Implementing the Adaptive Wavelet Collocation Method for Analyzing Calcium Dynamics in Astrocyte Branch Structures

Brandon Gusto

November 13, 2017

1 Motivation for the Adaptive Wavelet Collocation Method

The adaptive wavelet collocation method (AWCM) is a method for numerically solving differential equations. It has a number of merits which make it an attractive alternative to traditional finite element, finite volume, and finite difference methods for particular problems. The AWCM has been successfully applied to parabolic, elliptic, and hyperbolic PDEs, in applications such as aeroacoustics, turbulence, flame interactions, and others. Some important qualities of the method which make it so capable are that

- wavelets are localized in space and scale
- the wavelet basis forms a multiresolution analysis

Probably the most attractive quality of wavelets for solving partial differential equations (PDEs) or systems of PDEs is for their tremendous ability to compress the solution, providing high resolution where it is needed, and fewer grid points where the solution is smooth. The wavelet basis is ideal for problems which have length scales ranging from the large structures, down to the Kolmogorov scale which governs the size of the smallest eddies in fluid turbulence; and time scales ranging from microseconds for turbulence down to nanoseconds for combustion or the propagation of a flame front.

2 Fundamentals of Wavelets

A wavelet is a mathematical function which can be used as 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).

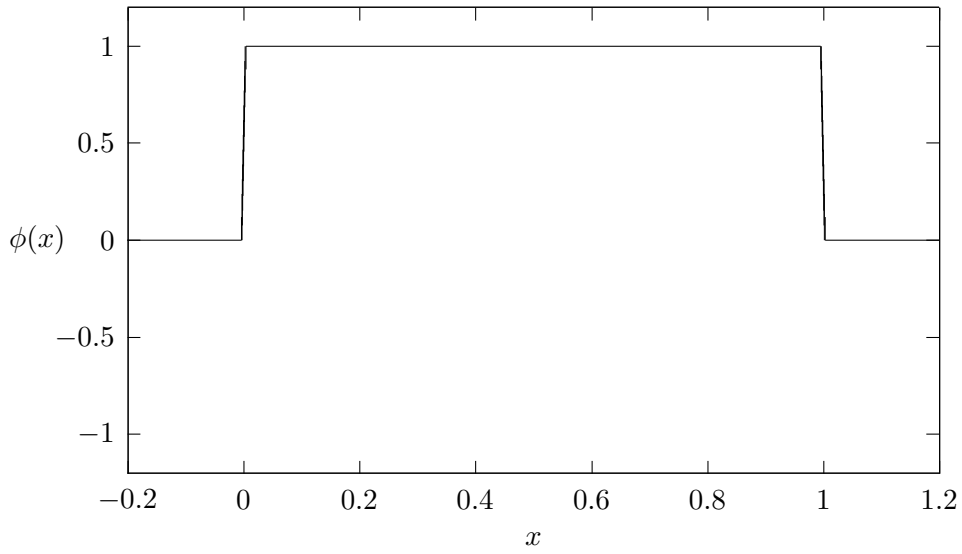

Figure 1: Haar scaling function

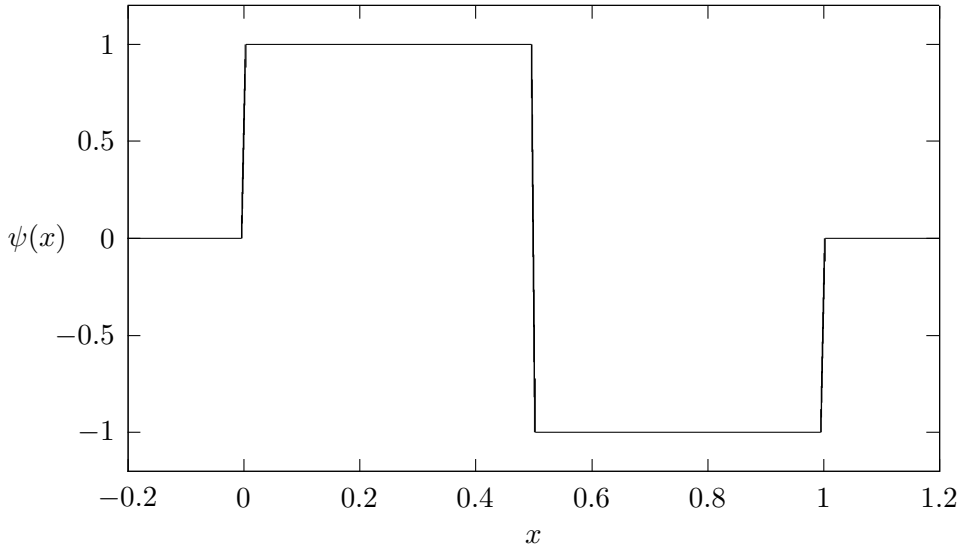

Figure 2: Haar wavelet function

Since the wavelet functions are localized in space, it is necessary to shift them in order to approximate a function. This action is referred to as *translation*. Furthermore, functions may exhibit very sharp changes over an interval which is smaller than the piecewise constant wavelet, and so a wavelet of higher frequency would be necessary to fit the function. By using a *dilation* of the original wavelet, the function can be well approximated. Define a relationship between the

scaling function $\phi(x)$ and the translates of its dilation $\phi(2x)$ by the dilation equation given by

$$\phi(x) = \sum_{k=0}^{N} c_k \phi(2x - k). \tag{1}$$

The three pillars(?) of wavelets which make them such a useful tool for approximation are **Orthogonality**, **Compact Support**, and **Multiresolution Analysis**.

2.1 Orthogonality

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

$$\int_{-\infty}^{\infty} \phi(x)\psi(x)dx = 0, \tag{2}$$

and other wavelets

$$\int_{-\infty}^{\infty} \psi_k^j(x)\psi_{k'}^{j'}(x)dx = 0. \tag{3}$$

2.2 Compact Support

Wavelet functions have compact support, meaning that they are uniformly zero outside of a specified interval. 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. While successively higher-frequency sine and cosine terms can be superimposed, the result is the necessary storage of more basis functions which cancel them out away from the sharp signal.

2.3 Multiresolution Analysis

Let us define the space which is spanned by the basis of scaling functions $\phi_k^0(x)$ at the coarsest level of resolution by V_0. We understand from the previous section that the set of wavelet functions are orthogonal to the set of scaling functions, and so the space spanned by the basis of wavelet functions, denoted by W_0, is thusly orthogonal to V_0. The refined space V_1 can be described by

$$V_1 = V_0 \oplus W_0, \tag{4}$$

which represents all piecewise functions defined on half-intervals. Equation (4) 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

$$V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1. \tag{5}$$

Considering successively finer approximation spaces spanned by $\phi(4x), \phi(8x), \ldots, \phi(2^j x)$, yields the spaces

$$V_j = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_j. \tag{6}$$

The spaces V_j are nested, since for example the function $\phi(x)$ is a combination of $\phi(2x)$ and $\phi(2x-1)$. In other terms, $V_0 \subset V_1 \subset \cdots \subset V_j$.

2.4 Dyadic Grid

In developing an adaptive wavelet collocation algorithm, an appropriate numerical grid should be described. The multi-resolution properties of wavelets previously described warrant a multilevel grid. This is called a dyadic grid, and it 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 $j = 0, \ldots, J$ be computed by

$$x_k^j = 2^{-(j+\delta)}k, \quad \text{for } k = 0, \ldots, 2^{j+\delta}\ , \tag{7}$$

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

$$\mathcal{G}^j = \{x_k^j \in \Omega : k \in \mathcal{K}^j\},\ j \in \mathcal{Z}, \tag{8}$$

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^j are a perfect subset of the points x^{j+1}. This can be demonstrated by the relation that $x_k^j = x_{2k}^{j+1}$ for $k \in \mathcal{K}^j$.

3 Second-Generation Wavelets

Second-generation wavelets refer to the family of functions which are no longer defined as translates and dilations of a single mother function. The necessity of such functions

3.1 Interpolating Subdivision

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

$$f(x_{2k+1}^{j+1}) = \sum_{l=-N+1}^{N} L_{k+l} f(x_{k+l}^j), \tag{9}$$

where the interpolating weights are Lagrange polynomial coefficients given by

$$L_{k+l}(x) = \prod_{\substack{i=k-N+1 \\ i \neq k+l}}^{k+N} \frac{x - x_i}{x_{k+l} - x_i}. \tag{10}$$

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

3.2 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 (9) for the calculation of the scaling and detail wavelet coefficients. The forward wavelet transform is given by

$$\begin{aligned} d_k^j &= \frac{1}{2}\left(c_{2k+1}^{j+1} - \sum_l w_{k,l}^j c_{2k+2l}^{j+1}\right), \\ c_k^j &= c_{2k}^{j+1}, \end{aligned} \tag{11}$$

and the inverse transform is given by

$$\begin{aligned} c_{2k+1}^{j+1} &= 2d_k^j + \sum_l w_{k,l}^j c_{k+l}^j, \\ c_{2k}^{j+1} &= c_k^j. \end{aligned} \tag{12}$$

3.3 Construction of Interpolating Scaling and Wavelet Functions

The wavelet and scaling function do not have a closed-form expression.

3.3.1 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.

3.3.2 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.

4 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 on $\mathcal{G}^J$,

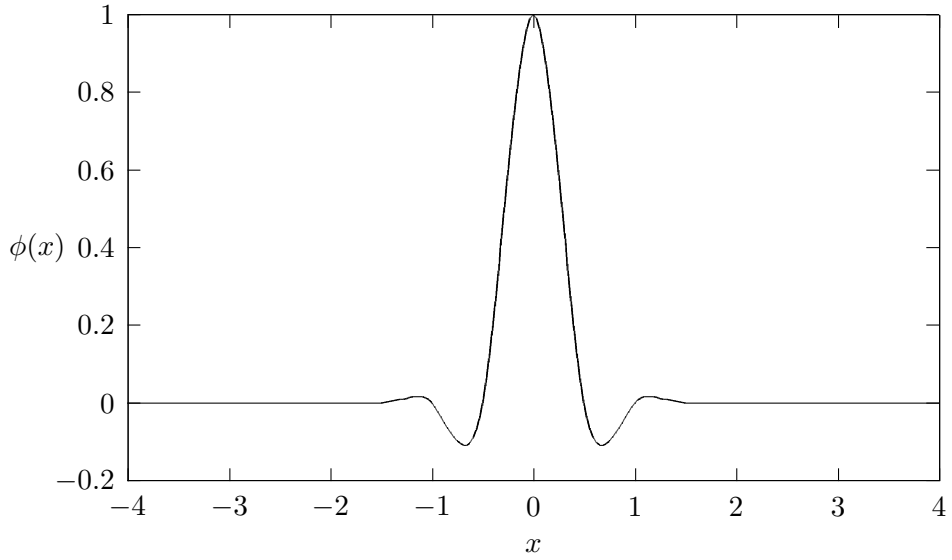

Figure 3: An example of a scaling function, $\phi(x)$, for $N = 3$.

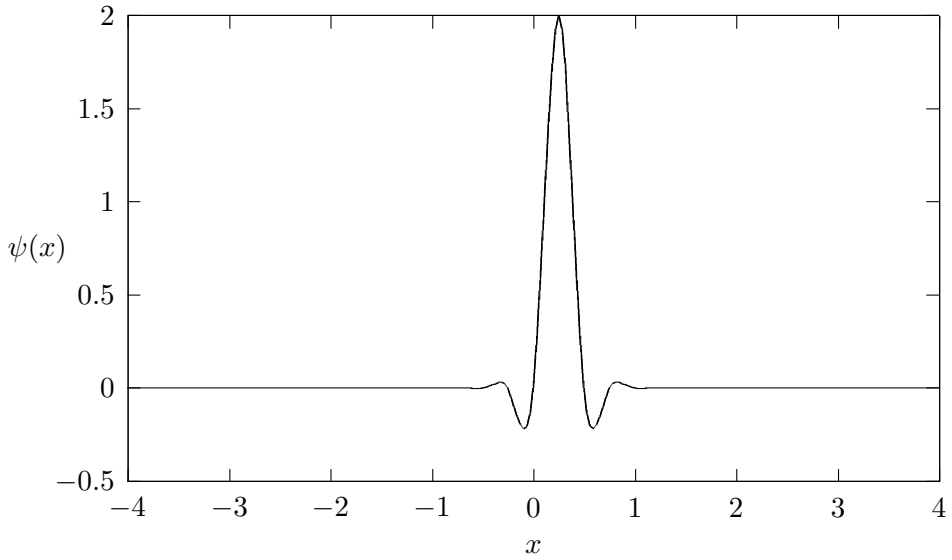

Figure 4: An example of a wavelet, $\psi(x)$, for $N = 3$.

the forward wavelet transform is performed down to the coarsest level of resolution. The function is then represented by

$$f^J(x) = \sum_{k \in \mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{l \in \mathcal{L}^j} d_l^j \psi_l^j(x). \tag{13}$$

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

$$f_{\geq}^J(x) = \sum_{k \in \mathcal{K}^0} c_k^0 \phi_k^0(x) + \sum_{j=0}^{J-1} \sum_{\substack{l \in \mathcal{L}^j \\ |d_l^j| \geq \epsilon}} d_l^j \psi_l^j(x). \tag{14}$$

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

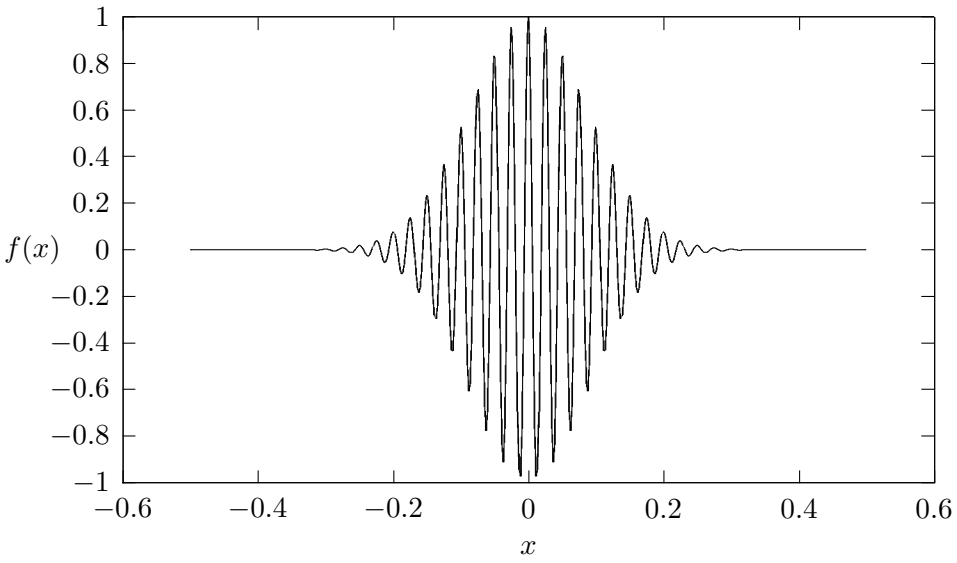

Figure 5: Approximation of $f(x) = \cos(80\pi x)\exp(-64x^2)$, with $N = 3$ and 256 points on $\mathcal{G}^J$. All coefficients are kept.

5 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.

5.1 Differentiating Lagrange Polynomials

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

$$\frac{d}{dx}L_{k+l}(x) = L_{k+l}(x) \sum_{\substack{i=k-N+1 \\ i\neq k+l}}^{k+N} \frac{1}{x_{k+l} - x_i}. \tag{15}$$

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

5.2 Non-Uniform Finite Difference Stencils

6 Single-Precision Finite Differencing with Richardson Extrapolation

When evaluating finite differences with finite-precision machines. one must consider not just the truncation error of the scheme, but also the induced roundoff error. To illustrate, consider the 2-point centered finite difference scheme,

$$\overline{f}'(x) \approx \frac{f(x+h) - f(x-h)}{2h}. \tag{16}$$

The derivation of the scheme and its truncation error is done using the Taylor series expansions about the points $x + h$ and $x - h$. This yields

$$f(x+h) = f(x) + f'(x)h + \frac{f^{(2)}(x)h^2}{2!} + \frac{f^{(3)}(x)h^3}{3!} + \mathcal{O}(h^4), \tag{17}$$

and

$$f(x-h) = f(x) - f'(x)h + \frac{f^{(2)}(x)h^2}{2!} - \frac{f^{(3)}(x)h^3}{3!} + \mathcal{O}(h^4). \tag{18}$$

Subtracting these expansions and rearranging yields

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{f^{(3)}(x)h^2}{3!}, \tag{19}$$

where the final term represents the truncation error of the scheme. But in reality, computations are carried out on finite-precision computers, with the amount of precision given by ϵ_{mach}. So the

total error must include a roundoff term. The roundoff error is given by the difference between the computer's representation of a number, $\overline{f}(x)$ with the actual number, $f(x)$. Let

$$\overline{f}(x+h) - \overline{f}(x-h) = f(x+h) - f(x-h) + C\epsilon_{\text{mach}} f(x), \tag{20}$$

where C is some constant of order 1. Then the roundoff error in the finite difference approximation is

$$\begin{aligned} \epsilon_{\text{roundoff}} &= \frac{\overline{f}(x+h) - \overline{f}(x-h)}{2h} - \frac{f(x+h) - f(x-h)}{2h} \\ &\approx \frac{C\epsilon_{\text{mach}}}{2h}. \end{aligned}$$

Combining the truncation error and the roundoff error in the approximation of a derivative yields

$$f'(x) = \overline{f}'(x) + \frac{f^{(3)}(x)h^2}{3!} + \frac{C\epsilon_{\text{mach}}}{2h}. \tag{21}$$

Since the truncation term is diminishing with decreasing step-size h, while the roundoff term is increasing, (21) can be viewed as an optimization problem, with a control parameter h. Differentiating with respect to h, and setting the resulting equation to zero yields the optimal step-size

$$h^* \approx (\epsilon_{\text{mach}})^{1/3}. \tag{22}$$

As the step-size decreases from unity, the error of the approximation will decrease according to the truncation error, until h^* is reached, and then the error will increase due to roundoff error.

7 Open Areas of Research / Project Ideas

- AWCM combined with level-set method
- Wavelet Finite Element in an Arbitrary Lagrangian-Eulerian formulation.
- Single-precision implementation of AWCM for parallelization on GPU's.