Commun. Comput. Phys.
doi: 10.4208/cicp.OA-2016-0246

Vol. **22**, No. 5, pp. 1286-1308
November 2017

Mesh Density Functions Based on Local Bandwidth Applied to Moving Mesh Methods

Elliott S. Wise*, Ben T. Cox and Bradley E. Treeby

Department of Medical Physics and Biomedical Engineering, University College London, 2–10 Stephenson Way, London, NW1 2HE, United Kingdom.

Received 14 December 2016; Accepted (in revised version) 19 April 2017

Abstract. Moving mesh methods provide an efficient way of solving partial differential equations for which large, localised variations in the solution necessitate locally dense spatial meshes. In one-dimension, meshes are typically specified using the arclength mesh density function. This choice is well-justified for piecewise polynomial interpolants, but it is only justified for spectral methods when model solutions include localised steep gradients. In this paper, one-dimensional mesh density functions are presented which are based on a spatially localised measure of the bandwidth of the approximated model solution. In considering bandwidth, these mesh density functions are well-justified for spectral methods, but are not strictly tied to the error properties of any particular spatial interpolant, and are hence widely applicable. The bandwidth mesh density functions are illustrated in two ways. First, by applying them to Chebyshev polynomial approximation of two test functions, and second, through use in periodic spectral and finite-difference moving mesh methods applied to a number of model problems in acoustics. These problems include a heterogeneous advection equation, the viscous Burgers' equation, and the Korteweg-de Vries equation. Simulation results demonstrate solution convergence rates that are up to an order of magnitude faster using the bandwidth mesh density functions than uniform meshes, and around three times faster than those using the arclength mesh density function.

AMS subject classifications: 65M06, 65M50, 65M70

PACS: 02.60.Cb, 02.60.Lj, 02.70.Hm, 02.70.Jn, 43.25.+y

Key words: Moving mesh method, pseudospectral method, bandwidth.

1 Introduction

Many scientific and engineering problems require solutions to partial differential equations (PDEs). When smooth, these solutions can be efficiently computed using spectral methods. However, often solutions are not equally smooth everywhere. In particular, they might exhibit features which are tightly localised in space. These include

*Corresponding author. *Email addresses:* elliott.wise.14@ucl.ac.uk (E. S. Wise), b.cox@ucl.ac.uk (B. T. Cox), b.treeby@ucl.ac.uk (B. E. Treeby)

 ©2017 Global-Science Press

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

shock fronts, narrow pulses, and sharp corners. Such features require dense computational meshes to accurately resolve. Because spectral methods typically use standardised meshes, the global mesh density is determined by the sampling requirements of these localised features. This results in much of the spatial domain being oversampled, increasing computational expense for no accuracy gain. As an example of when this can become a critical issue, three-dimensional, full-wave simulations of nonlinear medical ultrasound fields may require many tens of gigabytes of memory to store acoustic field variables at each time-step due to large, densely sampled simulation domains [16]. These sampling requirements arise when acoustic nonlinearity causes very high frequencies to form, often within small regions where the acoustic pressure is particularly high.

Adaptive moving mesh methods can reduce the trade-off between accuracy and computational expense by providing more optimal sampling. They place mesh nodes according to a monitor function (sometimes called a mesh density function in one dimension) that is computed from (and locally dependent on) the calculated solution itself. Moving mesh methods have traditionally been implemented using finite-difference and finite-element methods, but spectral implementations offer the opportunity to improve computational efficiency further. Some examples of spectral moving mesh methods include Fourier [8,9], Galerkin [20], and Chebyshev [22] types. These all used the arclength monitor function, which clusters mesh nodes according to the gradient of the model solution. For these problems, this choice is justified by physical considerations: the model solutions in all cases feature localised steep gradients. However, it is not clear why the arclength monitor function might produce a mesh that is optimal.

One justification for the arclength monitor function is given in [15, §2.4]. Here, it is shown that derivative-based monitor functions can be derived from interpolation error bounds for piecewise polynomial interpolants. The arclength monitor function, while not strictly optimal, can be seen to be very similar to these. But this approach does not naturally extend to spectral interpolants. An alternative is to directly consider smoothness properties of the approximated solution itself. A notable one-dimensional example is found in the work of Tee et al. [11,12,23,24]. This approach is designed for solutions whose analytic continuations contain singularities. It works by first approximating the analytic continuation, after which a mesh mapping is computed that is parametrised by the locations of the approximated singularities. These mesh mappings seek to ensure that a spectral interpolant through the composition of the approximated solution and inverse mesh mapping converges on the true solution faster than a spectral interpolant through the solution alone. In [25], a mesh density function is presented which is based on Tee et al.'s approach. This work demonstrated that a singularity-based mesh density function was significantly more effective than the arclength mesh density function in reducing the trade-off between accuracy and computational expense. However, an obvious limitation of this approach is that it requires the model solution's analytic continuation to include singularities (or at least near-singular behaviour).

Recently, Subich [21] demonstrated a more general one-dimensional spectral moving mesh method, which uses a mesh density function that is given by the envelope of the

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

high-frequency components of the model solution. However, no direct justification is provided for why a solution envelope should correspond to a beneficial mesh density. Moreover, the threshold beyond which frequencies are considered to be high is chosen based on model-specific and interpolant-specific considerations. Nonetheless, using the local frequency content of a solution to form a mesh density function is an intuitive and general approach that has remained largely unexplored.

This paper presents a family of mesh density functions that are based on the local bandwidth of the model solution. This approach is justified by a nonuniform analogue of the Nyquist sampling theorem, and so does not depend on either problem-specific or interpolant-specific considerations. This makes them applicable to both spectral and non-spectral methods, as well as a wide range of problem types. The mesh density functions are demonstrated in this paper through application to function approximation using Chebyshev polynomials, and to a variety of acoustics problems. The acoustics problems are primarily solved using a periodic spectral moving mesh method, with some additional validation for periodic finite-difference moving mesh methods.

2 Bandwidth mesh density functions

2.1 Derivation

In this section, mesh density functions are derived from the local spatial frequency content of the solution to a model PDE. For notational purposes, it is useful to consider such a solution as a signal u that is a function of a spatial coordinate x with corresponding wavenumbers k. Sampling criteria for such signals are typically based on spatial frequency considerations. The most famous is the Nyquist-Shannon sampling theorem for band-limited signals, which states that perfect reconstruction requires a sampling rate of at least twice the signal's bandwidth, which is measured from DC to the maximum frequency present.[†] The Nyquist-Shannon sampling criterion is typically used with uniform sampling, but this is not strictly required. Band-limited signals with a spatially varying frequency content can be perfectly reconstructed using non-uniform samples taken at a rate equal to twice the local bandwidth [3]. Of course, not all signals possess a hard cut-off in their frequency content. Hence, it is useful to consider statistical measures of bandwidth.

Now let the signal u be 2π-periodic and normalised such that

$$\int_{-\pi}^{\pi} |u|^2 dx = \int_{-\infty}^{\infty} |U|^2 dk = 1,$$

where U is the Fourier transform of u. The local amplitude A, phase φ, and spatial frequency φ_x (subscripts denote differentiation) of this signal are inherently coupled when

[†]A similar criterion applies to signals whose power is zero below some lower bound on the absolute frequency (non-baseband signals), but this requires explicit knowledge of the lower bound, and so is less general.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

the signal is directly analysed. The analytic signal provides a way of decoupling them. It does this by attaching an imaginary counterpart which is in quadrature with the signal. This is typically done via the Hilbert transform[‡] $\mathcal{H}$, yielding

$$v = u + i\mathcal{H}u = Ae^{i\varphi}.$$

To define a local, statistical measure of bandwidth, the analytic signal is considered to have a joint position-wavenumber power density $P(k,x)$ which satisfies the marginals

$$P(x) = \int_{-\infty}^{\infty} P(k,x)dk = |v|^2, \qquad P(k) = \int_{-\pi}^{\pi} P(k,x)dx = |V|^2,$$

where V is the Fourier transform of v. Position-wavenumber power densities can be computed using a variety of transformations, including the short-time Fourier transform and the wavelet transform, but it suffices to consider the concept alone here. A useful family of statistical bandwidth measures are the even-order spectral moments

$$\langle k^{2m} \rangle = \int_{-\infty}^{\infty} k^{2m} P(k)dk. \tag{2.1}$$

The order of the spectral moments can be chosen based on the desired weighting that is to be given to the power density's tail. Since the $m=1$ moment corresponds to the variance (assuming symmetry in $P(k)$ about k=0), and the square root of the variance is commonly used as a measure of bandwidth, the second spectral moment is used for the derivation that follows. Similar derivations are easily made for other choices of m. From the global spectral moment, a local equivalent may be derived using the conditional power density

$$P(k|x) = \frac{P(k,x)}{P(x)}$$

as

$$\begin{aligned}
\langle k^2 \rangle &= \int_{-\infty}^{\infty} k^2 P(k)dk \\
&= \int_{-\infty}^{\infty} k^2 \left(\int_{-\pi}^{\pi} P(k,x)dx \right) dk \\
&= \int_{-\infty}^{\infty} k^2 \left(\int_{-\pi}^{\pi} P(k|x)P(x)dx \right) dk \\
&= \int_{-\pi}^{\pi} \left(\int_{-\infty}^{\infty} k^2 P(k|x)dk \right) P(x)dx \\
&= \int_{-\pi}^{\pi} \langle k^2 \rangle|_x P(x)dx.
\end{aligned} \tag{2.2}$$

[‡]Appendix A describes a method for computing a periodic Hilbert transform using nonuniformly sampled function values.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

Here, the local second spectral moment has been defined by [6]

$$\langle k^2 \rangle|_x = \int_{-\infty}^{\infty} k^2 P(k|x) dk. \tag{2.3}$$

This choice of definition is intuitive when $P(k|x)$ is considered as a local frequency distribution, and when (2.1) is compared with (2.3). While (2.3) could be used to compute the local bandwidth of a signal, it is convenient to take an alternative approach that avoids explicitly computing a joint power density.

An alternative way of computing local spectral statistics is to consider the operator

$$K = \begin{cases} \frac{1}{i}\frac{d}{dx} & \text{in the position representation,} \\ k & \text{in the wavenumber representation.} \end{cases}$$

For the wavenumber representation, this operator can be used to compute the global second spectral moment as

$$\langle k^2 \rangle = \langle V|K^2|V \rangle = \int_{-\infty}^{\infty} k^2 |V|^2 dk,$$

as expected. Similarly, for the position representation the global second spectral moment is given by [6]

$$\langle k^2 \rangle = \langle v|K^2|v \rangle = \int_{-\pi}^{\pi} \left| \frac{v_x}{v} \right|^2 |v|^2 dx. \tag{2.4}$$

Making a comparison with (2.2), the left term of the integrand in (2.4) is considered to be the local second spectral moment [4, 6]:

$$\langle k^2 \rangle|_x = \left| \frac{v_x}{v} \right|^2. \tag{2.5}$$

In [5], this interpretation of similar operators is justified by showing that it leads to established results for a number of quantum mechanical statistics. Taking the square root of (2.5) then gives a definition of the local bandwidth, which is used to define the first of two proposed mesh density functions: the *ordinary bandwidth mesh density function*

$$\rho = \left| \frac{v_x}{v} \right|. \tag{2.6}$$

A complication arises when this mesh density function is computed from an analytic signal v whose amplitude drops to zero, since computing the local bandwidth becomes ill-posed in these regions. One approach to regularisation is to include an amplitude-weighting. This leads to the *amplitude-weighted bandwidth mesh density function*, defined as

$$\rho = |v_x|. \tag{2.7}$$

The difference between the ordinary and amplitude-weighted bandwidth mesh density functions is as follows: The ordinary bandwidth mesh density function was derived with equal consideration given to every point on the function. Hence, the resultant sampling minimises local errors. The introduction of amplitude weighting will increase the sampling in regions with large amplitudes, and thereby reduce the overall absolute error.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

2.2 Examples

To examine the effect of mesh transformations resulting from the bandwidth mesh density functions, a Gaussian function and a Runge-type function are considered. These both have known Hilbert transforms, and so the ordinary and amplitude-weighted bandwidth mesh density functions can be explicitly computed. They also represent two important classes of function: The Gaussian is an entire function, and the Runge-type function is analytic in a neighbourhood of the real axis. Since neither are band-limited, they are both well-suited to the statistical bandwidth definitions given in Section 2.1, and are smooth enough to provide spectral accuracy when a spectral method is applied to their approximation. The spectral method that is used to approximate the function is a Chebyshev-type method, implemented using the Chebfun software package [7], which adaptively represents functions to near machine precision using Chebyshev polynomials.

For the first example, consider the following Gaussian function and its Hilbert transform

$$u = \exp(-x^2), \qquad \mathcal{H}u = \frac{2F(x)}{\sqrt{\pi}}.$$

Here, F is the Dawson integral. The derivative of the corresponding analytic signal $v = u + i\mathcal{H}u$ is

$$v_x = -2x\exp(-x^2) + i\frac{2-4xF(x)}{\sqrt{\pi}}.$$

From this, the ordinary and amplitude-weighted bandwidth mesh density functions were computed via formulae (2.6) and (2.7). Mesh transformations were then generated using Chebfun by numerically computing (and scaling/translating) the indefinite integrals of the reciprocal of the mesh density functions on the domain $x \in [-15,15]$. The original function u was then composed with each mapping, and the Chebyshev coefficients corresponding to the original function and its transformed counterparts were computed. These were used as a proxy for the approximation error, and compared to examine the effect of the transformations on the convergence rate of each Chebyshev interpolant. Fig. 1 depicts the original and transformed functions, along with their Chebyshev coefficients.

Two convergence regimes are evident in Fig. 1. The untransformed Gaussian function exhibits a convergence rate that is faster than geometric (consistent with it being entire), while the transformed Gaussian functions both exhibit geometric convergence. Thus, the transformed meshes provide benefit for low-frequency spectral components, but not for higher-frequency ones, and so their effectiveness depends on the desired accuracy of the spectral approximation.

A similar investigation was carried out for the arclength and curvature monitor functions, which are respectively defined by

$$\rho = \sqrt{1+|u_x|^2}, \qquad \rho = \left(1+|u_{xx}|^2\right)^{\frac{1}{4}}. \tag{2.8}$$

Their effectiveness was assessed by the number of Chebyshev coefficients that were needed to represent the original Gaussian under mesh transformations resulting from their use,

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

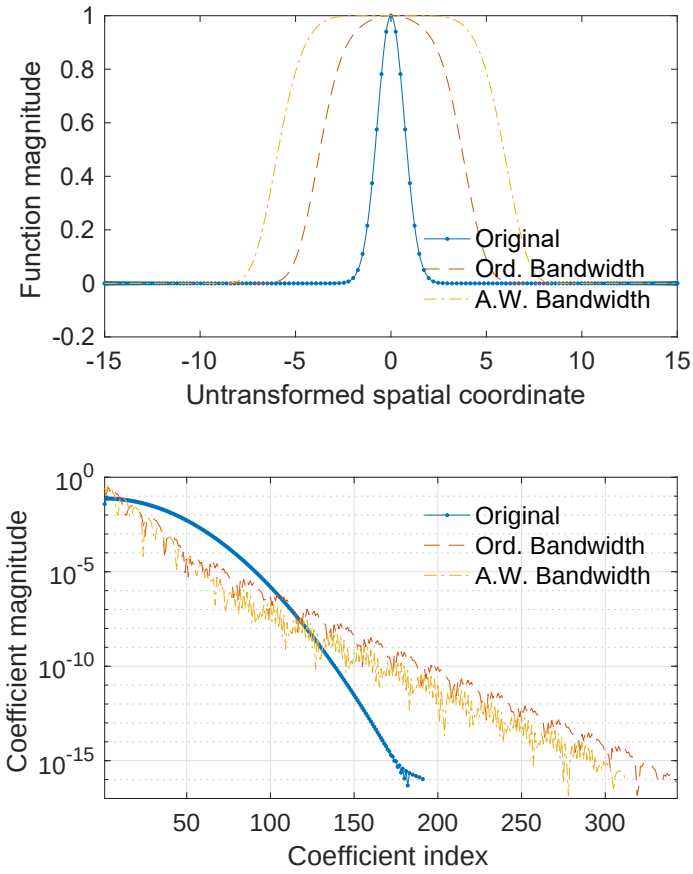


Figure 1: Chebyshev interpolation of a Gaussian function under mesh transformations corresponding to the ordinary and amplitude-weighted bandwidth mesh density functions. (Top) Function values plotted against an untransformed spatial coordinate. The widening that results from composing the Gaussian with each mesh transformation corresponds to an increased function sampling density. (Bottom) Chebyshev coefficients corresponding to the functions depicted in top subfigure. The convergence rate for the original function is faster than geometric, while the transformed functions exhibit geometric convergence. For this Gaussian function, the transformations provide some benefit for low accuracy Chebyshev interpolants only.

computed as described above. For the original Gaussian function, this number was 183. For the arclength- and curvature-based transformations, they were 1,021 and 1,513 respectively. This drastic increase is likely to be the result of a lack of smoothness in the arclength and curvature mesh density functions themselves, and can be alleviated by through a secondary smoothing step (as is often done in moving mesh methods), as discussed in Section 3, below. Nonetheless, these results illustrate the limitation of applying the arclength and curvature mesh density functions to spectral methods, particularly when there are no large, localised variations in the approximated function's first or second derivatives.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

For the second example, consider the following Runge-type function and its Hilbert transform

$$u = \frac{1}{1+25x^2}, \qquad \mathcal{H}u = \frac{5x}{1+25x^2}.$$

The corresponding analytic signal and derivative are

$$v = \frac{i}{5x+i}, \qquad v_x = -\frac{5i}{(5x+i)^2}.$$

As above, the decay rate of Chebyshev coefficients was used to assess the effectiveness of the bandwidth mesh density functions in producing mesh transformations, this time on the interval $x \in [-1,1]$. In addition, a mesh was generated using the singularity-based approach of Tee et al. [11, 12, 23, 24], discussed in Section 1. This method makes use of the function's known singularities at $\pm 0.2i$. Fig. 2 depicts the original and transformed functions, along with their Chebyshev coefficients.

Unlike for the Gaussian function, all three mesh transformations provide an improvement in the convergence rate of the Runge-type function's Chebyshev coefficients at all frequency scales. There is also a significant difference between the convergence rates under each mesh transformation. In order of effectiveness, these are the singularity-based mesh transformation, followed by the transformations resulting from the amplitude-weighted and ordinary bandwidth mesh density functions. The performance gap between the singularity-based technique and the bandwidth mesh density functions is significant. However, for many approximated functions, such as those generated numerically as solutions to PDEs, exact singularity locations are unknown, and must be approximated. This reduces the performance gap significantly, as will be shown in Section 3.3.

Finally, the arclength and curvature monitor functions were again examined based on the number of Chebyshev coefficients that was required to represent the Runge-type function under transformations resulting from their use. For the original Runge-type function, this number was 189. For the arclength- and curvature-based transformations, they were 1,329 and 7,415 respectively. As before, this illustrates the limitation of applying such mesh density functions to spectral methods, especially since the Runge-type function also exhibits no obvious localised, derivative-based features.

3 Numerical experiments

3.1 Numerical methods

To examine the application of the bandwidth mesh density functions to PDEs, two moving mesh methods are used. These are outlined below, but it is important to note that the bandwidth mesh density functions are agnostic to the algorithmic choices that have been made. The two moving mesh methods are differentiated by the numerical method

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

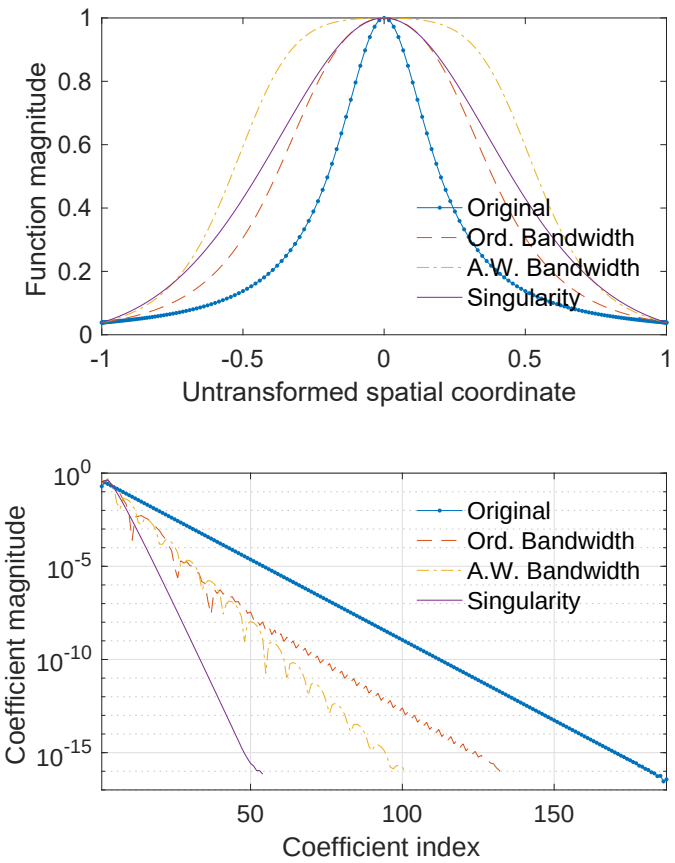


Figure 2: Chebyshev interpolation of a Runge-type function under mesh transformations corresponding to the ordinary and amplitude-weighted bandwidth mesh density functions, and the singularity-based mesh mapping of [11, 12]. (Top) Function values plotted against an untransformed spatial coordinate. The widening that results from composing the Runge-type function with each mesh transformation corresponds to an increased function sampling density. (Bottom) Chebyshev coefficients corresponding to the functions depicted in top subfigure. All exhibit geometric convergence rates, and the mesh transformations are beneficial in all cases.

they use for computing spatial gradients in the model/mesh PDEs. Both assume periodic model solutions, but one uses spectral interpolants and the other uses centred finite-differences. For the spectral interpolants, gradients are computed using either a collocating Fourier interpolant (when taken with respect to the computational coordinate) or a rational trigonometric interpolant [1,23] (when taken with respect to the physical coordinate). For the finite-difference method, gradients are computed using centred finite-differences of various accuracy-orders taken with respect to the computational coordinate. Physical gradients are then computed using the chain rule

$$\frac{\partial}{\partial x} = \frac{\partial s}{\partial x}\frac{\partial}{\partial s},$$

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

where x and s are the physical and computational coordinates, respectively. The notation $x=x(t,s)$ is used to refer to x as both the physical coordinate itself, and as the time-varying transformation between the physical and computational coordinates.

Mesh node relocation is controlled using the moving mesh PDE (MMPDE) [13]

$$x_t=\tau^{-1}(\rho x_s)_s.$$

The function $\rho(x)$ is the solution-dependent mesh density specification, which the MMPDE satisfies as $t\to\infty$. The rate at which this happens is controlled by the scalar parameter τ, which must match the time-scale over which solution features evolve. Note that to compute the mesh gradient x_s, a slight adjustment must be made to the approaches described in the preceding paragraph. Gradients are computed via the formula $x_s=(x-s)_s+1$ (assuming x and s share a domain size). This is done because $x-s$ is periodic for the work presented in this paper, while x is not. Strictly speaking, this approach may not guarantee monotonicity in the implied continuous mesh mapping, but in practice a well-sampled mesh density function ensures that this is the case.

A mesh density function is usually spatially smoothed before being applied to a MMPDE. This is done for a number of reasons. First, it ensures that the mesh density function is well-sampled by the mesh, and hence that the MMPDE can be efficiently solved. Second, it ensures smooth mesh transformations, which in turn produce fast convergence rates when the model is solved. Third, it lessens the inherent increase in stiffness that comes with mesh adaptivity, improving the efficiency of time-stepping algorithms [14]. Let $\tilde{\rho}$ and ρ be the smoothed and unsmoothed mesh density functions respectively. These are related by the equation

$$\tilde{\rho}-\beta^{-2}\tilde{\rho}_{ss}=\rho, \tag{3.1}$$

where the parameter β controls the degree of smoothing [14, 15]. With homogeneous Neumann boundary conditions, (3.1) constrains the relative rate of change in the smoothed mesh density by $|\tilde{\rho}_s|/\tilde{\rho}\leq\beta$, and similar behaviour is observed for periodic boundary conditions. Equation (3.1) is solved using a Fourier interpolant via

$$\tilde{\rho}=\mathcal{F}^{-1}\left\{\frac{\mathcal{F}\{\rho\}}{1+\beta^{-2}k^2}\right\},$$

where k are wavenumbers corresponding to s. The smoothing parameter can be chosen to be discretisation-dependent, so that it has a similar effect to nearest-neighbour smoothing. This choice aims to ensure that the mesh density function is well-sampled, since, for example, a well-sampled peak will be smoothed (and reduced) less than a poorly sampled one. Unless otherwise stated, here $\beta=(\Delta s\sqrt{2})^{-1}$ [14], where Δs is the discretisation size in the computational coordinate. A numerical experiment is described in Appendix B, which analyses this choice for one of the model problems presented below.

The model and mesh PDEs are coupled together using a quasi-Lagrange approach. This replaces the usual time-derivatives in the model PDE with an advective derivative

$$\frac{d}{dt}=\frac{\partial}{\partial t}+\frac{\partial x}{\partial t}\frac{\partial}{\partial x}.$$

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

It is notable that this yields an implicit system of PDEs, and that the coupled model/mesh system may be stiffer than either PDE individually. Time-stepping is performed using the method of lines, with Matlab's `ode15i` function [19] used to integrate the system of ordinary differential equations that results from the spatial discretisation previously discussed. This algorithm solves ODEs of the form $f(t,y,y_t)=0$ using adaptive-order backward differentiation formulae, with an adaptive timestep size. Here, y is a vector of solution values and mesh node positions, and $y_t=dy/dt$.

3.2 Example problems

The bandwidth mesh density functions are demonstrated through application to four problems and three acoustic models. These each exhibit different feature types. The first problem is based on a heterogeneous advection equation, and exhibits the formation and propagation of a sharp crest. The second problem is based on the viscous Burgers' equation, and exhibits the formation of a stationary shock front. These problems are used to demonstrate the performance of spectral and finite difference moving mesh methods in Section 3.3 and Section 3.4, respectively. The third problem is also based on the viscous Burgers' equation, and exhibits the formation, propagation, and merging of multiple shock fronts. The fourth problem is based on the Korteweg-de Vries equation, and exhibits the formation of multiple solitons, and their subsequent interactions. All use dimensionless units and a periodic domain $x\in[-\pi,\pi)$. The time-stepping algorithm was provided with relative and absolute error tolerances of 10^{-9} and 10^{-10}, to ensure that errors in the spatial numerical method dominate. The mesh speed parameter is $\tau=10^{-2}$. For the illustrations presented in Figs. 3-6 below, the amplitude-weighted bandwidth mesh density function was used in conjunction with the spectral moving mesh method described in Section 3.1.

3.2.1 A heterogeneous advection equation

The first model presented in this section is an advection equation with a heterogeneous sound speed:

$$u_t-u_xx_t=c(x)u_x,\qquad c(x)=[1+0.9\cos(x)]^{-1}.$$

Recall that the usual time-derivative has been modified in the above expression to account for the movement of the mesh nodes. This model describes linear wave propagation, with a propagation speed that is slower in the middle of the domain than the edges. It is solved using a sinusoidal initial condition

$$u(0,x)=\cos(x-\pi),$$

and the resulting wave is propagated until $t=2\pi$, when it has travelled the full length of the spatial domain and periodic wrapping begins to occur. For this problem, the initial and final waveforms are equal, and can be compared to measure the accuracy of a given simulation. The heterogeneous sound speed causes the peak in the wave to sharpen as

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

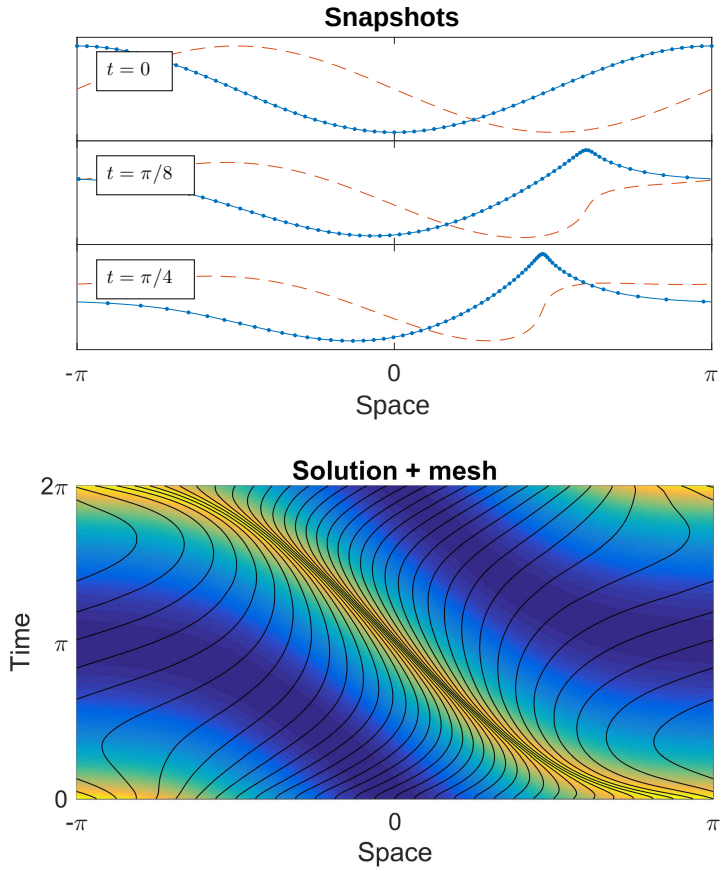


Figure 3: A solution to the heterogeneous advection equation. (Top) Snapshots showing the formation of a sharp wave crest. The solution and its computed Hilbert transform are represented by solid and dashed lines respectively. Dots indicate mesh nodes. (Bottom) A combined solution/mesh plot. Colours indicate the model solution (yellow high, blue low), and the trajectories of mesh nodes are shown as black lines. The mesh has been downsampled to $N=32$ nodes for clarity.

it propagates through the centre of the domain, making an adaptive mesh beneficial. A solution to this problem is depicted in Fig. 3, computed using $N=64$ mesh nodes. The snapshots in the upper subplot show the formation of the sharp wave crest, and the lower subplot shows the trajectories of the mesh nodes, which cluster densely around this crest.

3.2.2 The viscous Burgers' equation

The second model presented in this paper is the viscous Burgers' equation. Within a moving mesh framework, it is given by

$$u_t - u_x x_t = uu_x + \varepsilon u_{xx}.$$

The nonlinear term causes this model to evolve shock fronts, with their severity controlled by diffusion at a rate given by ε.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

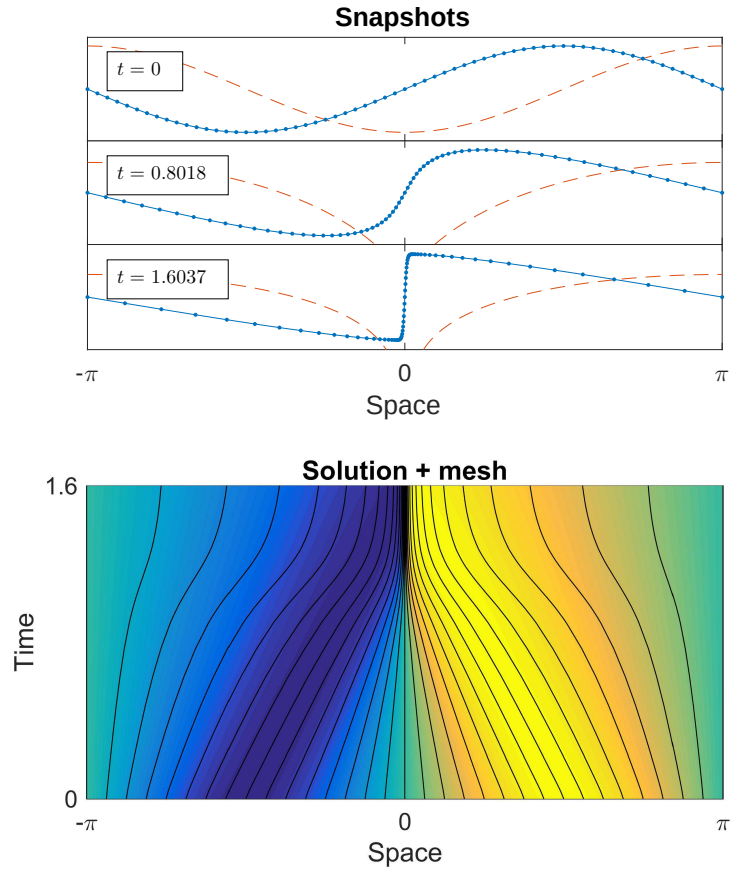


Figure 4: A solution to the viscous Burgers' equation. (Top) Snapshots showing the formation of a steep, stationary shock front. The solution and its computed Hilbert transform are represented by solid and dashed lines respectively. Dots indicate mesh nodes. (Bottom) A combined solution/mesh plot. Colours indicate the model solution (yellow high, blue low), and the trajectories of mesh nodes are shown as black lines. The mesh has been downsampled to $N=32$ nodes for clarity.

The first of two problems that use Burgers' equation exhibits a single, stationary shock front. It is given by the following initial condition and diffusion coefficient:

$$u(0,x)=\sin(x), \qquad \varepsilon=10^{-2}.$$

(This problem is modified from that described in [10] to suit a domain of length 2π.) The simulation is terminated at $t=1.6037$, which is approximately when the shock front is steepest. A solution to this problem is depicted in Fig. 4, computed using $N=64$ mesh nodes. The snapshots in the upper subplot show the formation of the shock front, and the mesh node trajectories in the lower subplot smoothly converge around this shock front.

The second problem that uses Burgers' equation exhibits seven propagating shock fronts that merge over time. It is given by the following initial condition and diffusion

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

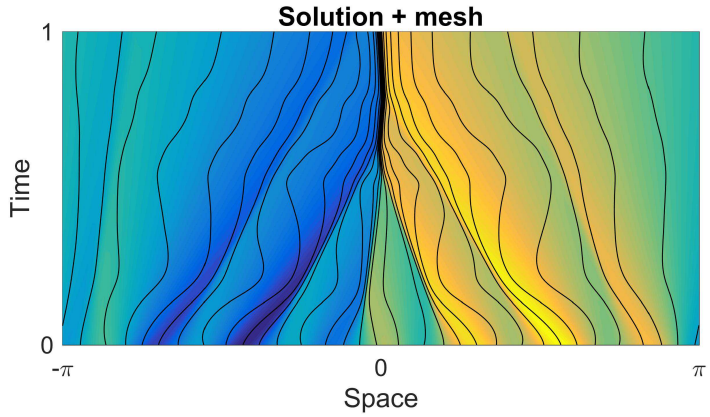


Figure 5: A combined solution/mesh plot for Burgers' equation. Colours indicate the model solution (yellow high, blue low), and the trajectories of mesh nodes are shown as black lines. The mesh has been downsampled to $N=32$ nodes for clarity. The mesh nodes track each shock front.

coefficient:

$$u(0,x)=2\sin(x)+\cos(7x), \qquad \varepsilon=10^{-2}.$$

This simulation is terminated at $t=1$, when most of the shock fronts have merged, and the remainder have diffused significantly. A solution to this problem is depicted in Fig. 5, computed using $N=128$ mesh nodes. The mesh nodes follow each wavefront smoothly, and become denser as the central shock front increases in severity. This kind of problem was identified by Hale [11] as being problematic for singularity-based methods, which fail to generate a mesh when singularities coalesce (in this problem, each wavefront corresponds to a set of singularities).

3.2.3 The Korteweg-de Vries equation

The third model presented in this paper is the Korteweg-de Vries equation, which combines nonlinear wave propagation with dispersion. Within a moving mesh framework, it is given by

$$u_t-u_xx_t=\alpha uu_x+\beta u_{xxx}.$$

The Korteweg-de Vries model admits solitons in its solutions. These are waveforms whose size, shape, and velocity are constant provided they remain well separated. Their speed is amplitude-dependent, and when two solitons interact the faster soliton is shifted forwards and the slower soliton is shifted back. After interacting, solitons regain their original shape. It is not clear that solitons are difficult to resolve since they appear visually smooth, but their analytic continuations are known to include singularities that limit convergence rates [23]. The initial condition and parameters

$$u(0,x)=\cos(x), \qquad \alpha=-\pi, \qquad \beta=-(0.022)^2\pi^3,$$

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

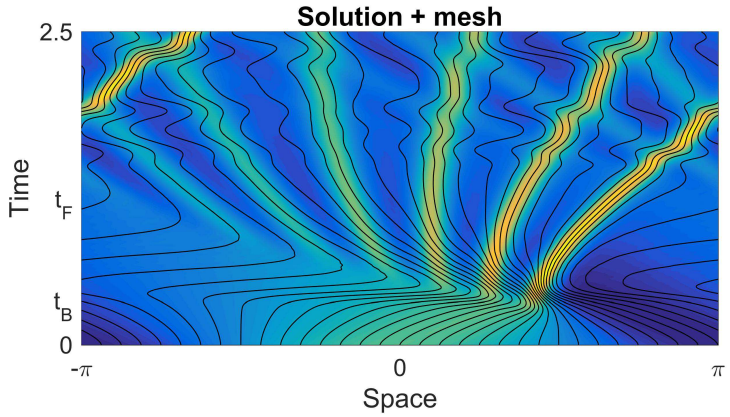


Figure 6: A combined solution/mesh plot for the Korteweg-de Vries equation. Colours indicate the model solution (yellow high, blue low), and the trajectories of mesh nodes are shown as black lines. The mesh has been downsampled to $N=32$ nodes for clarity. The mesh nodes compress around each soliton, and smoothly track them as they interact.

were chosen to match the problem presented in [26], modified to suit a domain of length 2π. The initial condition first steepens, before dispersion causes a number of solitons to form and propagate. Soliton formation begins at approximately $t_B=1/\pi$, and completes at around $t_F=3.6t_B$. After this, the solitons propagate until the periodic boundary conditions cause them to interact. The simulation was terminated at $t=8t_B$. A solution to this problem is depicted in Fig. 6, computed using $N=128$ mesh nodes. The mesh nodes compress around each soliton, and continue to track them as they interact.

3.3 Spectral convergence rates

This section provides a performance evaluation for the bandwidth mesh density functions when applied to the periodic spectral method described in Section 3.1. They are judged by the rates at which approximated solutions to the problems in Section 3.2 converge as the number of mesh nodes increases, measured using the maximum absolute error in the final simulated solution. This illustrates the ability of meshes resulting from the bandwidth mesh density functions to reduce the trade-off between computational resource usage and accuracy.

First, the advection problem was solved with a varying number of mesh nodes. Three different mesh specifications were used: one uniform specification, and two based on the ordinary and amplitude-weighted bandwidth mesh density functions. To evaluate their performance, the initial and final waveforms were interpolated onto 10,001 uniformly distributed mesh nodes and compared, since they should be equal to one another. Fig. 7 depicts these results. Both adaptive meshes produced error convergence rates which are more than five times faster than those produced by the uniform mesh. It also seems that there is a slight advantage to using the amplitude-weighted bandwidth mesh density

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

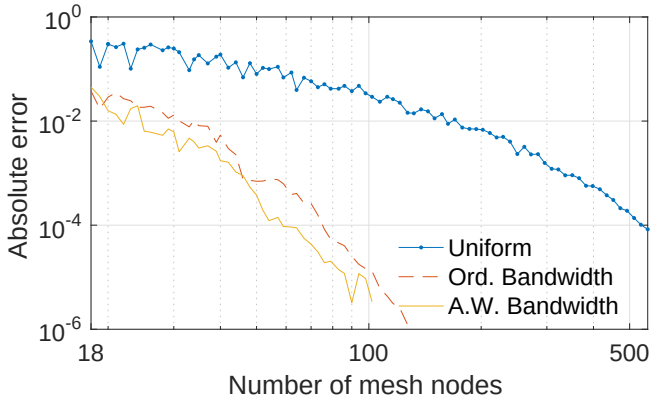


Figure 7: The error in the final waveform for the advection simulation depicted in Fig. 3 computed using a spectral moving mesh method. The solutions computed using the bandwidth mesh density functions exhibit much faster convergence rates that those that used a uniform mesh.

function over the ordinary one. The arclength and curvature mesh density functions (2.8) were also applied to the advection problem, but produced unstable time-stepping due to their lack of smoothness.

A similar evaluation was performed for the first Burgers' equation problem (which exhibits one shock front). A set of uniform mesh results were first generated for varying N, computed to near machine precision using Chebfun's `spin` algorithm [7]. Results were then computed for four adaptive methods. The first was the singularity-based method of Tee et al. [11,12,23,24]. This was applied with odd numbers of nodes in the range $N=15$ to $N=99$ (odd N ensures a node at $x=0$ for their implementation). The remaining results were computed using the spectral moving mesh method described in Section 3.1, in conjunction with the arclength, ordinary bandwidth, and amplitude-weighted bandwidth mesh density functions. These used even node numbers in the range $N=16$ to $N=100$. All approximated solutions were interpolated onto 10,001 uniformly distributed mesh nodes and compared to the results computed using the `spin` algorithm when the uniform mesh was at its densest. Fig. 8 depicts this comparison. The uniform mesh produces very slow convergence for this problem. The arclength mesh density function provides a significant improvement, but nonetheless converges far more slowly than the remaining methods. The singularity-based method and the two bandwidth mesh density specifications clearly perform best. These all produced error convergence rates that were more than an order of magnitude faster than those produced by a uniform mesh, and two to three times faster than those using the arclength mesh density function. Once again, the amplitude-weighted bandwidth mesh density function outperforms its ordinary counterpart, this time by a larger margin. It also outperforms the singularity-based approach, though by a smaller margin. As with the advection problem, the curvature mesh density function was also applied to the Burgers' equation problem, but was again non-smooth, causing instability in the simulation.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

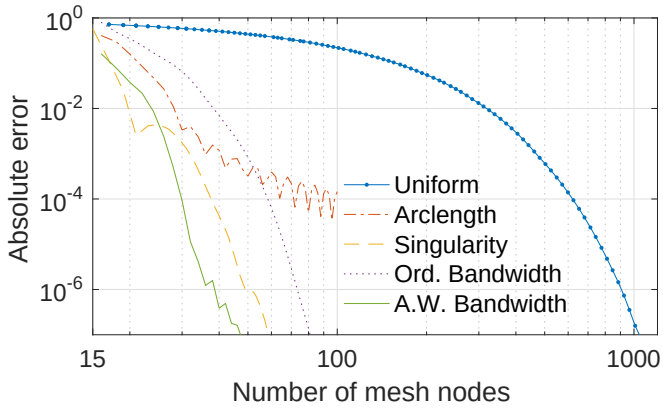


Figure 8: Convergence plots for solutions to the viscous Burgers' equation simulation depicted in Fig. 4 using a spectral moving mesh method. The bandwidth mesh density functions clearly produce faster convergence rates than a uniform mesh specification or the arclength mesh density function, and converge at a similar rate to the singularity-based method of Tee et al. [11, 12, 23, 24].

In addition to the results presented in Fig. 8, results were gathered using the amplitude of the solution's analytic signal as a mesh density function. It was found that this produced a small benefit over a uniform mesh. This is likely because the feature of interest is a steep gradient, which produces an analytic signal with an infinite amplitude in the limit as the gradient's magnitude increases to infinity. This may help to explain why the amplitude-weighted bandwidth mesh density function outperforms the ordinary bandwidth mesh density function for this problem. In contrast, when the amplitude was used as a mesh density function for the advection problem, no benefit was found because the amplitude of the solution's analytic signal is approximately constant. This illustrates the problem specificity of amplitude-weighting, and may motivate alternative mesh density functions for other applications, for instance one which includes an amplitude weighting for small amplitudes only.

3.4 Finite-difference convergence rates

This section provides a performance evaluation for the bandwidth mesh density functions when applied to the periodic finite-difference moving mesh method described in Section 3.1. This highlights the fact that the frequency considerations on which the bandwidth mesh density functions are based are relevant for piecewise polynomial interpolants as well as spectral ones.

First, the performance evaluation for the advection equation was repeated. Three sets of results were computed using the amplitude-weighted bandwidth mesh density function, with each corresponding to a different accuracy-order for the finite differences. Fig. 9 depicts these results. The spectral convergence rates have been replaced with algebraic ones, as expected, and as the accuracy-order of the finite-difference method in-

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

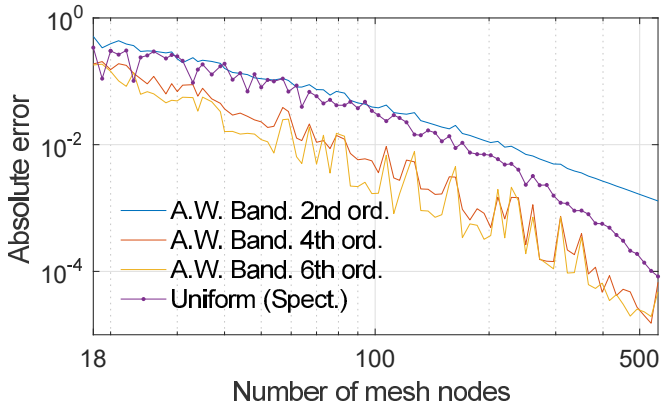


Figure 9: The error in the final waveform for the advection simulation depicted in Fig. 3 computed using centred, periodic finite-difference moving mesh methods. The numbers in the legend indicate the accuracy-order of the finite-difference method that was used. Adaptive meshes computed using the amplitude-weighted bandwidth mesh density function improve the performance of these finite difference methods to the point where they're comparable with a spectral method on a uniform mesh.

creases, the accuracy improves. Fig. 9 also depicts the previous spectral results for a uniform mesh. It is clear that the introduction of mesh adaptation can improve the performance of finite-difference methods to such an extent that they exceed that of the uniform spectral method.

Second, the performance evaluation for Burgers' equation was repeated using the finite-difference moving mesh method. In contrast to the results in Section 3.3, only the arclength and amplitude-weighted bandwidth mesh density functions were examined. Fig. 10 depicts these results. The spectral convergence rates have been replaced with algebraic ones, as expected, and as the accuracy-order of the finite-difference method increases, the accuracy improves noticeably in almost all cases. Comparing each finite-difference method, it is clear that the amplitude-weighted bandwidth mesh density function improve upon the results obtained using the arclength mesh density function significantly. Fig. 10 also depicts the previous spectral results for a uniform mesh, and it is clear that the adaptive meshes drastically outperform a uniform mesh in this case.

4 Summary and discussion

Two new mesh density functions have been presented and applied to both static function approximation, and to a number of time-varying model problems in acoustics. They are called the bandwidth mesh density functions, and are derived from the local spatial frequency content of the model solution. Their performance compares favourably with other approaches, and they are applicable to a wide range of problem types. In particular, they have been shown to outperform the arclength mesh density function and to match

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

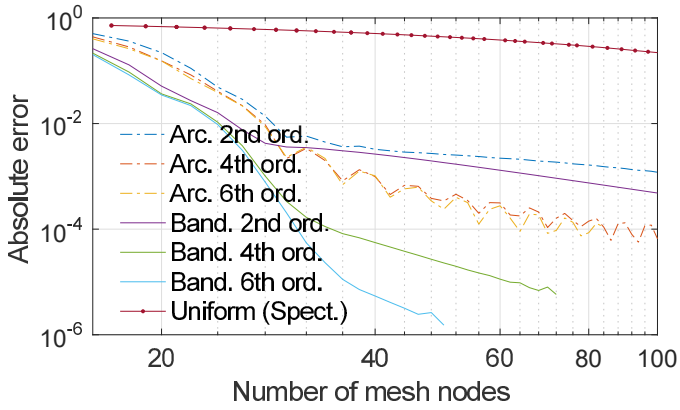


Figure 10: Convergence plots for solutions to the viscous Burgers' equation simulation depicted in Fig. 4 using centred, periodic finite-difference moving mesh methods. The numbers in the legend indicate the accuracy-order of the finite-difference method that was used. The amplitude-weighted bandwidth mesh density function clearly produces faster convergence rates than the arclength mesh density function.

the performance of singularity-based methods. These numerical experiments were conducted using Chebyshev polynomials, and both a periodic spectral moving mesh method and a periodic finite difference moving mesh method, with good performance observed for all three.

The bandwidth mesh density functions presented in this paper are not the only ones that could be derived from consideration of the local spatial frequency content of a solution. Possible alternative choices that could be made include:

- Using a different order of spectral moment, or using a different spectral statistic altogether;
- Decoupling the amplitude and phase of signals using an approach that doesn't involve the analytic signal [17];
- Computing the local statistics using an explicit position-wavenumber power density;
- Regularising the ordinary bandwidth mesh density function by adding a small constant to the denominator, or by introducing amplitude-weighting for small amplitudes only.

It is also pointed out in [21] that not all frequency components in a solution will result in aliasing error in spectral collocation methods. Hence, it may be useful to apply a high-pass filter as a preconditioning step before computing bandwidth mesh density functions.

The bandwidth mesh density functions can be extended to multidimensional problems using the Riesz transforms. For a d-dimensional signal, the Riesz transforms yield

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

d components which can be combined with the original signal to form what is called a monogenic signal. A similar approach to that presented in this paper may then be used to compute a multi-dimensional equivalent of the bandwidth mesh density functions. The focus of future work will be on investigating this, and on applying the resultant monitor functions to large-scale medical acoustics simulations.

Acknowledgments

The authors would like to thank the EPSRC for funding this work.

Appendices

A An algorithm for computing the periodic Hilbert transform of a function from nonuniform samples

The bandwidth mesh density functions require a complex-valued analytic signal to be defined from the real-valued solution to a model PDE. In one dimension, this is typically done using the Hilbert transform, which can be defined in the frequency domain by

$$\mathcal{H}u = \mathcal{F}^{-1}\{(-i\text{sign}(k))\mathcal{F}u\}.$$

For equispaced samples, this is easily approximated using fast Fourier transforms. For nonuniform samples, computing a Fourier transform is not so straightforward. An alternative is to use its definition as a convolution

$$\mathcal{H}u = \frac{1}{2\pi}\text{p.v.}\int_{-\pi}^{\pi} u(\tau)h(\tau-x)d\tau, \qquad h(x) = \cot\left(\frac{x}{2}\right), \tag{A.1}$$

where p.v. indicates the Cauchy principal value, and h is the circular Hilbert kernel.

When computing this integrand, special consideration needs to be made for the singularity at $\tau = x$. As described in [2], this singularity can be made removable by rewriting (A.1) as

$$\mathcal{H}u = \frac{1}{2\pi}\int_{-\pi}^{\pi}(u(\tau)-u(x))h(\tau-x)d\tau + \frac{u(x)}{2\pi}\text{p.v.}\int_{-\pi}^{\pi}h(\tau-x)d\tau. \tag{A.2}$$

The singularity is now present only in the second integral; the first integral is instead of indeterminate form at $\tau = x$. The singularity is easily dealt with by noting that the circular Hilbert kernel is antisymmetric and 2π periodic, meaning the Cauchy principal value of the second integral is equal to zero.

The indeterminate point in the first integral in (A.2) is evaluated by computing the limit of the integrand as $\tau \to x$. To do so, note that the Hilbert kernel can be obtained by

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

making the Cauchy kernel, defined as $1/x$, 2π-periodic:

$$\cot\left(\frac{x}{2}\right)=2\left[\frac{1}{x}+\sum_{n=1}^{\infty}\left(\frac{1}{x-2\pi n}+\frac{1}{x+2\pi n}\right)\right].$$

Now let $t=\tau-x$ and $f(t)=u(\tau)-u(x)$. From the first integrand in (A.2), write

$$f(t)h(t)=2f(t)\left[\frac{1}{t}+\sum_{n=1}^{\infty}\left(\frac{1}{t-2\pi n}+\frac{1}{t+2\pi n}\right)\right].$$

As $t\to 0$, each of the terms in the sum will cancel one another. Thus,

$$\lim_{t\to 0}f(t)h(t)=\lim_{t\to 0}\frac{2f(t)}{t}.$$

Because $f(0)=0$, this fraction is indeterminate. It can be solved using L'Hospital's rule:

$$\lim_{t\to 0}f(t)h(t)=2f'(0).$$

Returning to the original variables, this is written as

$$\lim_{\tau\to x}[(u(\tau)-u(x))h(\tau-x)]=2u'(\tau),$$

which can be substituted into the left integrand in (A.2) when it is solved.

Finally, to compute the Hilbert transform, a change of variables is made from x to s. Letting $\hat{u}=\mathcal{H}u$, this gives

$$\hat{u}(x)=\frac{1}{2\pi}\int_{-\pi}^{\pi}F(x,\tau)\tau_s ds,\qquad F=\begin{cases}(u(\tau)-u(x))h(\tau-x), & \tau\neq x,\\ 2u_\tau, & \tau=x.\end{cases}$$

This is solved using the trapezoid rule, discretised at a set of equispaced, discrete s nodes. Let subscripts now denote the index of a variable, rather than differentiation. The variable τ is discretised equal to x, and so the resultant expression is

$$\hat{u}_i=\frac{1}{2\pi}\sum_j F_{i,j}\left(\frac{\partial x}{\partial s}\right)_j\Delta s,\qquad F_{i,j}=\begin{cases}(u_j-u_i)h(x_j-x_i), & i\neq j,\\ 2\left(\frac{\partial u}{\partial x}\right)_i, & i=j.\end{cases}$$

B Analysis of the mesh density smoothing parameter

One free parameter in many moving mesh methods is the degree of smoothing that is applied to a computed monitor function prior to its application within a MMPDE. For the advection problem outlined in Section 3.2.1 (and examined in Sections 3.3-3.4), the mesh smoothing parameter's effect was investigated by varying it from $\beta=1$ to $\beta=20$.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

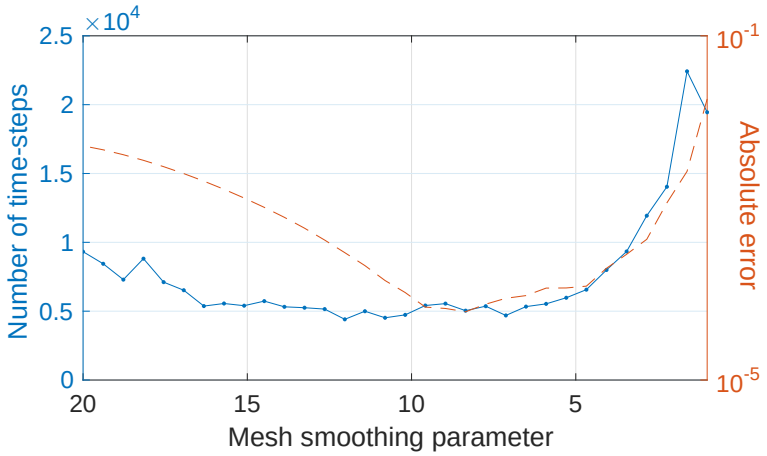


Figure 11: Error and number of timesteps for the advection simulation depicted in Fig. 3 with a varying smoothing parameter. Smoothing increases from left to right. Both the number of timesteps required (solid line) and the error (dashed line) decrease slightly as the mesh density function is increasingly smoothed, before turning around and increasing at a faster rate. Beyond the turning point, the mesh density function may become so smooth that the resulting mesh is no longer beneficial. Note that the minimum error is achieved for a smoothing parameter that is very close to the discretisation-dependent choice $\beta=(\Delta s\sqrt{2})^{-1}$ which equals 7.2 for this example.

Fig. 11 depicts the error in the resulting solution, as well as the number of timesteps that were taken. As the mesh density function is increasingly smoothed, both the number of timesteps required and the error slowly decrease, before turning and increasing at a faster rate. This reflects the fact that the mesh density function has been smoothed to a point where it no longer produces a beneficial mesh [18]. The turning point in the error is close to the smoothing parameter choice mentioned in Section 3.1, which is $\beta=(\Delta s\sqrt{2})^{-1}\approx 7.2$ for this example.

References

[1] R. Baltensperger. Some results on linear rational trigonometric interpolation. *Computers & Mathematics with Applications*, 43(6-7):737–746, 2002.

[2] Zhong Chen and YongFang Zhou. An efficient algorithm for solving hilbert type singular integral equations of the second kind. *Computers & Mathematics with Applications*, 58(4):632–640, 2009.

[3] James J. Clark, Matthew R. Palmer, and Peter D. Lawrence. A transformation method for the reconstruction of functions from nonuniformly spaced samples. *IEEE Trans. Acoust., Speech, and Signal Process.*, 33(5):1151–1165, Oct 1985.

[4] L. Cohen. Instantaneous 'anything'. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 105–108, Washington, DC, USA, 1993. IEEE Computer Society.

[5] Leon Cohen. Local values in quantum mechanics. *Phys. Lett. A*, 212(6):315–319, 1996.

[6] Keith L. Davidson and Patrick J. Loughlin. Instantaneous spectral moments. *J. Franklin Inst.*, 337(4):421–436, 2000.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press

[7] T. A Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide*. Pafnuty Publications, 2014.
[8] W. M. Feng, P. Yu, S. Y. Hu, and Z. K. Liu. A fourier spectral moving mesh method for the cahn-hilliard equation with elasticity. *Commun. Comput. Phys.*, 5(2-4):582–599, 2009.
[9] W. M. Feng, P. Yu, S. Y. Hu, Z. K. Liu, Q. Du, and L. Q. Chen. Spectral implementation of an adaptive moving mesh method for phase-field equations. *J. Comput. Phys.*, 220(1):498–510, Dec 2006.
[10] Herv Guillard and Roger Peyret. On the use of spectral methods for the numerical-solution of stiff problems. *Computer Methods in Applied Mechanics and Engineering*, 66(1):17–43, 1988.
[11] Nicholas Hale. *On The Use Of Conformal Maps To Speed Up Numerical Computations*. PhD thesis, University of Oxford, 2009.
[12] Nicholas Hale and T. Wynn Tee. Conformal maps to multiply slit domains and applications. *SIAM J. Sci. Comput.*, 31(4):3195–3215, 2009.
[13] Weizhang Huang, Yuhe Ren, and Robert D. Russell. Moving mesh partial differential equations (mmpdes) based on the equidistribution principle. *SIAM J. Numer. Anal.*, 31(3):709–730, 1994.
[14] Weizhang Huang and Robert D. Russell. Analysis of moving mesh partial differential equations with spatial smoothing. *SIAM J. Numer. Anal.*, 34(3):1106–1126, 1997.
[15] Weizhang Huang and Robert D. Russell. *Adaptive moving mesh methods*, volume 174 of *Applied Mathematical Sciences*. Springer Science+Business Media, 223 Spring Street, New York, NY 10013, USA, 2011.
[16] Jiri Jaros, Alistair. P. Rendell, and Bradley E. Treeby. Full-wave nonlinear ultrasound simulation on distributed clusters with applications in high-intensity focused ultrasound. *Int. J. High Perform. Comput. Appl.*, 30(2):137–155, 2016.
[17] Patrick J. Loughlin and Berkant Tacer. On the amplitude- and frequency-modulation decomposition of signals. *J. Acoust. Soc. Am.*, 100(3):1594–1601, 1996.
[18] L. S. Mulholland, W. Z. Huang, and D. M. Sloan. Pseudospectral solution of near-singular problems using numerical coordinate transformations based on adaptivity. *SIAM J. Sci. Comput.*, 19(4):1261–1289, 1998.
[19] L. F. Shampine. Solving 0 = f (t, y (t), y(t)) in matlab. *J. Numer. Math.*, 10(4):291–310, 2002.
[20] Jie Shen and Xiaofeng Yang. An efficient moving mesh spectral method for the phase-field model of two-phase flows. *J. Comput. Phys.*, 228(8):2978–2992, 2009.
[21] Christopher J. Subich. A robust moving mesh method for spectral collocation solutions of time-dependent partial differential equations. *J. Comput. Phys.*, 294:297–311, 2015.
[22] Juan J. Tapia and P. Gilberto López. Adaptive pseudospectral solution of a diffuse interface model. *J. Comput. Appl. Math.*, 224(1):101–117, 2009.
[23] T. Wynn Tee. *An adaptive rational spectral method for differential equations with rapidly varying solutions*. PhD thesis, University of Oxford, 2006.
[24] T. Wynn Tee and Lloyd N. Trefethen. A rational spectral collocation method with adaptively transformed chebyshev grid points. *SIAM J. Sci. Comput.*, 28(5):1798–1811, 2006.
[25] Elliott S. Wise, Ben T. Cox, and Bradley E. Treeby. A monitor function for spectral moving mesh methods applied to nonlinear acoustics. In *Recent Developments in Nonlinear Acoustics: 20th International Symposium on Nonlinear Acoustics including the 2nd International Sonic Boom Forum*, volume 1685, page 070009. AIP Publishing, 2015.
[26] N. J. Zabusky and M. D. Kruskal. Interaction of "solitons" in a collisionless plasma and the recurrence of initial states. *Phys. Rev. Lett.*, 15(6):240–243, 1965.

https://doi.org/10.4208/cicp.OA-2016-0246 Published online by Cambridge University Press