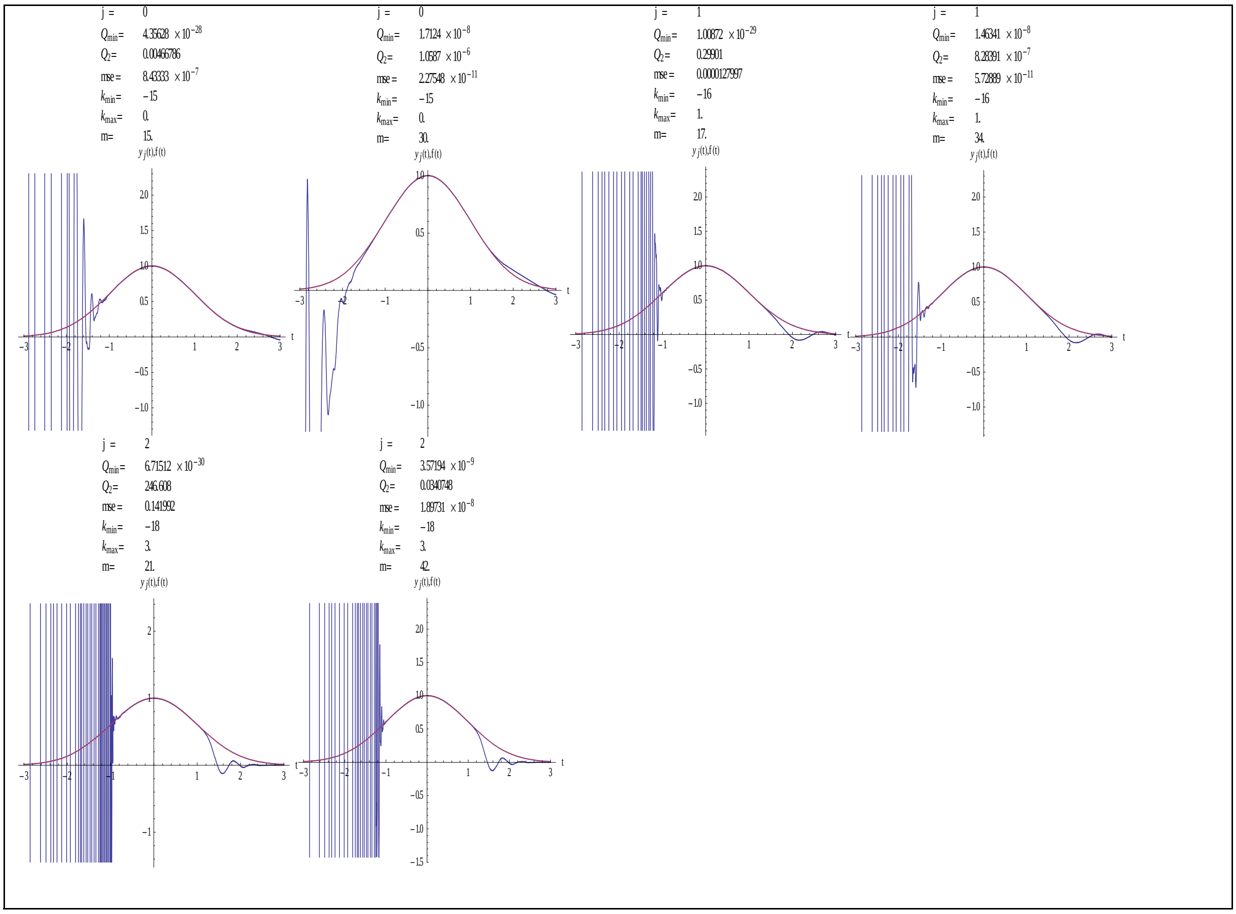


Figure 22. Graphs of y_j and y with the Daubechies wavelet on the interval [-3, 3]

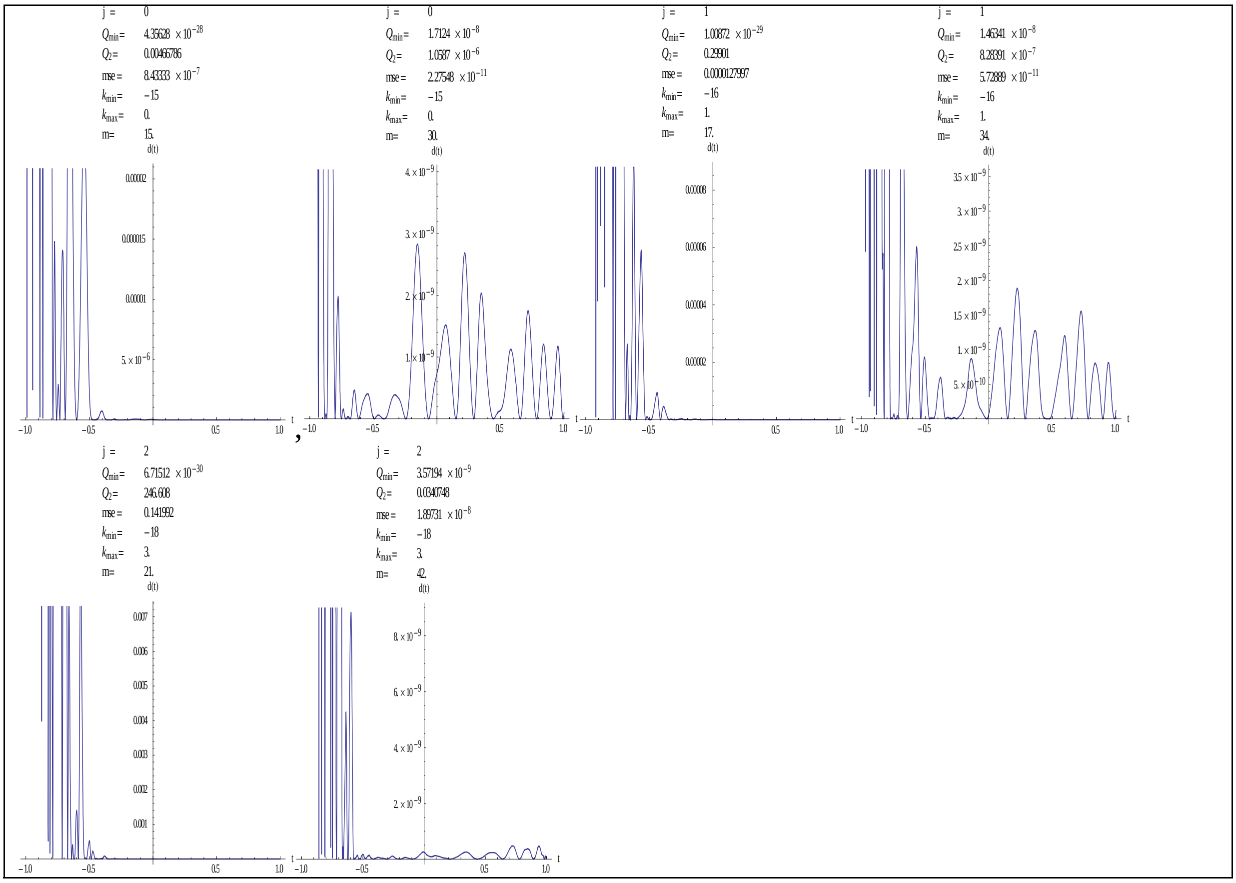
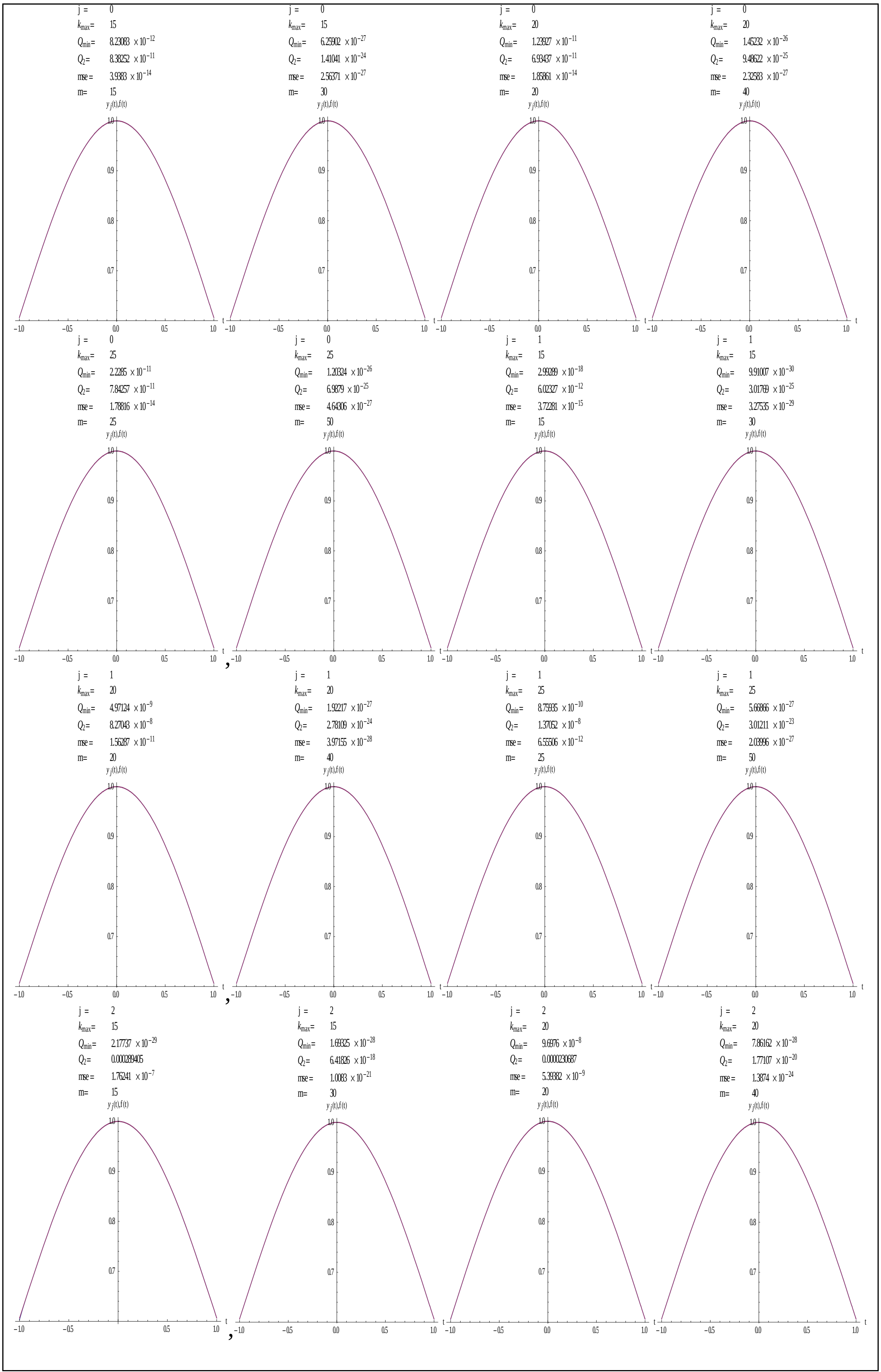


Figure 23. Graphs of d

Now the same curves for the Shannon wavelet:



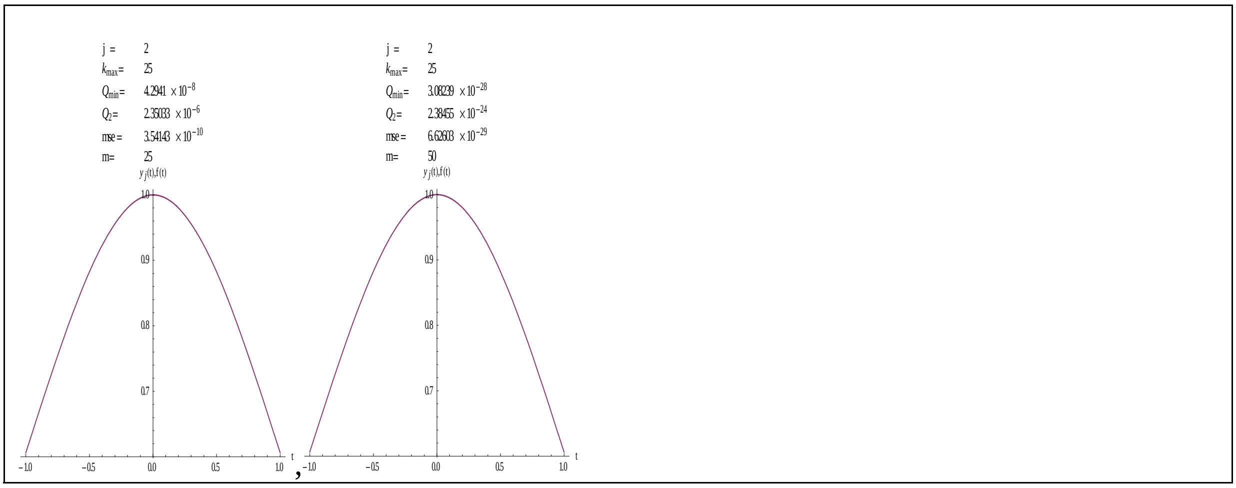
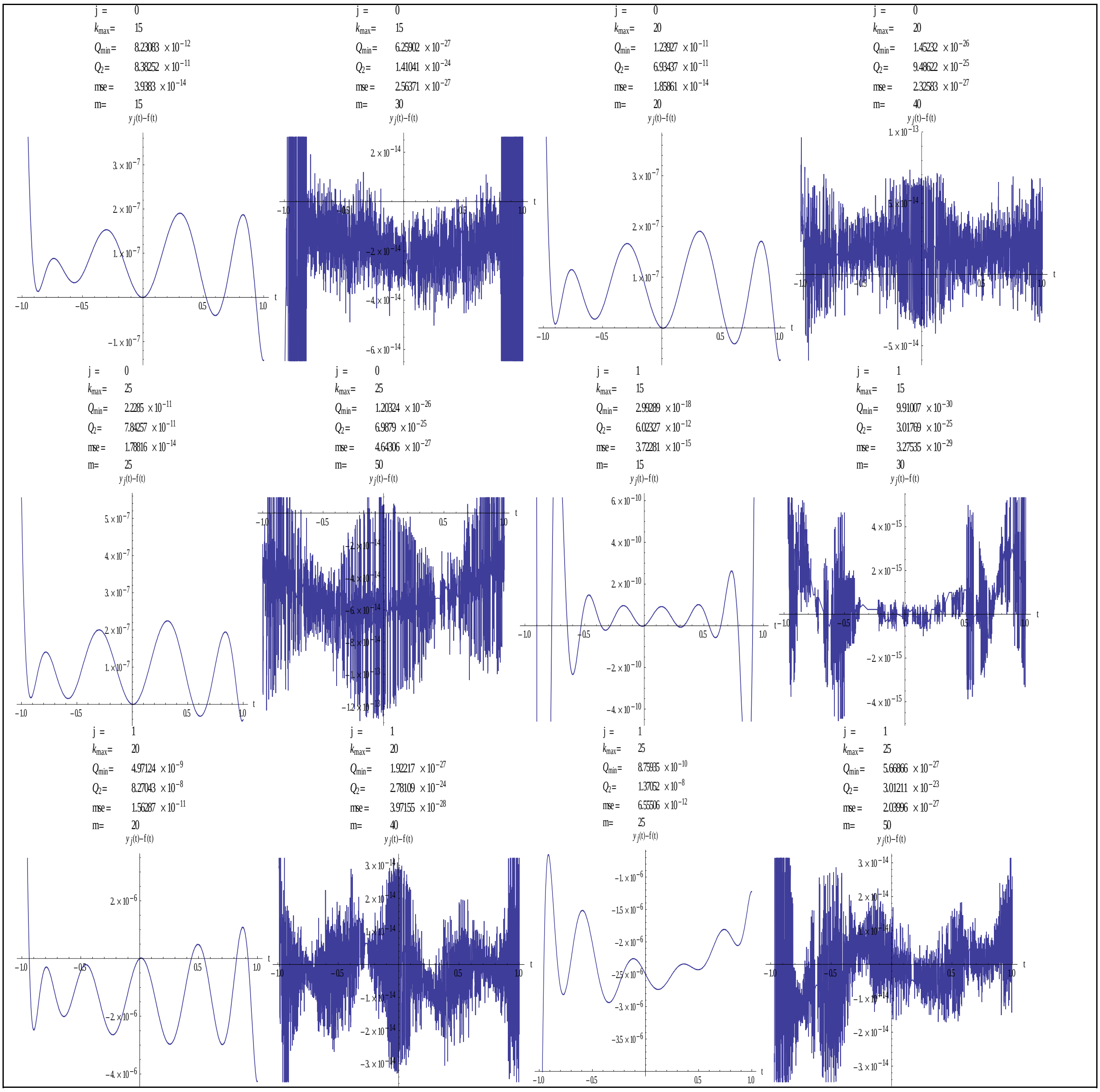


Figure 24. Graphs of y_j and y with the Shannon wavelet



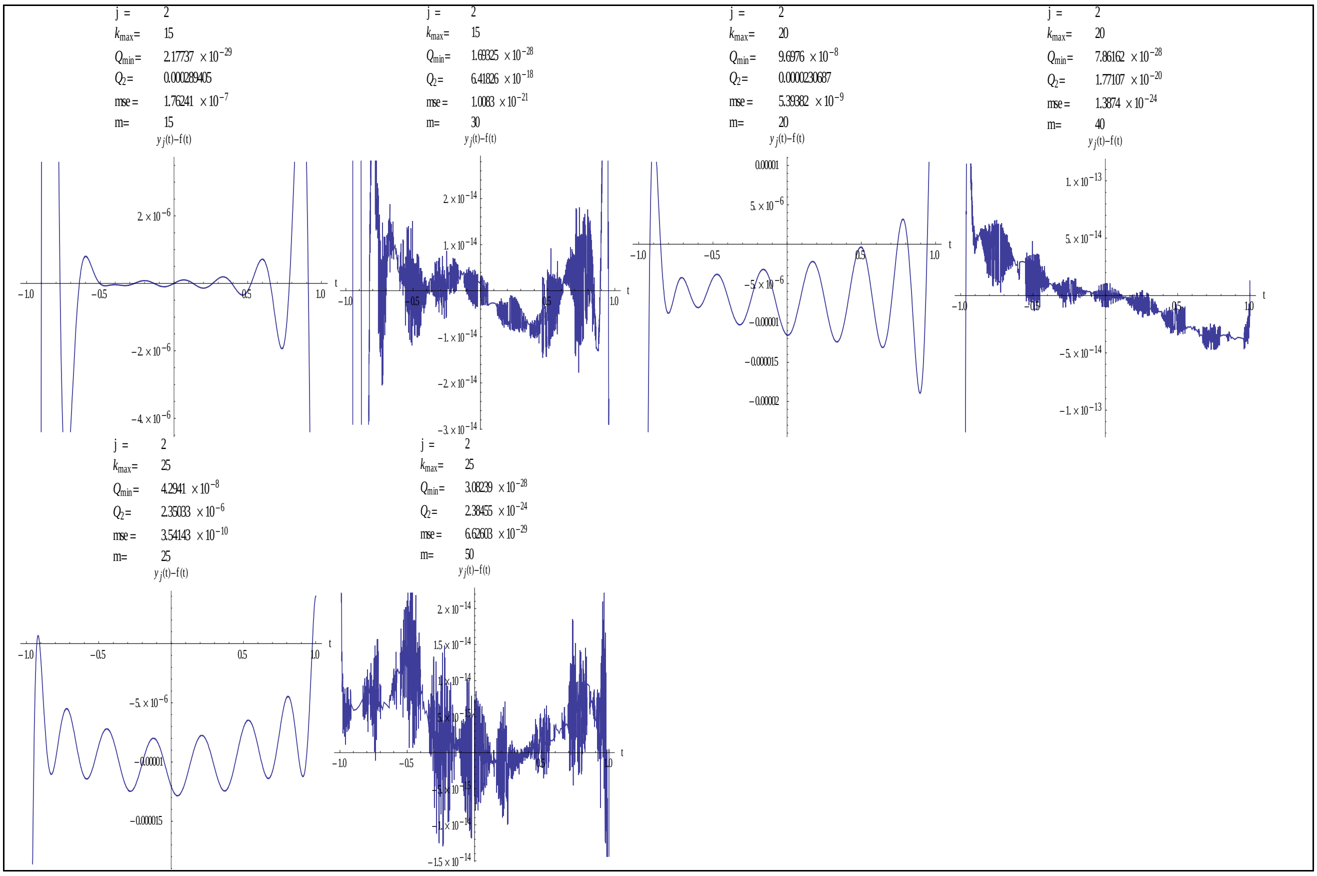
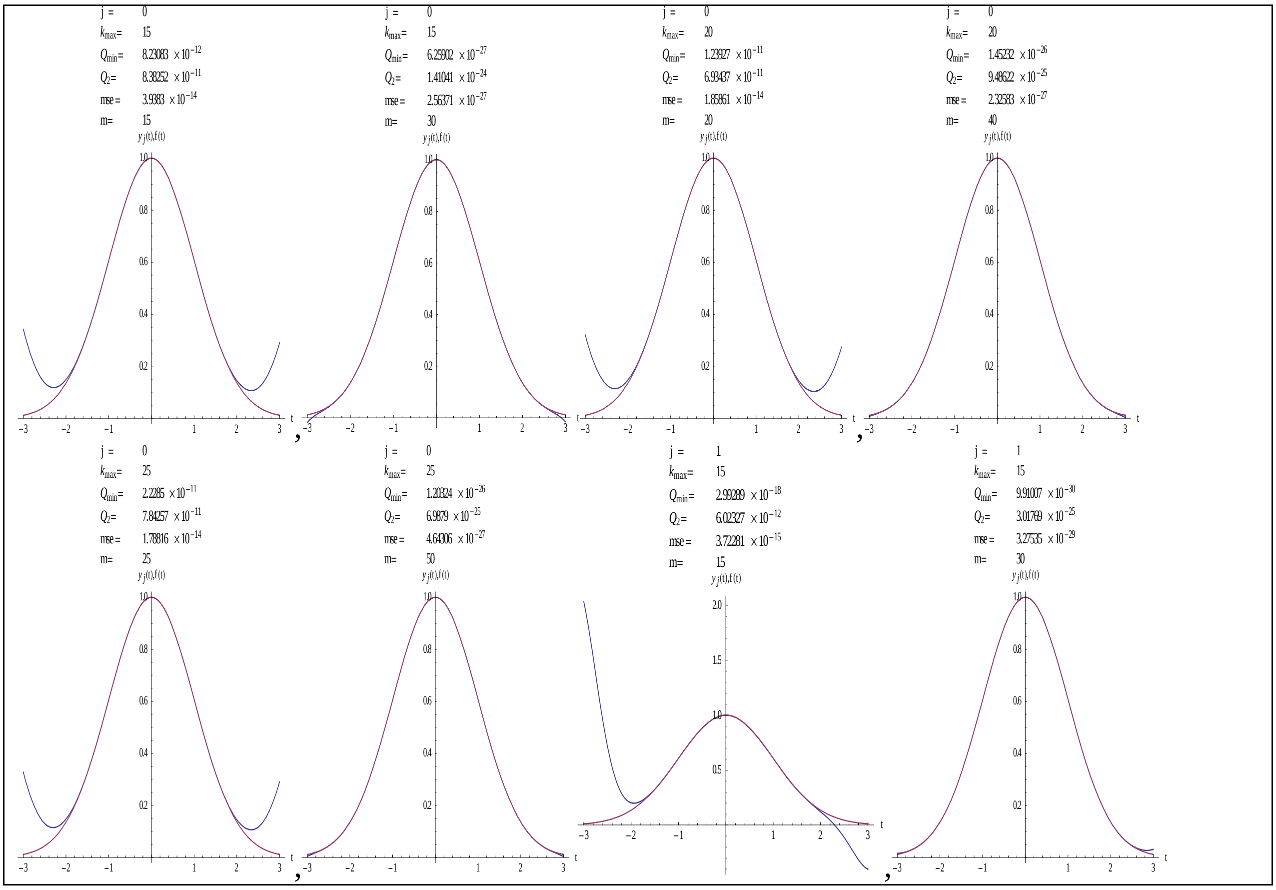


Figure 25. Graphs of y_j - y with the Shannon wavelet



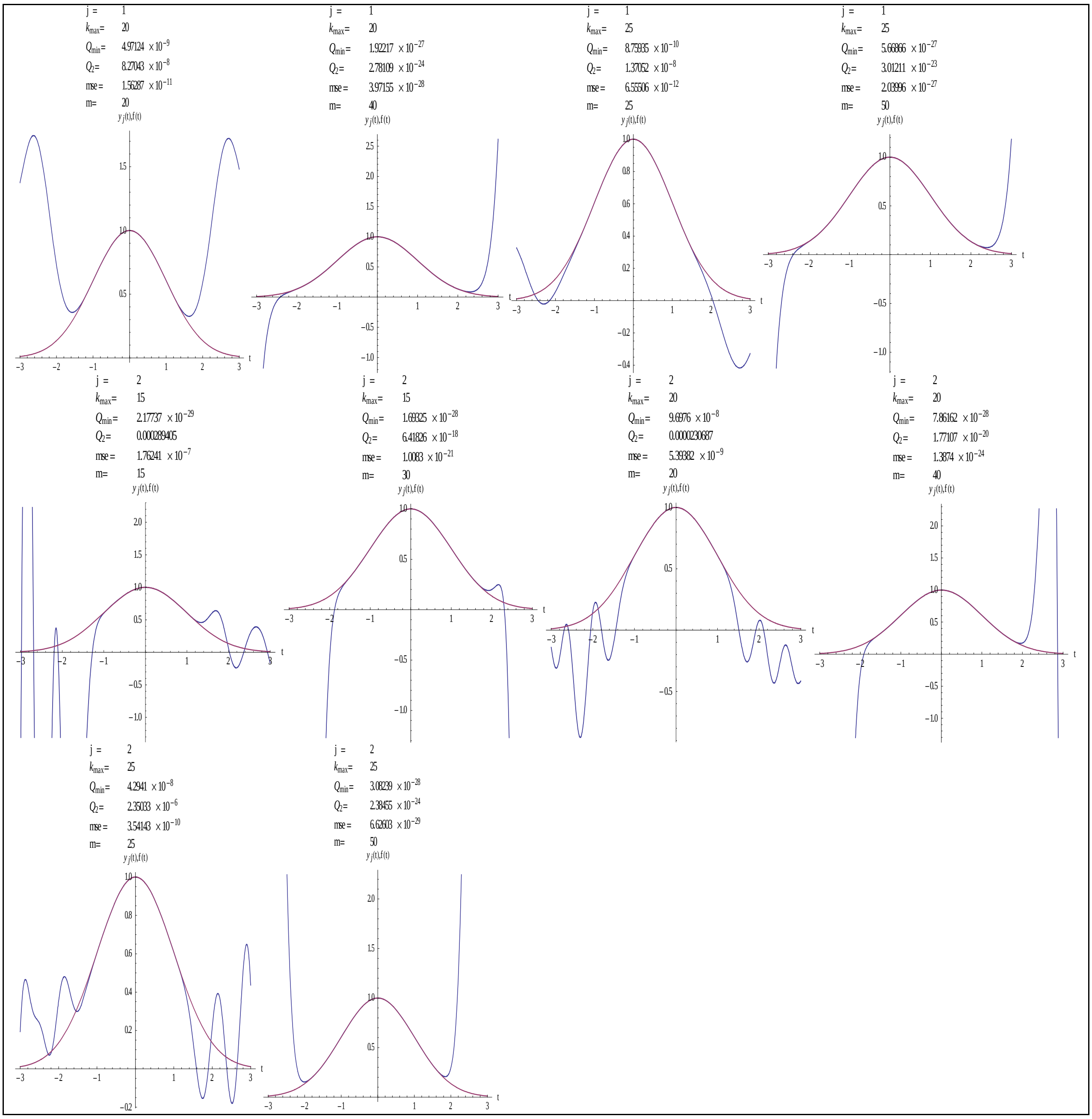
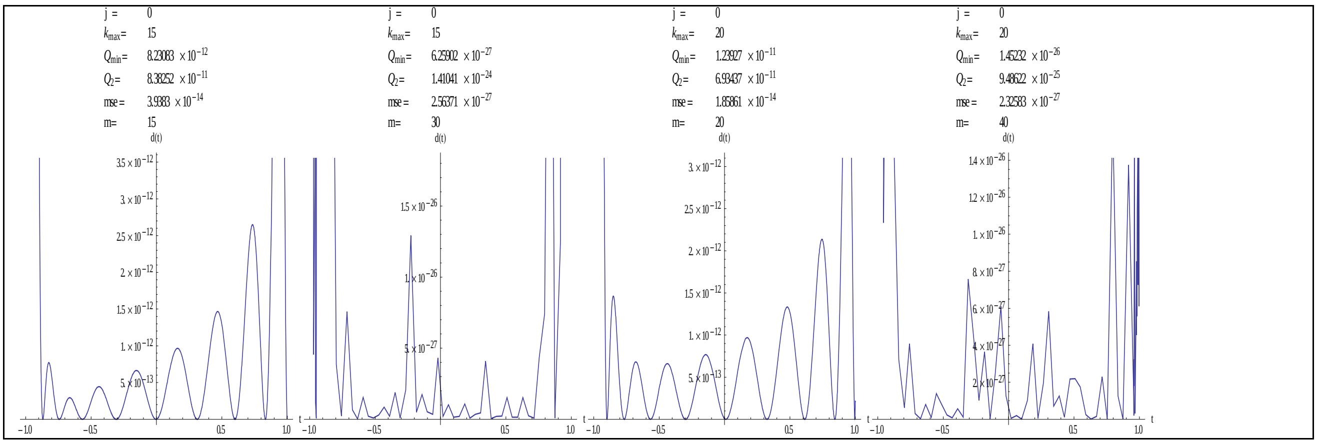


Figure 26. Graphs of y_j and y with the Shannon wavelet on the interval [-3, 3]



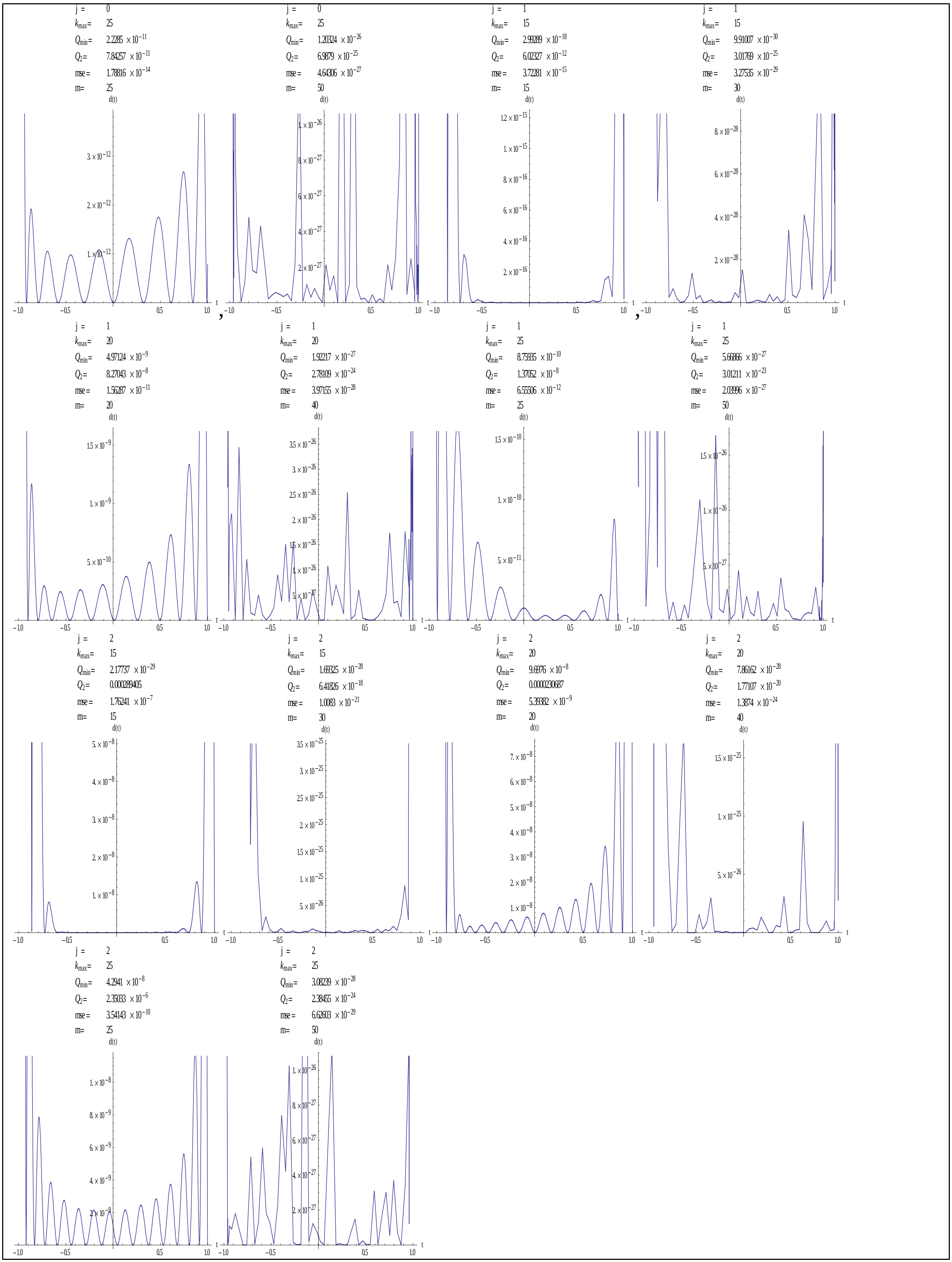


Figure 27. Graphs of d with the Shannon wavelet

The best extrapolation with the Daubechies wavelet:

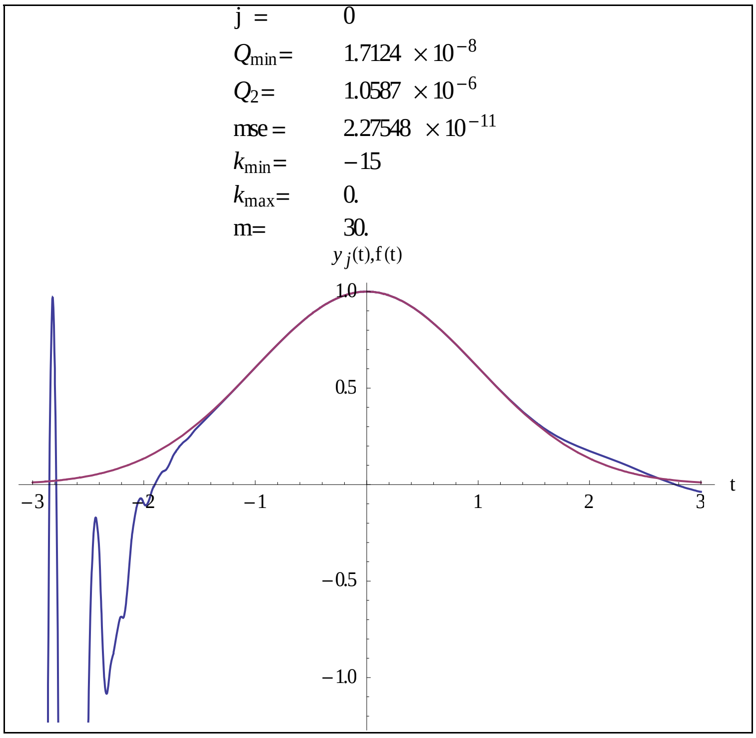


Figure 28. Best extrapolation with the Daubechies wavelet

An extrapolation with the Shannon wavelet:

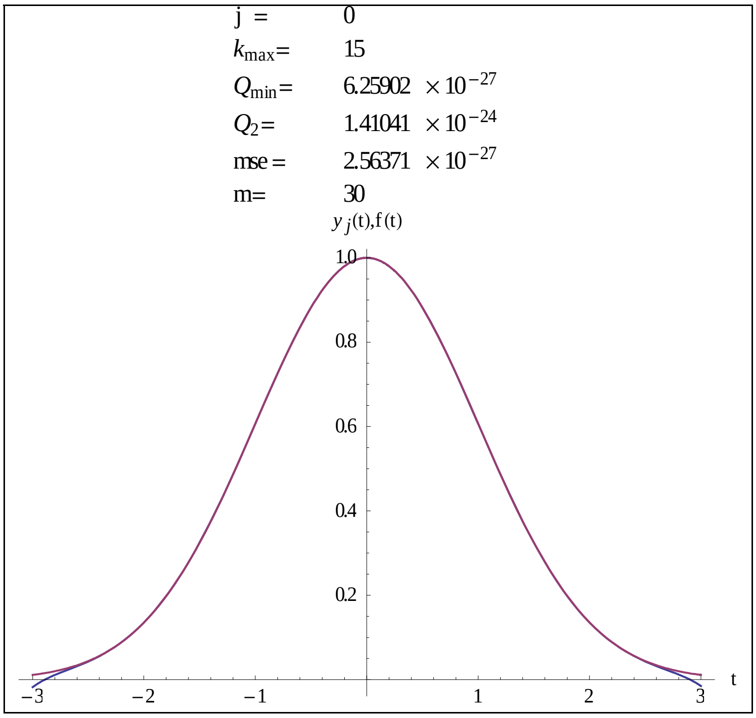


Figure 29. An extrapolation with the Shannon wavelet

With the Shannon wavelet be got much smaller *sse*'s and the extrapolation was much better, too. That is interesting, because the Shannon wavelet has no big order, but here we don't calculate an orthogonal projection on V_j. With the Daubechies wavelet we need less coefficients c_k. In many simulations we saw that we need a bigger j with the Daubechies wavelets (order 5, 7 and 8) to get an as good approximation as with the Shannon wavelet.

References

[1] S. Bertoluzza. *Adaptive wavelet collocation method for the solution of Burgers equation.* Transport Theory and Statistical Physics, 25: 3-5, (1996).

[2] T. S. Carlson, J. Dockery, J. Lund. *A Sinc-Collocation Method for Initial Value Problems.* Mathematics and Computation, Vol. 66, No. 217, (1997).

[3] D. L. Donoho. *Interpolating Wavelet Transforms*. Tech. Rept. 408. Department of Statistics, Stanford University, Stanford, (1992).

[4] E. Hairer, G. Wanner. *Vol. 1 : Nonstiff Problems.* Springer, 2. Ed., (1993).

[5] E. Hairer, G. Wanner. *Vol. 2 : Stiff and Differential-Algebraic Problems.* Springer, 2. Ed., (1996).

[6] L. Qian. *On the Regularized Whittaker-Koltel'nikov-Shannon Sampling Theorem.* Proceedings of the Amarican Mathematical Society, Vol. 131, No. 4, (2002).

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

[8] G. Strang. *Wavelets and Dilation Equations: A Brief Introduction.* SIAM Review Vil. 31, No. 4, (1989)

[9] M. Unser, T. Blu. *Comparison of Wavelets from the Point of View of their Approximation Error.* Proc. of SPIE Vol. 3458, Wavelet Applications in Signal and Image Processing, (1998)

[10] M. Unser (1996). *Vanishing moments and the approximation power of wavelet expansions.* Proceedings of the 1996 IEEE International Conference on Image Processing, (1996).

[11] R. Vuduc. *A Wavelet Collocation Method for Solving PDEs*. J. Comp. Phys., (2000).