Sparse Wavelet Representation through Iterated Interpolating Wavelets

Hyun Lim

I. INTERPOLATING BASIS

The starting point for the method are iterated interpolating functions. These are functions defined on the various levels of a dyadic grid. The coarsest level or level zero, is the base grid and it contains some number of grid points N_0. The next level, level one, includes the points on the base grid, and points half way between the points on the grid. The next level, level two, contain all the points from level one, and points half way between the points on level one. This is repeated indefinitely.

We shall take the notation $x_{j,k}$ to be the x-locations of the points at level j. k is an index labeling which point on level j, and it runs from 0 to $2^j N_0$. We shall denote the highest index at level j by $N_j = 2^j N_0$. If the index k is evenly divided by $2^{(j-m)}$, then the point in question also corresponds to a point on level m (for $m < j$). Another way of saying this, is that at a given level, only the points with an odd index appear at that level for the first time. All even points exist on one of the previous levels. We can write this as:

$$x_{j,k} = x_{j+m,2^m k} \tag{1}$$

To define a function on all the points at all the levels, it suffices to give value for the function at the base level. Given these initial values for the base grid, the corresponding points at all other levels are taken to be equal. That is, given the values $f(x_{0,k})$ we can write:

$$f(x_{j,2^j k}) = f(x_{0,k}) \tag{2}$$

This is just copying the values at level zero onto their corresponding higher level points. To fill in the missing values at each level, we use interpolation from the previous level. For the sake of discussion we will think of using a four point interpolation, but it is easy to generalize to higher order interpolation. To get value $f(x_{j,k})$ we need four values at level $j-1$:

$$f(x_{j,k}) = -\frac{1}{16}f(x_{j-1,(k-3)/2}) + \frac{9}{16}f(x_{j-1,(k-1)/2}) + \frac{9}{16}f(x_{j-1,(k+1)/2}) - \frac{1}{16}f(x_{j-1,(k+3)/2}) \tag{3}$$

Notice that for the points with index 1 and $N_j - 1$, there are not two points on either side, so we use the modified expression:

$$f(x_{j,1}) = \frac{5}{16}f(x_{j-1,0}) + \frac{15}{16}f(x_{j-1,1}) - \frac{5}{16}f(x_{j-1,2}) + \frac{1}{16}f(x_{j-1,3}) \tag{4}$$

$$f(x_{j,N_{j-1}}) = \frac{1}{16}f(x_{j-1,N_{j-1}-3}) - \frac{5}{16}f(x_{j-1,N_{j-1}-2}) + \frac{15}{16}f(x_{j-1,N_{j-1}-1}) + \frac{5}{16}f(x_{j-1,N_{j-1}}) \tag{5}$$

In this way, we can use the given values at level 0 to find the values on level 1. And with the values at level 1 we can find the values at level 2, and so on. At this point, we have a scheme for taking any collection of values at the base level and generating an interpolated version of that function to any desired level of our grid.

We can form a basis for this class of functions by marking out some special functions $\phi_{0,k}(x)$. These are defined such that at the base level, they have the values:

$$\phi_{0,k}(x_{0,l}) = \delta_{k,l} \tag{6}$$

Here $\delta_{k,l}$ is the Kronecker delta. The function takes a nonzero value only at one point of the base level. To fill out the values for the rest of the levels we use equations (2) - (5). Note that even though the function is only nonzero at one of the base level points, in general *these functions are nonzero at a large number of the higher level points.* Also note that nevertheless these functions do have compact support.

This set of functions forms a basis and any function defined by its values on the base level can be written in terms of them quite easily. Say we have some function f with values given at all the points on the base grid, and which has values interpolated to higher levels using the interpolation scheme outlined above. Given the values at the base level,

$$f(x_{0,k}) = f_{0,k}$$

we can write

$$f(x_{0,l}) = \sum_{k=0}^{N} f_{0,k}\phi_{0,k}(x_{0,l}) \tag{7}$$

The two sides of the previous expression match on the base grid:

$$\begin{aligned} f(x_{0,l}) &= \sum_{k=0}^{N} f_{0,k}\phi_{0,k}(x_{0,l}) \\ &= \sum_{k=0}^{N} f_{0,k}\delta_{k,l} = f_{0,l} \end{aligned}$$

Here we use equation (6) and the properties of the Kronecker delta. Since the expressions match on the base level, and the functional parts of each expression is propagated to higher levels via the interpolation scheme, these two expression give the same results. To close the discussion, we can see that these functions are independent of one another as each only has a single nonzero value on the base level. So, there is no way to write one of these functions in terms of the others, as it would be impossible to generate a nonzero by linear combination of zero values.

One thing worth noting about this basis is that the various basis functions in the interior are just shifted versions of another. More specifically,

$$\phi_{0,k}(x) = \phi_{0,m}(x + (x_{0,m} - x_{0,k})) \tag{8}$$

They have the same shape, but are translated in x. This does not occur at the boundary, because near the boundary there are different interpolation stencils being used.

II. MULTILEVEL BASIS

The procedure in the previous section can be extended. We can define a similar basis at every level. Take the function $\phi_{j,k}(x)$ to be defined so that at points at level j, the function is either zero or one:

$$\phi_{j,k}(x_{j,l}) = \delta_{k,l} \tag{9}$$

This function is completed by using interpolation to fill in values at higher levels, and by using equation (2) to fill in values at lower levels. The higher level basis functions also satisfy an equivalent translation property as in (8):

$$\phi_{j,k}(x) = \phi_{j,m}(x + (x_{j,m} - x_{j,k})) \tag{10}$$

Further, these functions satisfy a scaling relation as well:

$$\phi_{j,k}(2x) = \phi_{j+1,2k}(x) \tag{11}$$

So in general, the multilevel basis we have formed is a dilated translated version of a single abstract function $\phi(x)$:

$$\phi_{j,k}(x) = \phi(2^j x - x_{j,k}) \tag{12}$$

Of course, something difficult occurs on the boundary, but there is a similar recurrence between the boundary basis elements on one level with the boundary basis elements on higher levels.

Note one interesting property of these functions. If we consider a basis element with an odd position index, then the value of the function on all grid points at a lower level will be zero:

$$\phi_{j,k}(x_{g,m}) = 0 \text{ when } g < j \tag{13}$$

This is because the one nonzero value at level j will be for a location that is not in any of the other lower level grids.

Another interesting property of this multilevel basis is that the basis at level j can be written in terms of the basis at the $j+1$:

$$\phi_{j,k} = \sum_{l=0}^{N_{j+1}} h_{j,k}^{j+1,l} \phi_{j+1,l}(x) \tag{14}$$

To make this an equality at that is needed is to pick h appropriately. As in the previous section, if we can get the two sides to agree on the level $j+1$, they will agree on all subsequent levels.

$$\begin{aligned} \phi_{j,k}(x_{j+1,m}) &= \sum_{l=0}^{N_{j+1}} h_{j,k}^{j+1,l} \phi_{j+1,l}(x_{j+1,m}) \\ &= \sum_{l=0}^{N_{j+1}} h_{j,k}^{j+1,l} \delta_{l,m} = h_{j,k}^{j+1,m} \end{aligned}$$

The h coefficients are just values of the j level basis on the near by $j+1$ level points, which are either known from (9) or can be found using the interpolation scheme. Note that the vast majority of these values are zero, and it is only a few points near the index $2k$ (on level $j+1$) that will contribute. For particular case of the four point interpolation we have This procedure can be iterated. The basis at any level can be written in terms of the basis at any higher level. Indeed, there was nothing special in the previous expressions about the difference in levels being 1. For any two levels j and g, where $g > j$:

m	$h_{j,k}^{j+1,m}$
$2k-4$	0
$2k-3$	-1/16
$2k-2$	0
$2k-1$	9/16
$2k$	1
$2k+1$	9/16
$2k+2$	0
$2k+3$	-1/16
$2k+4$	0

$$\phi_{j,k}(x) = \sum_{l=0}^{N_g} h_{j,k}^{g,l} \phi_{g,l}(x) \tag{15}$$

where

$$h_{j,k}^{g,l} = \phi_{j,k}(x_{g,l}) \tag{16}$$

Finally, it it possible to relate the multilevel connection to a product of single level connections, by using (14):

$$\phi_{j,k}(x) = \sum_{l_1=0}^{N_{j+1}} h_{j,k}^{j+1,l_1} \sum_{l_2=0}^{N_{j+2}} h_{j+1,l_1}^{j+2,l_2} \cdots \sum_{l_n=0}^{N_{j+n}} h_{j+n+1,l_{n-1}}^{j+n,l_n} \phi_{j+n,l_n}(x) \tag{17}$$

Each of h factors in the nested summation are easy to compute from the previous table (or similar set of numbers for a different interpolation scheme), but the work involved can be immense.

III. INTERPOLATING WAVELET BASIS

We have seen in the previous two sections a complete basis for each level independently, and that the basis on one level can be written in terms of the basis at a higher level. This actually is a ver nice mechanism for representing complicated functions on a very high level (that is, very fine grid) using an economy of points.

To see how, consider a basis made from following functions:

$$\{\phi_{0,k}, \psi_{j,k}\} \tag{18}$$

Here

$$\psi_{j,k} = \phi_{j+1,2k+1} \tag{19}$$

and j takes the values from 0 to $J-1$, where J is some specified maximum refinement level of the grid. The ψ function are those from the multilevel basis that live not on the base grid. This set of functions acts as a basis for the finest level grid available. Each point of the finest grid can be associated with a single function from the given set. Note that even though $\psi_{j,k}$ is centered on a point at level $j+1$, we nevertheless index ψ function at level j.

Thus, we can write any function on the finest grid as a weighted sum of these functions:

$$f(x) = \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x) + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \tag{20}$$

The coefficient in the first term are just the values of the function at the base grid. This is easy to verify by plugging the base grid locations into (20):

$$\begin{aligned}
f(x_{0,l}) &= \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{0,l}) + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x_{0,l}) \\
&= \sum_{k=0}^{N_0} f_{0,k}\delta_{k,l} + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x_{0,l}) \\
&= f_{0,l} + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x_{0,l}) \\
&= f_{0,l} + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\phi_{j+1,2k+1}(x_{0,l}) \\
&= f_{0,l} + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\cdot 0 \\
&= f_{0,l}
\end{aligned}$$

We have used, in order, (6), the definition of the Kronecker delta, (19) and (13).

To derive the coefficients $d_{j,k}$, we can proceed recursively. Start with a point at level 1:

$$\begin{aligned}
f(x_{1,l}) &= \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x_{1,l}) \\
f_{1,l} &= \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) + \sum_{j=0}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\phi_{j+1,2k+1}(x_{1,l}) \\
&= \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) + \sum_{j=0}^{J-1}\sum_{k=0}^{N_0-1} d_{0,k}\phi_{j+1,2k+1}(x_{1,l}) \\
&= \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) + \sum_{j=0}^{J-1}\sum_{k=0}^{N_0-1} d_{0,k}\delta_{2k+1,l} \\
&= \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) + d_{0,(l-1)/2} \qquad (21)
\end{aligned}$$

In the first step, we use the notational shorthand $f(x_{j,k}) = f_{j,k}$ and the definition (19). In the second step, we use (13) to eliminate all the terms in the second summation that vanish. Finally, we use (9), and the Kronecker delta to eliminate all the rest of the terms. We find:

$$d_{0,(l-1)/2} = f_{1,l} - \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) \tag{22}$$

This can be simplified slightly by using (14) in the second term, and then applying (9):

$$\begin{aligned}
d_{0,(l-1)/2} &= f_{1,l} - \sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x_{1,l}) \\
&= f_{1,l} - \sum_{k=0}^{N_0} f_{0,k} \sum_{m=0}^{N_1} h_{0,k}^{1,m}\phi_{1,m}(x_{1,l}) \\
&= f_{1,l} - \sum_{k=0}^{N_0} f_{0,k} \sum_{m=0}^{N_1} h_{0,k}^{1,m}\delta_{m,l} \\
&= f_{1,l} - \sum_{k=0}^{N_0} f_{0,k} \sum_{m=0}^{N_1} h_{0,k}^{1,l}
\end{aligned}$$

For the case of a four point interpolation the last sum expands out to give (see the table of h values before, and not that the summation is over locations in the base level:

$$\begin{aligned}
d_{0,(l-1)/2} &= f_{1,l} - f_{0,(l-3)/2}h_{0,(l-3)/2}^{1,l} - f_{0,(l-1)/2}h_{0,(l-1)/2}^{1,l} - f_{0,(l+1)/2}h_{0,(l+1)/2}^{1,l} - f_{0,(l+3)/2}h_{0,(l+3)/2}^{1,l} \\
&= f_{1,l} - \left(-\frac{1}{16}f_{0,(l-3)/2} + \frac{9}{16}f_{0,(l-1)/2} + \frac{9}{16}f_{0,(l+1)/2} - \frac{1}{16}f_{0,(l+3)/2}\right)
\end{aligned}$$

We recognize this last sum as exactly the formula for interpolating from level 0 to level 1. So the coefficients $d_{0,(l-1)/2}$ are just the difference between the actual value at level 1 points and the interpolated values at level 1 based on the level 0 values. For brevity, we will on occasion call the term in parentis $P_{1,l}$, the interpolation to point l at level 1 from the points at the previous level. We can even extend this notation to other levels:

$$P_{j,k}(f) = -\frac{1}{16}f_{j-1,(k-3)/2} + \frac{9}{16}f_{j-1,(k-1)/2} + \frac{9}{16}f_{j-1,(k+1)/2} - \frac{1}{16}f_{j-1,(k+3)/2} \tag{23}$$

Or equivalent expression for other interpolation schemes. So, this leads to the terse expression:

$$d_{0,(l-1)/2} = f_{1,l} - P_{1,l}(f) \tag{24}$$

A. The Nested Basis

The last result is important. Because the $d_{0,k}$ coefficients are just the difference between the value at the level 1 points and the result of the interpolations of the level 0 at that location, we can always recombine the sum in (20) to start at level 1.

We will treat each term separately. We start with the sum over the base level functions:

$$\begin{aligned}
\sum_{k=0}^{N_0} f_{0,k}\phi_{0,k}(x) &= \sum_{k=0}^{N_0} f_{0,k} \sum_{l=0}^{N_1} h_{0,k}^{1,l}\phi_{1,l}(x) \\
&= \sum_{k=0}^{N_0} f_{1,2k} \sum_{l=0}^{N_1} h_{0,k}^{1,l}\phi_{1,l}(x) \\
&= \sum_{k=0}^{N_0} f_{1,2k} \sum_{l=0}^{N_1} h_{0,k}^{1,2l}\phi_{1,2l}(x) + \sum_{k=0}^{N_0} f_{1,2k} \sum_{l=0}^{N_0-1} h_{0,k}^{1,2l+1}\phi_{1,2l+1}(x) \\
&= \sum_{k=0}^{N_0} f_{1,2k}\phi_{1,2k}(x) + \sum_{k=0}^{N_0} f_{0,k} \sum_{l=0}^{N_0-1} h_{0,k}^{1,2l+1}\phi_{1,2l+1}(x)
\end{aligned} \tag{25}$$

In the first step, we have used this the fact that the value of the function at an even level 1 point is the same as the equivalent point at level 2. In the second step, we have written the even terms form the inner sum separate from the odd terms. The even terms result always only result in a single nonzero h. This is seen in the provided values for the four point case, but is a general feature of the interpolating schemes. Further, this single value is always 1, hence the last line.

Now for the second term in (20), the basic idea is to split the sum over levels into the zero level contribution and the rest. We will leave the rest untouched, but we will plug in our result for $d_{0,l}$ and find some happy cancellations.

$$\begin{aligned}
\sum_{j=0}^{J-1}\sum_{k=0}^{N_j} d_{j,k}\psi_{j,k}(x) &= \sum_{k=0}^{N_0-1} d_{0,k}\psi_{0,k}(x) + \sum_{j=1}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \\
&= \sum_{k=0}^{N_0-1} d_{0,k}\phi_{1,2k+1}(x) + \sum_{j=1}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \\
&= \sum_{k=0}^{N_0-1}\left(f_{1,2k+1} - \sum_{w=0}^{N_0} f_{0,w}h_{0,w}^{1,2k+1}\right)\phi_{1,2k+1}(x) + \sum_{j=1}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \\
&= \sum_{k=0}^{N_0-1} f_{1,2k+1}\phi_{1,2k+1}(x) - \sum_{w=0}^{N_0} f_{0,w}\sum_{k=0}^{N_0} h_{0,w}^{1,2k+1}\phi_{1,2k+1}(x) + \sum_{j=1}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \qquad (26)
\end{aligned}$$

The manipulations here are: we separate the $j = 0$ terms from the sum; we use (19); we plug in our expression for $d_{0,l}$; and we rearrange terms and move summations past factors that do not depend on the index being summed over.

Here, the second term in (25) will exactly cancel the second term from (26). Adding the two terms together will result in:

$$\begin{aligned}
f(x) &= \sum_{k=0}^{N_0} f_{1,2k}\phi_{1,2k}(x) + \sum_{k=0}^{N_0-1} f_{1,2k+1}\phi_{1,2k+1}(x) + \sum_{j=1}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \\
&= \sum_{k=0}^{N_1} f_{1,k}\phi_{1,k}(x) + \sum_{j=1}^{J-1}\sum_{k=0}^{N_j-1} d_{j,k}\psi_{j,k}(x) \qquad (27)
\end{aligned}$$

Equation (27) is essentially identical to the original expansion in the multilevel basis (20). The upshot of this is that we can now easily get the $d_{1,k}$ coefficients. They will be, by analogy with (24), $d_{1,k} = f_{2,2k+1} - P_{2,2k+1}(f)$.This process can then repeat and will yield the following:

$$d_{j,k} = f_{j+1,2k+1} - P_{j+1,2k+1}(f) \qquad (28)$$

B. Evaluating the function

What we have seen is that there are two expressions for our function. One is the collection of function values at all the points, and the other is the function values at the base level with $d_{j,k}$ values at the points in between. These $d_{j,k}$ values are not the function values, but we can use them to recreate the function values. This involves more or less rearranging (28) for the values of the functions at the higher levels.

Notice that in the above, we only ever needed to use a few values of the basis functions. This is because the iterated interpolation property of the way that each basis function is constructed creates the details at the finer levels from interpolation from the coarser levels. It cannot be emphasized enough that this is not the same as saying that the functions only have non-zero values at a few locations. Each and every basis function, at each level, has a large number of nonzero values. The special way that we construct them allows us to use a few values to generate the others, but this is not the same as these functions only being defined in a few places.

C. The Sparse Representation

Now that we are able to compute the $d_{j,k}$ coefficients given some function $f(x)$, we can now represent that function with fewer points than exist on the finest level.

You can think (20) as a zero level contribution and a series of corrections. As we have seen, the $d_{j,k}$ are the difference between the function value and the interpolated value. If this coefficient at a certain point is near to zero, then if we are willing to accept a certain level of error in our representation, we can ignore that coefficient, because the interpolated value will be close enough for our needs. This is the essence of the sparse representation of a function using this method.

If we chose some tolerance ϵ, we can group the terms in (20) according to whether $|d_{j,k}| < \epsilon$ or not. For those terms that are less than ϵ, we can ignore them in our sum, and we will pick up an error there that is on the order of ϵ. Each of terms in (20) are naturally associated with a grid point. The $f_{0,k}$ terms are associated with points on the base level, and the $d_{j,k}$ terms are associated with points on higher levels. When throwing our d terms that are too small, you can thus throw out points of the grid. This can be a huge savings computationally.

But we should be careful. Just because we do not have to store extra information at points where $|d_{j,k}| < \epsilon$, does not mean there is no value at that location. Indeed, the basis elements of all the other terms that will overlap the missing point will contribute to the value of the function represented by the expression (20). So really the lack of one of the locations in the grid is really a statement that all the relevant information at that location is covered by other functions. Indeed, because of all of the base level points will always be in the grid, there will be a number of the base level basis functions that will contribute at every location in the grid. Further, there is no guarantee that the points are at the special locations given in the table, so many values of the basis functions will be needed. These can be given by repeatedly applying the values in the table, like in (17), but the resulting value will not be zero in general

IV. DERIVATIVES

It is often necessary to take derivatives of the function we represent using (20). One way to do this is just use the function values at each location and do standard finite differencing. Another that uses the expansion would be to take the derivatives of the basis elements, and use that to take the derivative. This expansion would look like:

$$\frac{df}{dx} = \sum_{k=0}^{N_0} f_{0,k} \frac{d\phi_{0,k}}{dx} + \sum_{j=0}^{J-1} \sum_{k=0}^{N_j - 1} d_{j,k} \frac{d\psi_{j,k}}{dx} \tag{29}$$

If it were possible to compute the derivatives of the basis functions in some way, we would be set. Such expressions exist, and the values can be found in other documents on the C-SWARM wiki, as well as some graph of what these derivative functions look like.

Right away we can see that some of the special properties of the wavelet basis are not longer true for the derivative of the basis. When the wavelet basis functions are zero, the derivative is not necessarily zero. Indeed, at all the zeros of the wavelet basis in the interesting part of the function, the derivative is nonzero. So the property (13) will no longer hold.

To see what is involved with the previous expression, let's try to find the derivative at some location on the base grid, $x_{0,l}$:

$$\frac{df(x_{0,l})}{dx} = \sum_{k=0}^{N_0} f_{0,k} \frac{d\phi_{0,k}(x_{0,l})}{dx} + \sum_{j=0}^{J-1} \sum_{k=0}^{N_j - 1} d_{j,k} \frac{d\psi_{j,k}(x_{0,l})}{dx}$$

Unlike before, there is very little we can do with this expression. In the first set of terms, when l near k we will have contributions because the basis function is compactly supported, and so is it's derivatives. These terms would all work out to be taking the derivative of the basis functions at the zero level points. This would be a small set of values that we would have to compute in some way. However, we have less luck with the second terms. We cannot use (13) to get all zeros here. Instead, we will pick up a contribution from all of the higher levels where $x_{0,l}$ overlaps the support of $d\psi_{j,k}/dx$. This could be a huge number of points, spanning all the way to the finest grid. There are a large number of terms, but each term in this case would use the same set of relative locations as those in the first terms.

More specifically, we can imagine mapping each basis function onto the special function $\phi(x)$ (note the lack of index). So really the derivative values we would need all correspond to a small number of locations on $d\phi(x)/dx$. So while the sum would be large, there would be few actual values of the derivative we would need.

This changes when we go to a point at a higher level, $x_{1,l}$:

$$\frac{df(x_{1,l})}{dx} = \sum_{k=0}^{N_0} f_{0,k} \frac{d\phi_{0,k}(x_{1,l})}{dx} + \sum_{j=0}^{J-1} \sum_{k=0}^{N_j-1} d_{j,k} \frac{d\psi_{j,k}(x_{1,l})}{dx}$$

Starting with the first terms, we see now that we also need to evaluate $d\phi(x)/dx$ at locations between the locations we needed in the previous example. The locations from the previous example suffice for the second terms for this point at level 1.

When we go to a point at level two, we have to add more points to our evaluation of $d\phi/dx$, the ones in between the set of points we have computed for in the previous two examples. In the second set of terms, we now have two classes of contributions. For those level above 1, we can use still those same locations in $d\phi/dx$ from the first example. For the level 1 contributions, we need to use the new points from the second exmple.

In general, we shall need a large number of locations of $d\phi/dx$. However, we can make use of the fact that the basis at one level can be written in terms of the basis at the next highest level (and actually any higher level if we iterate the procedure). If we take the derivative (14) we find:

$$\frac{d\phi_{j,k}}{dx} = \sum_{l=0}^{N_{j+1}} h_{j,k}^{j+1,l} \frac{d\phi_{j+1,l}}{dx} \tag{30}$$

These can all be easily mapped into derivative of $\phi(x)$. What is more important is that if we have the derivatives at the level $j+1$ locations for the level $j+1$ function, we can get the derivatives at the level $j+1$ locations for the level j function. And here is where we make some headway. If we can compute the derivatives for the level j function at the level j points (there will only be a few that are nonzero), we can use the scaling relationship to get the derivatives at the level $j+1$ points of the level $j+1$ function. Then we can use (30) to get the derivatives at the level $j+1$ points of the level j function. In this manner, we can fill out all the values of the derivatives that we shall need.

One comment is worth making. The number of terms that we have to compute in (29) will in general be many. And in general to get the derivative values at the relevant locations there will be more work in nested summations for each term of (29).

V. INTEGRATION OF ITERATED INTERPOLATING WAVELETS

A. Notations

We shall refer to the single function from which the scaled dilated wavelet basis is formed by the symbol ϕ. The relation to the scaled dilated form at level j at position $x_{j,k} = 2^{-j}k\Delta x$ (where Δx is the spacing at the base level of the grid and k labels points):

$$\phi_{j,k} = \phi\left(\frac{2^j}{\Delta x}x - k\right) \tag{31}$$

We will derive some relationships about ϕ which can easily be extended to any element of a basis formed of these scaled dilated functions.

B. Zeroth Moment

We begin with the total integral of the function over the full grid. For this section, we will imagine the grid extends infinitely in each direction. Then, in a later section, we will go over what changes when we move to a finite grid. In particular, there are some changes in the points near the boundary of the domain

First, we define an approximation to the area under ϕ:

$$M_j \equiv \sum_k \Delta_j \phi(x_{j,k}) \tag{32}$$

where $\Delta_j = \Delta x/2^j$ is the spacing of the grid at level j. Given the definition of our basis function, $M_0 = \Delta x$. Given the definition of M_j, the following is true:

$$\lim_{j\to\infty} M_j = \int_{-\infty}^{\infty} \phi(x)dx \tag{33}$$

We will make use of the definition of the function ϕ to relate M_j to M_{j+1} and we shall see that we can find this limit.

We repeat the definition for M_{j+1}:

$$M_{j+1} = \sum_k \Delta_{j+1}\phi(x_{j+1,k}) \tag{34}$$

The nested wavelet grid has the property that points existing on level j also exist on level $j+1$ with even index. So we can split the sum into two parts, one for repeat points, and one for the points that are new at level $j+1$:

$$M_{j+1} = \sum_k \Delta_{j+1}\phi(x_{j+1,2k}) + \sum_k \Delta_{j+1}\phi(x_{j+1,2k+1}) \tag{35}$$

We can move the width out of the summation, and relate it to the width at the previous level:

$$M_{j+1} = \frac{1}{2}\Delta_j \sum_k \phi(x_{j+1,2k}) + \frac{1}{2}\Delta_j \sum_k \phi(x_{j+1,2k+1}) \tag{36}$$