

Study of Multiresolution Concepts for Numerically Solving PDEs

Brandon Gusto

May 8, 2018

Motivation for Multiresolution Methods

Computer simulation of physical phenomenon has become a centerpiece of modern engineering design. As more accuracy is demanded, and more complex physics need to be resolved, the demand on computing power increases dramatically. In recent years, processing power for a single chip has stopped making significant year-over-year gains. In light of this, computational scientists around the world are coming up with new ways to handle the tremendous amount of computations necessitated by numerical simulations. Computer simulations are designed to numerically solve the (usually simplified) system of partial differential equations which govern the behavior of the physics of interest. Real-world simulations typically run with a computational grid that contains tens or hundreds of millions of unknowns. Since the 1980s, computational scientists have devised methods which try to reduce the number of degrees of freedom (unknowns) in a system. Adaptive mesh refinement (AMR) [berger] was the first method of adaptively refining the solution where more accuracy is needed. AMR is still the standard in most production and research codes today. AMR typically refines the solution by running the simulation on a coarse and fine grid simultaneously, and estimating via Richardson extrapolation the truncation error.

An alternative to AMR methods for providing a reliable measure of error is multiresolution analysis (MRA). MRA provides a hierarchy of nested approximation spaces, with a coarsest and finest approximation. Errors between levels of resolution are stored in the detail wavelet spaces. In the last few decades, multiresolution methods have gained considerable attention and have been the focus of research in a variety of fields, most notably signal processing and computer graphics. Numerically solving differential equations is a more recent application of multiresolution properties, and has shown to be an attractive alternative to traditional methods for particular problems. Some methods simply use the compression properties of wavelets to provide spatial adaptivity to a numerical method, while other methods such as wavelet galerkin or wavelet collocation use the wavelet basis explicitly to reconstruct a field. Some important qualities of the method which make it so capable for spatial adaptivity are that

- wavelets are localized in space and scale
- the wavelet basis forms a multiresolution analysis
- the existence of a fast discrete wavelet transform makes the method $\mathcal{O}(\mathcal{N})$

One of the most attractive qualities of using wavelets for solving partial differential equations (PDEs) or systems of PDEs is for their ability to naturally compress the solution, providing high resolution where it is needed, and fewer grid points where the solution is smooth. In a study between the compression properties and time-to-solution [**amr**'**vs**'**mra**], multiresolution methods showed a greater ability for compression of solution when large disparity was present between length scales in the problem.

Brief Review of Wavelets

A wavelet is a mathematical function which can be used to generate a basis for representing either continuous functions or discrete signals. There exist many other bases which have been utilized, most notably the Fourier basis composed of sine and cosine functions. Unlike the Fourier basis functions however, wavelets are functions with finite length. The most simple example of a wavelet is the Haar wavelet, which has a scaling function, $\phi(x)$, and a mother wavelet $\psi(x)$ as shown in figure (1) and (2).

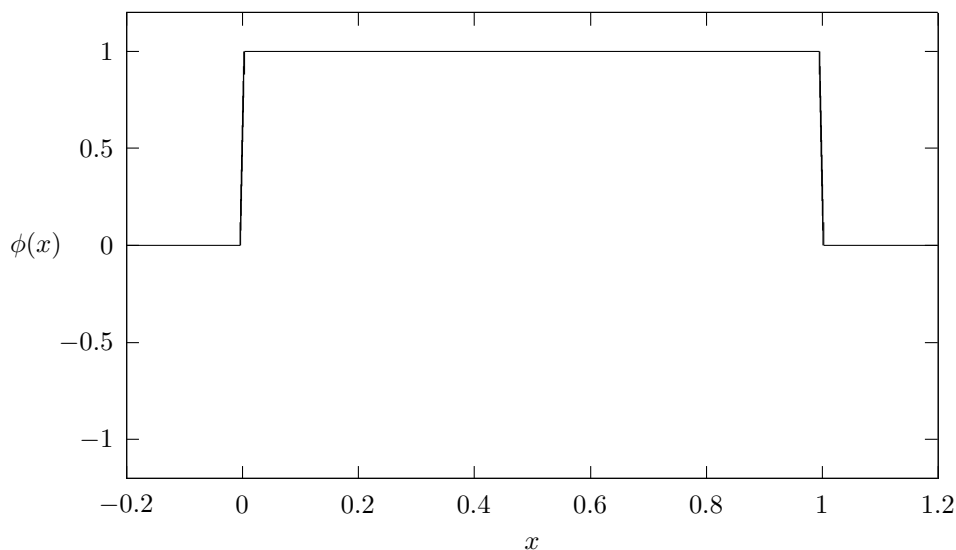


Figure 1: Haar scaling function

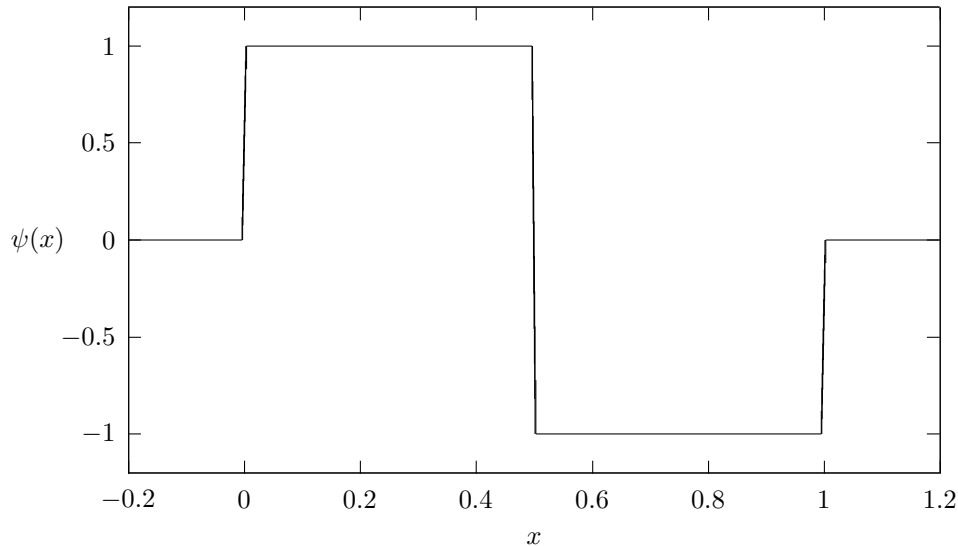


Figure 2: Haar wavelet function

The scaling and wavelet functions have compact support, thus to approximate functions it is necessary to shift them. This action is *translation*. Furthermore, since functions may exhibit sharp changes over an interval which is smaller than the piecewise constant wavelet, a wavelet of higher frequency is necessary to fit the function. A *dilation* of the original wavelet decreases its support, allowing it to approximate more complex functions. These basic operations allow for the formation of a basis. The basis is constructed from the scaling and wavelet functions as

$$\phi_{j,k}(x) = 2^{j/2} \phi(2^j x - k) \quad (1)$$

$$\psi_{j,k}(x) = 2^{j/2} \psi(2^j x - k). \quad (2)$$

Only in the limit as j tends to infinity does this basis fully represent a continuous function. The three main qualities of wavelets which make them such a useful tool for approximation are **compact support**, **orthogonality**, and **multiresolution analysis**.

Compact Support

Wavelets are uniformly zero outside of a specified interval, and this useful property is known as compact support. This property is advantageous for approximating sharp signals, where one would like to represent the spike using only a small handful of basis functions. The Fourier basis is not adequate for this

purpose, since the support for the basis is global; the sine and cosine functions have infinite length.

Orthogonality

The basis of wavelet functions constitute an orthonormal set. All translations and dilations of the wavelet $\psi(x)$ are orthogonal to each other,

$$\int_{-\infty}^{\infty} \psi_{j,k}(x) \psi_{j',k'}(x) dx = 0, \quad (3)$$

if $j \neq j'$ or $k \neq k'$. Furthermore, all combinations of translates and dilates of the wavelet are orthogonal to all translates and dilates of the scaling function,

$$\int_{-\infty}^{\infty} \phi_{j,k}(x) \psi_{j',k'}(x) dx = 0, \quad (4)$$

for all integers j, j', k , and k' .

Multiresolution Analysis

The nestedness of the subspaces spanned by the different wavelet scales is a most useful property, allowing functions to be approximated using as few basis functions as possible. Let V_0 be the space which is spanned by the basis of scaling functions $\phi_{0,k}(x)$ at the coarsest level of resolution. The space spanned by the wavelets at the same level $j = 0$ is W_0 . Due to the aforementioned orthogonality between scaling functions and wavelets, the space W_0 is orthogonal to V_0 . The refined space V_1 can be described by

$$V_1 = V_0 \oplus W_0, \quad (5)$$

which represents all piecewise functions defined on half-intervals. Equation (5) indicates that the space spanned by $\phi(2x)$ can be described by the ‘summation’ of the spaces spanned by V_0 and W_0 . One finer level, V_2 , is spanned by translates of $\phi(4x)$. It can be written as

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1. \quad (6)$$

Considering successively finer approximation spaces yields the spaces

$$V_j = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_j. \quad (7)$$

Thus the spaces V_j are nested, since the function $\phi(x)$ is a combination of $\phi(2x)$ and $\phi(2x - 1)$, and so on. In mathematical notation, $V_0 \subset V_1 \subset \cdots \subset V_j$.

Second-Generation Wavelet Methods

The wavelets previously described are only ever translates and dilates of a single mother wavelet function. This family of wavelets is classified as first-generation, and they are tied to the use of regular grids. The lifting scheme developed by [sweldens] abandoned the restriction to regular grids and the concept of translates and dilates. These so-called second-generation wavelets are adaptive wavelets, allowing wavelets to be built with points ‘missing’ from the grid (an irregular grid).

Adaptive Wavelet Collocation Method

The adaptive wavelet collocation method [vasilyev] has been successfully applied to parabolic, elliptic, and hyperbolic PDEs in applications such as aeroacoustics, turbulence, flame interactions, and others. The algorithm makes use of adaptive, second-generation wavelets to compress the solution.

Dyadic Grid

The multi-resolution properties of wavelets previously described warrant a multilevel, or dyadic grid. This grid should have as many levels as there are scales in the given problem. It can consist of either uniformly or non-uniformly spaced points. In the case of equally spaced grid points, let each grid level $j = 0, \dots, J$ be computed by

$$x_k^j = 2^{-(j+\delta)} k, \quad \text{for } k = 0, \dots, 2^{j+\delta}, \quad (8)$$

where δ is some integer shifting parameter, allowing one to dictate that the coarsest level of resolution have smaller spacing than 1 as in the case where $\delta = 0$. From here forth we keep omit the parameter δ , but keep in mind that the number of grid points at each level may be shifted at will based on a fixed choice of δ . The grid levels are formally defined by

$$\mathcal{G}^j = \{x_k^j \in \Omega : k \in \mathcal{K}^j\}, \quad j \in \mathcal{Z}, \quad (9)$$

where \mathcal{K}^j is the integer set representing the spatial locations in the grid at level j . The grids are nested, implying that $\mathcal{G}^j \subset \mathcal{G}^{j+1}$. In other words, the points x_k^j are a perfect subset of the points x_k^{j+1} . This can be demonstrated by the relation that $x_k^j = x_{2k}^{j+1}$ for $k \in \mathcal{K}^j$.

Interpolating Subdivision Algorithm

The interpolating subdivision scheme is central to the second-generation wavelet collocation approach. The scheme is used to approximate values at odd points x_{2k+1}^{j+1} by using $2N$ nearest points to construct interpolating polynomials of order $2N - 1$. Lagrange polynomials are used, and the method can be used

with a uniform grid or with nonuniform points such as Chebyshev points. The interpolating scheme is

$$f(x_{2k+1}^{j+1}) = \sum_{l=-N+1}^N L_{k+l} f(x_{k+l}^j), \quad (10)$$

where the interpolating weights are Lagrange polynomial coefficients given by

$$L_{k+l}(x) = \prod_{\substack{i=k-N+1 \\ i \neq k+l}}^{k+N} \frac{x - x_i}{x_{k+l} - x_i}. \quad (11)$$

Given a uniform grid spacing at level j of h^j , the accuracy of such an interpolation is of $\mathcal{O}((h^j)^{2N})$.

Wavelet Transform

The existence of a fast wavelet transform is one of the attractive qualities of the method. The transform makes use of the interpolating subdivision algorithm (10) for the calculation of the scaling and detail wavelet coefficients. The forward wavelet transform is given by

$$d_k^j = \frac{1}{2} \left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1} \right), \quad (12)$$

$$c_k^j = c_{2k}^{j+1},$$

and the inverse transform is given by

$$c_{2k+1}^{j+1} = 2d_k^j + \sum_l w_{k,l}^j c_{k+l}^j, \quad (13)$$

$$c_{2k}^{j+1} = c_k^j.$$

Scaling Functions

The construction of second-generation interpolating wavelets makes use of the interpolating subdivision algorithm. The scheme is used to interpolate functional values defined at points on level j , to odd points (i.e. x_{2k+1}^{j+1}) at the next higher level of resolution. This scheme is used to construct the scaling and detail wavelet functions. Examples of the scaling and detail functions are shown in Figure 1 and Figure 2 respectively. To obtain the scaling function $\phi_m^j(x)$, from (5) set $c_k^j = \delta_{k,m}$, $\forall k \in \mathcal{K}^j$, where $\delta_{k,m}$ is the Kronecker delta function defined by

$$\delta_{k,m} = \begin{cases} 1 & k = m \\ 0 & k \neq m. \end{cases}$$

Then let all $d_l^{j'} = 0, \forall l \in \mathcal{L}^{j'}, \forall j' \geq j$ and perform the inverse transform up to an arbitrarily high level of resolution J .

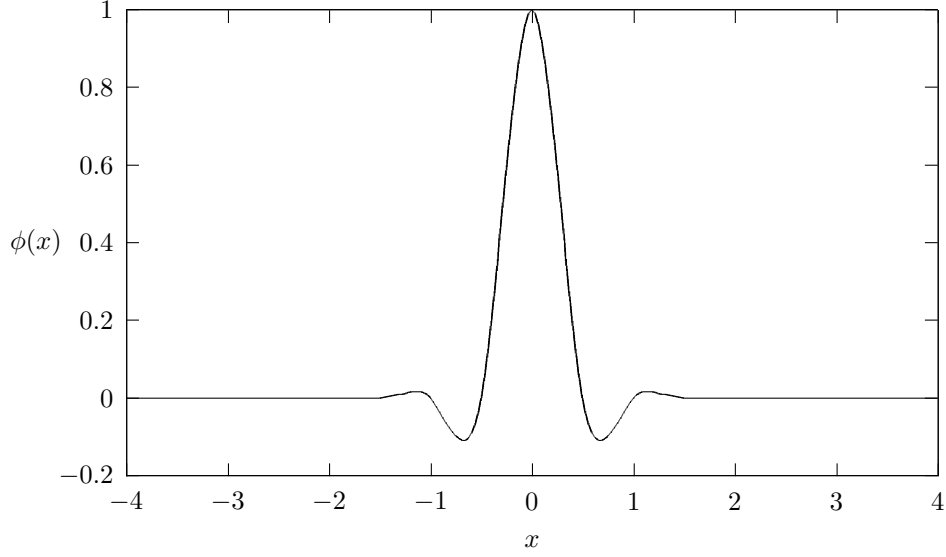


Figure 3: An example of a scaling function, $\phi(x)$, for $N = 3$.

Wavelet Functions

The wavelet ψ_l^j is computed by setting $d_m^{j'} = \delta_{j',j} \delta_{l,m}, \forall l \in \mathcal{L}^j, \forall j \geq j$, and also $c_k^j, \forall k \in \mathcal{K}^j$. Then perform the inverse wavelet transform up to an arbitrarily high level of resolution J .

Approximation of Functions

The approximation of a function $f(x)$ is done by setting the scaling coefficients at the arbitrary maximum level of resolution J to the function itself. Once the function is sampled this way for all c_k^J , the forward wavelet transform is performed down to the coarsest level of resolution. The function is then represented by

$$f^J(x) = \sum_{k \in \mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{l \in \mathcal{L}^j} d_l^j \psi_l^j(x). \quad (14)$$

Often, a large number of wavelet coefficients can be discarded, and the approximation (6) is still adequate. Define some threshold ϵ for the coefficients, then keep only those coefficients which satisfy $|d_l^j| \geq \epsilon$. The approximation (6) becomes

$$f_{\geq}^J(x) = \sum_{k \in \mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{\substack{l \in \mathcal{L}^j \\ |d_l^j| \geq \epsilon}} d_l^j \psi_l^j(x). \quad (15)$$

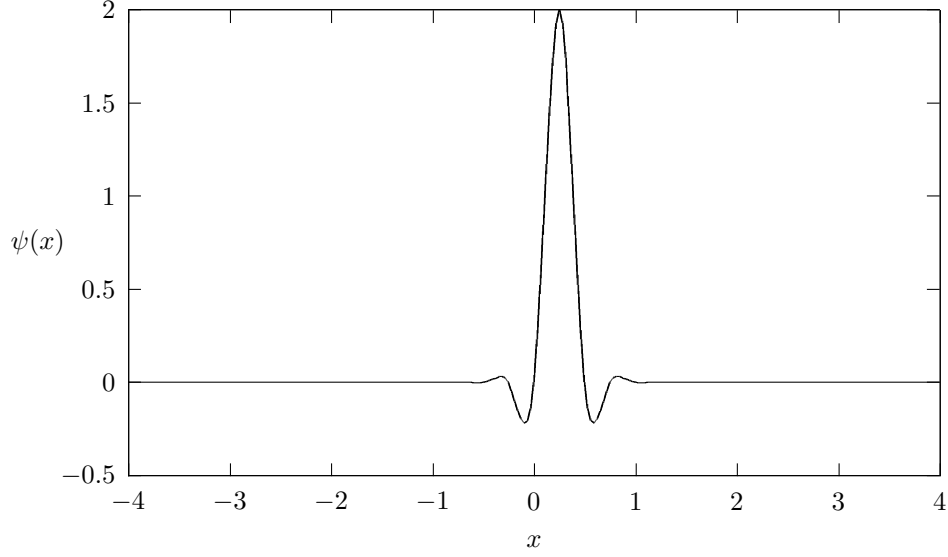


Figure 4: An example of a wavelet, $\psi(x)$, for $N = 3$.

In the figures below are approximations to the function $f(x) = \cos(80\pi x) \exp(-64x^2)$ with an increasing threshold ϵ .

Calculation of Spatial Derivatives

The calculation of spatial derivatives involves analyzing the detail coefficients, as they are a measure of how well the given function is approximated by the local interpolant. If a point x_k^j on level j of the dyadic grid, does not have detail coefficients above the threshold at points x_{2k+1}^{j+1} or x_{2k-1}^{j+1} , then the given function's error is bounded by the thresholding parameter ϵ . Thus a stencil consisting of the same points which constructed the underlying polynomial should result in an accurate result at that point, within some constant multiple of the tolerance ϵ .

The computational complexity of constructing Lagrange interpolating polynomial coefficients is $\mathcal{O}(N)$. Once these terms are known, the weights needed for computing spatial derivatives on the adaptive grid can be computed with only slightly more effort. This is also an $\mathcal{O}(N)$ operation. The analytic formula for the coefficients is given by

$$\frac{d}{dx} L_{k+l}(x) = L_{k+l}(x) \sum_{\substack{i=k-N+1 \\ i \neq k+l}}^{k+N} \frac{1}{x_{k+l} - x_i}. \quad (16)$$

However once this coefficient is calculated for each point in the stencil, it must

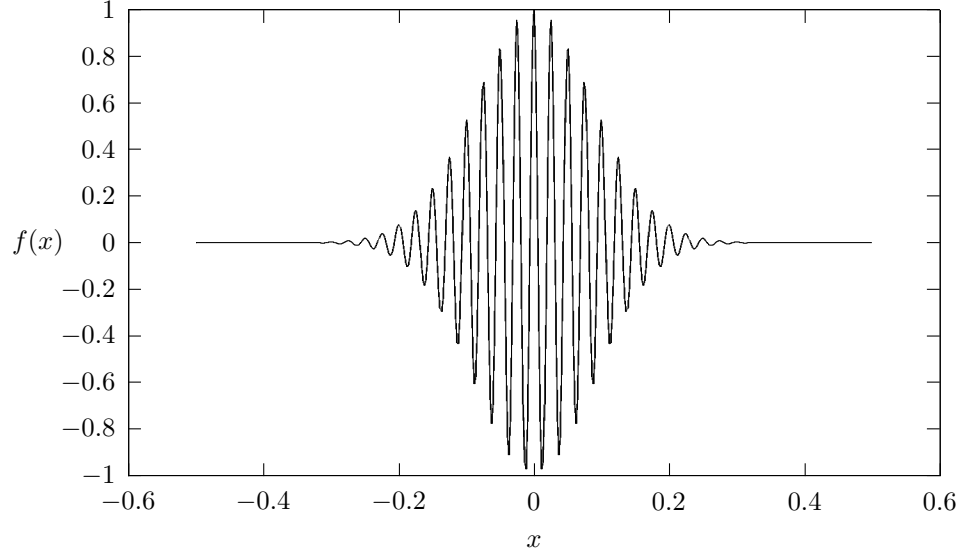


Figure 5: Approximation of $f(x) = \cos(80\pi x) \exp(-64x^2)$, with $N = 3$ and 256 points on \mathcal{G}^J . All coefficients are kept.

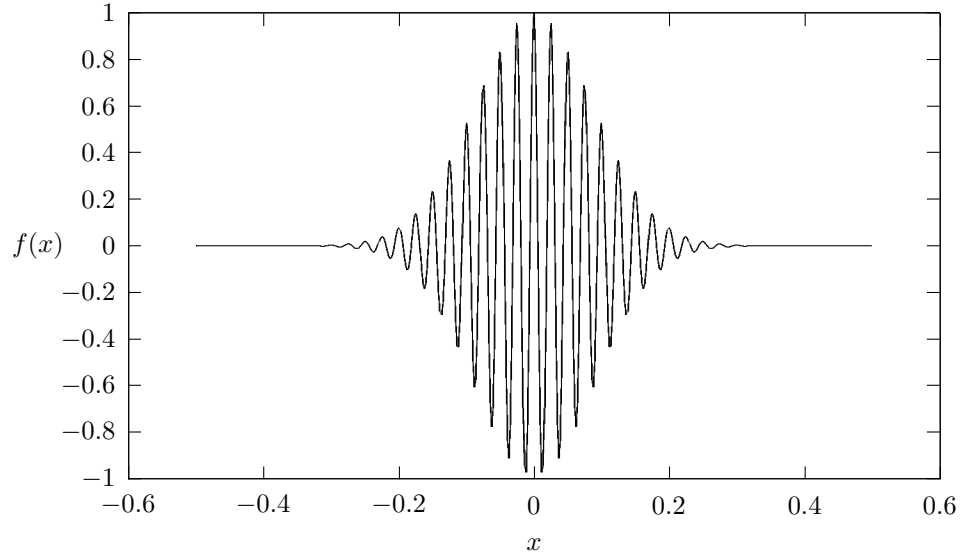


Figure 6: Same function with a tolerance of $\epsilon = 10^{-5}$.

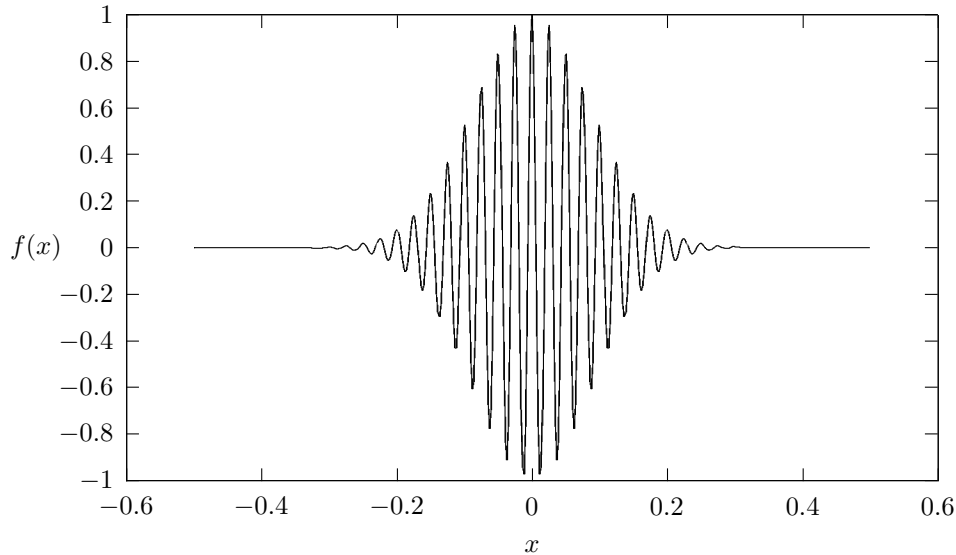


Figure 7: Again with a coarse tolerance of $\epsilon = 10^{-3}$.

be multiplied by its functional value, and all points are then summed again in the same way that the Lagrange polynomial is in the first place. Thus the total complexity of computing derivatives this way is $\mathcal{O}(N^2)$, which is not desirable. The order of accuracy for computing one spatial derivative this way is $\mathcal{O}((h^j)^{2N-1})$, where h^j is the local grid spacing. Note that for a stencil consisting of four points ($N = 2$), this method is not only computationally inefficient, but also 1 order of accuracy worse than a comparable four-point centered finite difference scheme, which has $\mathcal{O}((h^j)^{2N})$ accuracy. The only advantage of this method can be seen when the stencil is non-uniform, in which case the finite difference coefficients would have to be computed either in an implicit sense, or by interpolation, as done here.

An alternative and perhaps more efficient method in some cases, is to pre-compute every possible finite-difference stencil on the grid and store the coefficients in a lookup table [**rastigejev**].

Time-Stepping

Advancement in time of the adaptive solution is done as in any method of lines procedure. The forcing term for the differential equation (in time) is evaluated via appropriate finite differencing as previously explained. This procedure is repeated for as many times as there are stages in the time integration method (4 for RK4). At each iteration, to maintain stability of the compressed solution, a buffer zone of points is added. Typically the buffer zone ensures that

each point that originally met the threshold is surrounded by two points of the same resolution level. If these points are missing, it is possible that during time integration, the points could become relevant (physics moving into that area) but that algorithm would have no way of knowing it.

Adaptive Wavelet Collocation Code

A program was successfully developed in C++ to solve numerous one-dimensional partial differential equations with the AWCM. The code repository can be found at <https://github.com/blg13/Wavelet/tree/CollocationPoint>

Multiresolution Finite-Volume

Most multiresolution methods for PDEs do not explicitly compute wavelets, but rather use prediction operators and the multiresolution framework for compression of the solution. This method was pioneered by Harten [**Harten**]. The finite volume method works with cell averages, $u_{j,k}$, where j denotes the resolution level and k denotes the spatial index. The general idea is to represent the cell-averaged data on a fine grid as values on a coarse grid plus the differences at subsequent grid levels. The idea is identical in nature to the restriction and prolongation actions in the multigrid framework.

Structured & Regular Rectilinear Grid

As previously mentioned, the multiresolution analysis is tied to the structure of the grid. For regular, rectilinear grids the prediction operators are fixed and can be easily derived in more than one dimension. In one dimension the update (fine to coarse) operator is given by

$$\bar{u}_{j,k} = \frac{1}{2}(\bar{u}_{j+1,2k} + \bar{u}_{j+1,2k+1}) \quad (17)$$

and the prediction operator (coarse to fine) is given as

$$\hat{u}_{j+1,2k} = \bar{u}_{j,k} + \sum_{i=1}^m \gamma_k(\bar{u}_{j,k+i} + \bar{u}_{j,k-i}) \quad (18)$$

$$\hat{u}_{j+1,2k+1} = \bar{u}_{j,k} - \sum_{i=1}^m \gamma_k(\bar{u}_{j,k+i} + \bar{u}_{j,k-i}), \quad (19)$$

where if fifth-order interpolation is used ($m = 2$), then the coefficients are $\gamma_1 = -\frac{22}{128}$ and $\gamma_2 = \frac{3}{128}$. The detail coefficients computed at every resolution level are given by

$$\bar{d}_{j,k} = \bar{u}_{j,k} - \hat{u}_{j,k}. \quad (20)$$

If the prediction operator is stable, then the detail coefficients should be an indicator of regularity, or smoothness of the underlying function. In this way, spatial adaptivity can be obtained by only accepting cells with detail coefficients of significant value.

Irregular Connectivity Meshes

The need to compress large quantities of three-dimensional image data drove the graphics community over the last three decades to develop many methods of subdivision for creating level-of-detail models. These meshes have no regular structure, and thus first-generation multiresolution analysis is not possible.