Stable Dynamical Adaptive Mesh Refinement

Tomas Lundquist¹ · Jan Nordström^{2,3} · Arnaud Malan¹

Received: 3 July 2020 / Revised: 16 November 2020 / Accepted: 21 January 2021 /
Published online: 13 February 2021
© The Author(s) 2021

Abstract

We consider accurate and stable interpolation procedures for numerical simulations utilizing time dependent adaptive meshes. The interpolation of numerical solution values between meshes is considered as a transmission problem with respect to the underlying semi-discretized equations, and a theoretical framework using inner product preserving operators is developed, which allows for both explicit and implicit implementations. The theory is supplemented with numerical experiments demonstrating practical benefits of the new stable framework. For this purpose, new interpolation operators have been designed to be used with multi-block finite difference schemes involving non-collocated, moving interfaces.

Keywords Interpolation · Stability · Accuracy · Semi-boundedness · Transmission problem · Inner product preserving

1 Introduction

Time dependent adaptive mesh refinement (AMR) [1,3,27], i.e. where a computational mesh is modified dynamically during the course of a simulation, is a popular and extremely useful technique to reduce the cost of complex modeling tasks involving many types of partial differential equations (PDE). By adding more computational nodes in one region and/or removing nodes in another, local changes in solution characteristics over time can be accounted for, leading to a more efficient use of computer resources. See e.g. [2,11,17,24] for a selection of historical applications of AMR techniques.

✉ Jan Nordström
jan.nordstrom@liu.se

Arnaud Malan
arnaud.malan@uct.ac.za
www.incf.d.uct.ac.za

¹ Industrial CFD Research Group, Department of Mechanical Engineering, University of Cape Town, Cape Town, South Africa

² Department of Mathematics, Computational Mathematics, Linköping University, 581 83 Linköping, Sweden

³ Department of Mathematics and Applied Mathematics, University of Johannesburg, P.O. Box 524, Auckland Park 2006, South Africa

One of the core aspects of AMR is the transfer of numerical solutions values from the previous mesh to the next using interpolation. If AMR is applied after each or every few time steps, e.g. in order to track a moving solution profile, the accuracy and formal stability properties of such interpolations become important considerations. In this paper we set out to formulate a general framework for stable and accurate AMR interpolation. This new framework is explicitly intended to be used together with the energy method, focusing on semi-bounded semi-discretizations and especially those resulting from the use of discrete operators with a formal summation-by-parts (SBP) property [4,9,25].

Inner product preserving (IPP) interpolation has recently emerged as a powerful framework for locally adapting the refinement level of a computational mesh with the use of hanging nodes. Developed for numerical methods ranging from finite difference (FD) [20], mass-lumped discontinuous Galerkin (dG) [7] as well as finite volume and various hybrid methods [14,18], the framework of IPP is ideally suited to be used in conjunction with the energy method in order to prove stability of interpolation based interface coupling conditions between general blocks or elements in space. The successful use of IPP operators in the references cited above rely heavily on the interplay between the IPP concept and an underlying formal SBP framework for numerical differentiation on each of the individual blocks or element.

The kind of abrupt and repeated modification to the mesh associated with time-dependent AMR generates a sequence of so called transmission problems [22], i.e. semi-discretized systems of equations coupled in time using interpolation. Focusing on the example of numerical filters in [19], it was demonstrated that the stability of transmission problems in general is tightly linked to the IPP concept discussed above. In this work, we will therefore consider the use of IPP type operators for time dependent AMR interpolation. We develop a general methodology allowing for both explicit and implicit implementations. In terms of available IPP type operator that can be used for this purpose, we note that in the element based dG case, the same type of L_2 projection operators traditionally used for couplings in space [12,13,21] can be shown to satisfy the IPP framework [7]. Hence these operators can also be used in the AMR context of this paper. In the FD case however, new operators must in general be constructed.

The remainder of this paper is organized as follows. In Sect. 2 we outline a general theory of AMR stability based on IPP operators, mirroring and expanding some of the key results for numerical filters in [19]. In Sect. 3 we relate these theoretical findings to the theory of SBP operators. In particular, we discuss and compare the accuracy limitations inherent to both types of operators, and how these relate to the total accuracy of numerical simulations using AMR. In Sect. 4 we discuss IPP interpolation operator design for FD calculations with moving non-collocated interfaces in a multi-block environment. In Sect. 5 we present numerical calculations using both FD and dG methods, investigating convergence rates as well as demonstrating practical benefits of the formal stability associated with the developed AMR interpolation schemes. Finally, in Sect. 6 we draw conclusions.

2 General Stability Theory

As our starting point we let $\mathcal{D} = \mathcal{D}(X, U)$ denote a general semi-discretized non-linear PDE operator, posed on a given mesh of computational nodes denoted by X . That is, after imposing homogeneous boundary conditions, we consider the semi-discrete system of equations

$$U_t = \mathcal{D}(X, U). \quad (1)$$

We shall henceforth only consider operators $\mathcal{D}$ with a semi-bounded property, i.e. for a general vector Φ defined on the mesh, we have

$$(\Phi, \mathcal{D}(\Phi, X))_{\mathcal{P}(X)} = \Phi^T \mathcal{P}(X) \mathcal{D}(\Phi, X) \leq 0 \quad (2)$$

for some quadrature $\mathcal{P}(X)$ which defines a discrete L_2 inner product and norm on the mesh (i.e. $\mathcal{P}$ is a diagonal matrix with positive quadrature weights inserted along the diagonal).

Note that (2) implies that the solution U to (1) is non-growing in the discrete L_2 norm, and this can typically be seen as an analogue of the same property of the continuous PDE. In other words, semi-boundedness simply means that numerical stability follows from the energy method for a specific quadrature norm. Hence, (1) encompasses a great number of different PDE problems (as well as related numerical methods) of both hyperbolic and parabolic type. See e.g. [9] for a comprehensive introduction to the topics of semi-boundedness and energy stability.

2.1 AMR as a Transmission Problem

Now suppose that we wish to refine and/or coarsen the computational mesh locally after n time steps, and then interpolate the solution to the new mesh before carrying out time step number $n + 1$. Let X^n, X^{n+1} denote the two meshes at $t = t^n$ and $t = t^{n+1}$ respectively. Note that the mesh does not change continuously between the two time steps. In general, even the dimensionality changes abruptly as the mesh is refined and/or coarsened locally.

On the semi-discrete level, we consider a single AMR operation at time $t = t^n$ as a transmission problem [19,22]

$$\begin{aligned} U_t^n &= \mathcal{D}^n(X^n, U^n), \quad t \leq t^n \\ U_t^{n+1} &= \mathcal{D}^{n+1}(X^{n+1}, U^{n+1}), \quad t > t^n \\ U^{n+1}(t^n) &= \mathcal{F}[U^n(t^n), U^{n+1}(t^n)]. \end{aligned} \quad (3)$$

The symbol $\mathcal{F}$ is used here to represent the interpolation procedure of either implicit or explicit form, red where in the latter case $\mathcal{F}$ simplifies into $\mathcal{F}[U^n, U^{n+1}] = \mathcal{F}(U^n)$. The design goal of $\mathcal{F}$, apart from interpolating solution values accurately from one mesh to the other, is to preserve the stability of the underlying schemes due to semi-boundedness (2). To achieve this, we hence require that $\mathcal{F}$ satisfies the following contractivity condition with respect to the same quadratures

$$\Psi = \mathcal{F}(\Phi, \Psi) \Rightarrow \|\Psi\|_{\mathcal{P}^{n+1}} \leq \|\Phi\|_{\mathcal{P}^n}, \quad (4)$$

thus ensuring that the solution to (3) remains non-growing.

2.2 Inner Product Preserving Interpolation

A pair of IPP interpolation operators between the two meshes X^n and X^{n+1} with respect to the quadratures in (2) are defined by

$$(\Psi, \mathcal{I}_{FW}\Phi)_{\mathcal{P}^{n+1}} = (\mathcal{I}_{BW}\Psi, \Phi)_{\mathcal{P}^n}, \quad \text{i.e.} \quad \mathcal{P}^{n+1}\mathcal{I}_{FW} = \mathcal{I}_{BW}^T\mathcal{P}^n, \quad (5)$$

where the subscript FW indicates forward interpolation from the previous to the new mesh, and BW denotes backward interpolation in the opposite direction. Note that in the case of an explicit AMR implementation, only $\mathcal{I}_{FW}$ has to be implemented. Even in this case however, the backward operator $\mathcal{I}_{BW}$ as defined by (5) must be accurate, as we will show below.

We consider accuracy relations of the type

$$\mathcal{I}_{FW}\varphi(X^n) = \varphi(X^{n+1}) \quad (6)$$

$$\mathcal{I}_{BW}\varphi(X^{n+1}) = \varphi(X^n) \quad (7)$$

for all polynomials φ up to (and including) a certain order p , where $\varphi(X^n)$, $\varphi(X^{n+1})$ are vectors with the value of φ in each node. For the reader's convenience we restate below the central accuracy result for IPP operators originally proven in [18].

Proposition 1 *Let $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$ be a pair of IPP interpolation operators, defined in (5). If both (6) and (7) are satisfied for all polynomials φ of order up to p , then*

$$\varphi(X^{n+1})^T \mathcal{P}^{n+1} \varphi(X^{n+1}) = \varphi(X^n)^T \mathcal{P}^n \varphi(X^n) \quad (8)$$

also holds for all polynomials up to order p .

In terms of practical operator construction, this result infers an upper theoretical limit to the formal accuracy of $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$ based on properties of the quadratures $\mathcal{P}^n$ and $\mathcal{P}^{n+1}$ in (2). A more in-depth discussion of the accuracy restrictions following from Proposition 1 in the AMR context will follow later in Sect. 3.

2.3 Explicit Implementation

To start with, we consider the most straightforward, explicit implementation of an interpolation operator $\mathcal{I}_{FW}$ in the AMR procedure, i.e. let

$$\mathcal{F}(\Phi, \Psi) = \mathcal{I}_{FW}\Phi. \quad (9)$$

The following so-called transmission condition, first formulated for problems of the type (3) in [22], can be shown to imply the contractivity condition (4)

$$\mathcal{I}_{FW}^T \mathcal{P}^{n+1} \mathcal{I}_{FW} - \mathcal{P}^n \leq 0. \quad (10)$$

This condition was also considered for stationary non-collocated interfaces of dissipative type in [20].

We are now ready to state our first main new result, providing a link between IPP operator theory and the explicit transmission condition (10).

Proposition 2 *Consider a forward interpolation operator $\mathcal{I}_{FW}$ satisfying (6) for polynomials up to order p , and assume furthermore that the mesh quadratures satisfy (8) for polynomials up to order p . Then, in order for the transmission condition (10) to hold, it is necessary that also the IPP backward operator $\mathcal{I}_{BW}$ (5) satisfies (7) for polynomials up to order p .*

Proof We start by assuming (6) and (8) for some value p , as stated in the premise. Now let φ be any polynomial of order up to p . We then have, defining $\varphi_n = \varphi(X^n)$ and $\varphi_{n+1} = \varphi(X^{n+1})$,

$$\begin{aligned} \varphi_n^T (\mathcal{I}_{FW}^T \mathcal{P}^{n+1} \mathcal{I}_{FW} - \mathcal{P}^n) \varphi_n &= (\mathcal{I}_{FW} \varphi_n)^T \mathcal{P}^{n+1} (\mathcal{I}_{FW} \varphi_n) - \varphi_n^T \mathcal{P}^n \varphi_n \\ &= \varphi_{n+1}^T \mathcal{P}^{n+1} \varphi_{n+1} - \varphi_n^T \mathcal{P}^n \varphi_n = 0, \end{aligned}$$

where we have used both (6) and (8). It now follows from Lemma 14 in [23] that the matrix $\mathcal{I}_{FW}^T \mathcal{P}^{n+1} \mathcal{I}_{FW} - \mathcal{P}^n$ is indefinite unless we also have

$$(\mathcal{I}_{FW}^T \mathcal{P}^{n+1} \mathcal{I}_{FW} - \mathcal{P}^n) \varphi_n = 0$$

for all polynomials φ of order up to p . After again inserting (6), using the definition of $\mathcal{I}_{BW}$ in (5) and multiplying through with the inverse of $\mathcal{P}^n$, we find

$$\mathcal{I}_{BW}\varphi_{n+1} = \varphi_n. \quad (11)$$

Hence, $\mathcal{I}_{BW}$ satisfies (7) up to order p , which concludes the proof.

We note that Proposition 2 above can be seen as a straightforward extension of Proposition 1 in [19], which relates to explicit filtering procedures. In the context of filters however, we have $\mathcal{P}^{n+1} = \mathcal{P}^n$, which makes the assumption that (8) holds superfluous. Hence, Proposition 2 is a more general result which applies to all types of IPP operators.

2.4 Implicit Implementation

From a pure operator construction perspective, it is more convenient to only consider the IPP property (5), which is a linear constraint on the coefficients in $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$, rather than also the additional non-convex eigenvalue constraint in (10). As an alternative to the explicit AMR implementation considered above, we add an implicit correction term to (9)

$$\mathcal{F}(\Phi, \Psi) = \mathcal{I}_{FW}\Phi - \mathcal{I}_{FW}(\mathcal{I}_{BW}\Psi - \Phi). \quad (12)$$

We can now directly prove that the IPP property (5) is sufficient for contractivity (4), i.e. without the need for the additional constraint (10).

Proposition 3 *Consider the implicit AMR implementation $\Psi = \mathcal{F}(\Phi, \Psi)$ in (12), and let $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$ be IPP as in (5). Then*

$$\|\Psi\|_{\mathcal{P}^{n+1}}^2 = \|\Phi\|_{\mathcal{P}^n}^2 - \|\Phi - \mathcal{I}_{BW}\Psi\|_{\mathcal{P}^n}^2$$

holds, i.e. the operation is contractive (4).

Proof First note that $\Psi = \mathcal{F}(\Phi, \Psi)$ in (12) gives, $(\mathcal{I} + \mathcal{I}_{FW}\mathcal{I}_{BW})\Psi = 2\mathcal{I}_{FW}\Phi$. Multiplying with $\Psi^T \mathcal{P}^{n+1}$ from the left, we get

$$\|\Psi\|_{\mathcal{P}^{n+1}}^2 + (\Psi, \mathcal{I}_{FW}\mathcal{I}_{BW}\Psi)_{\mathcal{P}^{n+1}} = 2(\Psi, \mathcal{I}_{FW}\Phi)_{\mathcal{P}^{n+1}}.$$

From (5), this becomes

$$\|\Psi\|_{\mathcal{P}^{n+1}}^2 + \|\mathcal{I}_{BW}\Psi\|_{\mathcal{P}^n}^2 = 2(\mathcal{I}_{BW}\Psi, \Phi)_{\mathcal{P}^n}.$$

After adding and subtracting $\|\Phi\|_{\mathcal{P}^n}^2$ on the right hand side, we finally get

$$\begin{aligned} \|\Psi\|_{\mathcal{P}^{n+1}}^2 &= \|\Phi\|_{\mathcal{P}^n}^2 - \|\Phi\|_{\mathcal{P}^n}^2 - \|\mathcal{I}_{BW}\Psi\|_{\mathcal{P}^n}^2 + 2(\mathcal{I}_{BW}\Psi, \Phi)_{\mathcal{P}^n} \\ &= \|\Phi\|_{\mathcal{P}^n}^2 - \|\Phi - \mathcal{I}_{BW}\Psi\|_{\mathcal{P}^n}^2, \end{aligned}$$

which concludes the proof.

The analogous result for filtering procedures with $\mathcal{P}^{n+1} = \mathcal{P}^n$ was given in [19].

3 Accuracy Considerations

In terms of accuracy, we have seen in Proposition 1 that IPP interpolation operators and their underlying quadratures rules are interrelated through the conditions (6), (7) and (8). Recall

that these quadrature rules in turn relate to the underlying semi-discrete schemes through the property of semi-boundedness (2). In this section we will discuss accuracy in more depth, and in particular how Proposition 1 relates to the underlying schemes.

3.1 Differentiation and Integration

We consider a discrete first derivative operator $\mathcal{D}_{x_i}$, for the coordinate direction x_i on a mesh denoted by X . The accuracy relations on a Cartesian mesh are of the type

$$\mathcal{D}_{x_i} \varphi(X) = \frac{\partial}{\partial x_i} \varphi(X), \quad (13)$$

for polynomials φ of order up to (and including) s .

Semi-boundedness (2) can e.g. be achieved by formally postulating a SBP operator structure of all partial derivative approximations. By the use of compatibility conditions, the accuracy of such operators can be shown to be interrelated with their respective quadrature rules in a similar way that IPP operators are due to Proposition 1 [10,15,16]. In particular, a simple 1D analysis reveals that if differentiation is carried out to accuracy order s (13), then the associated quadrature must be exact for mesh polynomials up to order $2s - 1$, see [16].

In order to relate this and similar results to the condition (8) in Proposition 1, we note first that the integrand φ^2 in (8) is a polynomial of even order. Hence, if a quadrature $\mathcal{P}$ is exact for all polynomials up to order $2s - 1$, we then have

$$\varphi(X)^T \mathcal{P} \varphi(X) = \int_{\Omega} \varphi^2 d\Omega \quad (14)$$

for all polynomials φ up to order $s - 1$.

3.2 IPP Interpolation

If the quadratures in (8) are known to satisfy the integration conditions (14) already from properties of the semi-discretization, then the significance of Proposition 1 can be better understood by the following corollary.

Corollary 1 *Consider the case where (14) is satisfied up to order $s - 1$, but where neither (14) nor (8) holds to order s or higher. Then an IPP operator pair $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$ can not simultaneously satisfy (6) and (7) for any p larger than $s - 1$. In other words, at most one of them can be satisfied for $p = s$.*

Proof The premise is that (8) is true for all polynomials φ up to order $s - 1$ (since this follows from (14)), but not for all φ of order s . Moreover, Proposition 1 states that (6) and (7) together implies (8). Thus, if both (6) and (7) would hold for $p = s$, then this would lead to a contradiction since we have already established that (8) does not hold up to s .

Remark 1 Recall that the upper limit of $s - 1$ in (14) is related to the use of SBP operators of order s (13). In light of this, the upper accuracy limit of $p = s - 1$ given by Corollary 1 should be seen as a serious limitation. By Taylor's theorem, interpolating a smooth function on the mesh results in errors of order s . As a comparison, in the absence of AMR interpolation, the expected convergence rate of a finite difference SBP scheme is $s + 1$ [25,26]. This observation suggests that a numerical solution loses one order of accuracy even after a single

AMR operation, since the interpolation adds on a new error of order s which is independent of the order of the scheme.

If AMR is applied frequently (after each or every few time steps), the effect on solution accuracy is harder to predict, especially if the region in space where AMR interpolation is applied is small and/or moving. However, some level of error accumulation over time must in general be expected, possibly leading to a further reduction in convergence order.

In the case of finite difference operators, a solution to this suboptimal accuracy constraint resulting from Proposition 1 was proposed in [6]. By increasing the number of boundary nodes and modifying the associated quadrature weights for each grid size, the order of polynomials in (14) could be increased to s , thus removing the constraint of $s - 1$ in Corollary 1. This is a rather specialized approach however, and for standard SBP finite difference operators with fixed boundary closure coefficients, no such fix is possible [16].

3.3 Explicit AMR Interpolation

In the explicit implementation (9) of AMR, we only apply one of the two interpolation operators in an IPP pair, i.e. the forward operator $\mathcal{I}_{FW}$. In this case, Corollary 1 does not rule out the possibility of maintaining an accuracy of $p = s$, as long as $\mathcal{I}_{BW}$ (which is not used) has the lower value $p = s - 1$. However, even this is not always possible, as the following result shows.

Corollary 2 *Consider the premises of Corollary 1. If we assume that $\mathcal{I}_{BW}$ satisfies (7) for $p = s - 1$, and $\mathcal{I}_{FW}$ satisfies (6) for $p = s$, then for (10) to hold we need to additionally have*

$$\varphi_{n+1}^T \mathcal{P}^{n+1} \varphi_{n+1} \leq \varphi_n^T \mathcal{P}^n \varphi_n \quad (15)$$

for all polynomials φ of order s .

Proof First note that (15) is automatically satisfied with equality up to order $s - 1$ due to (14). For the case of order s , consider the quadratic form of the left hand side in (10) with respect to φ_n

$$\varphi_n^T \left(\mathcal{I}_{FW}^T \mathcal{P}^{n+1} \mathcal{I}_{FW} - \mathcal{P}^n \right) \varphi_n \leq 0.$$

Now, since we have assumed that (6) holds up to order s , this condition exactly reduces to (15). $\square$

Remark 2 To understand the nature of the constraint (15) better, it is instructive to consider the special case of $\mathcal{P}^n$ and $\mathcal{P}^{n+1}$ representing a trapezoidal rule on two different meshes X^n and X^{n+1} . If the integration is exact for all polynomials up to order $2s - 1$ due to compatibility conditions, i.e. (14), we may without loss of generality only consider a single monomial of order s in (15), i.e. the highest order term of the polynomial. The integrand φ^2 is then a convex positive function, implying that the piecewise linear trapezoidal segments will systematically overestimate the real integral value. The coarser the mesh, the more pronounced this overestimation becomes, which means that (15) is satisfied if the new mesh X^{n+1} is a refinement of X^n , but not the other way around.

This property of the trapezoidal rule carry over to higher order quadrature rules as well. In other words, as long as the mesh is refined and not coarsened, condition (2) holds, making it possible to construct a forward interpolation operator $\mathcal{I}_{FW}$ of order $p = s$.

Fig. 1 Schematic of AMR interpolation in 1D

3.4 Comments Regarding Explicit Versus Implicit AMR

As already mentioned, one advantage with the implicit implementation (12) of AMR interpolation is the simplicity gained from only requiring a pair of IPP operators (5), whereas the additional transmission condition (10) required in the explicit case is more complicated to achieve. However, we have seen in Corollary 2 and Remark 2 that the explicit implementation is potentially more accurate (raising the order from $s - 1$ to s) in the case of refinement, i.e. adding nodes locally. It is also more straightforward to implement. We will make practical comparisons of solution accuracy using the two approaches in the numerical section of this paper.

4 Finite Difference Application with a Moving Interface

In this section we discuss one application of the AMR theory outlined above. As a model, we consider a two-block FD grid with an advancing front between two areas of different grid resolution. The situation is illustrated in Fig. 1 for the 1D case and in Fig. 2 for the 2D case. Note that these two figures will also be explained in more detail in Sects. 4.1 and 4.2, respectively. In both cases, the interface is moved in a step-wise manner between times t^n and t^{n+1} by adding two nodes to the x -direction of the fine grid, and removing one from the coarse grid. In 2D or 3D, if the resulting interfaces are non-collocated in space, they must be handled weakly by the underlying scheme, with penalty terms included in $\mathcal{D}^n$ and $\mathcal{D}^{n+1}$.

4.1 Operators in One Dimension

Figure 1 illustrates the various components of an AMR operation in 1D. The fine region in both grids are associated with quadratures P_f^n and P_f^{n+1} , and the coarse regions with P_c^n and P_c^{n+1} . For interpolation between t^n and t^{n+1} , we only need to consider a smaller set of nodes, and do nothing outside of the interface region. In other words, we are only interested in a small subset of the quadrature weights, given by

$$P_f^n = \begin{pmatrix} \times & & & \\ & \ddots & & \\ & & \times & \text{---} \\ & & & \hat{P}_f^n \end{pmatrix}, \quad P_c^n = \begin{pmatrix} & & & \\ & \hat{P}_c^n & & \\ & & \times & \\ & & & \ddots \\ & & & & \times \end{pmatrix},$$

and we also divide P_f^{n+1} , P_c^{n+1} in the same way. Correspondingly, we consider interpolation operators of the form

$$I_{FW} = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \hat{I}_{FW} \\ & & & & 1 & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix}, \quad I_{BW} = \begin{pmatrix} 1 & & & \\ & \ddots & & \\ & & 1 & \\ & & & \hat{I}_{BW} \\ & & & & 1 & \\ & & & & & \ddots \\ & & & & & & 1 \end{pmatrix}.$$

The IPP property (5) thus reduces to

$$\hat{P}^{n+1} \hat{I}_{FW} = \hat{I}_{BW}^T \hat{P}^n,$$

where

$$\hat{P}^n = \begin{pmatrix} \hat{P}_f^n \\ \hat{P}_c^n \end{pmatrix}, \quad \hat{P}^{n+1} = \begin{pmatrix} \hat{P}_f^{n+1} \\ \hat{P}_c^{n+1} \end{pmatrix}.$$

Next, we will show how AMR operators designed in this way for one space dimension can be applied to higher dimensional grids.

4.2 Operators in Two Dimensions

To illustrate the extension of the 1D AMR procedure above into two space dimensions, see Fig. 2 where we simply extrude the 1D example grid from Fig. 1 in the vertical direction.

In doing this, we introduce new sets of quadrature weights P_c and P_f along the whole length of the interface. Corresponding to these, we consider a pair of IPP interpolation operators

$$P_f I_{c2f} = I_{f2c}^T P_c.$$

In the horizontal direction, we will make use of the 1D operators $\hat{I}_{FW}$ and $\hat{I}_{BW}$ already introduced. As before, we let the complete AMR interpolation operators (here denoted by $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$) be given by identity outside of the indicated interface region.

Consider a block-wise x -major lexicographical ordering of all nodes inside the interface region. By the use of Kronecker products, we may thus collect the full set of 2D nodal

Fig. 2 Schematic of AMR interpolation in 2D. The top part describes the big picture view of refinement with a moving interface, while the bottom part describes the steps involved in more detail

quadrature weights into the matrices

$$\hat{P}^n = \begin{pmatrix} \hat{P}_f^n \otimes P_f & \\ & \hat{P}_c^n \otimes P_c \end{pmatrix}, \quad \hat{P}^{n+1} = \begin{pmatrix} \hat{P}_f^{n+1} \otimes P_f & \\ & \hat{P}_c^{n+1} \otimes P_c \end{pmatrix}.$$

The complete IPP procedure for a non-collocated interface in 2D can be divided into a sequence of 1D operations using Kronecker products. For the forward in time operation $\mathcal{I}_{FW}$, those steps are described below, and also illustrated in Fig. 2.

1. First, the coarse grid is refined in the vertical direction in order to make the two grids collocated. With respect to the whole interface region, this amounts to applying the operator

$$\begin{pmatrix} \hat{I}_f^n \otimes I_f & \\ & \hat{I}_c^n \otimes I_{c2f} \end{pmatrix},$$

where $\hat{I}_c^n$, $\hat{I}_f^n$ and I_f denote identity matrices of the same dimensions as $\hat{P}_c^n$, $\hat{P}_f^n$ and P_f , respectively. In the lower part of Fig. 2, this step corresponds to moving from the first to the second grid from left to right.

2. At this point, all grid points in the interface region are aligned along horizontal lines. However, before applying the 1D AMR interpolation operators with the use of Kronecker products, we formally have to reorder the nodes between the original block-wise into a complete lexicographical node ordering. For this purpose we let $\tilde{E}^n$ denote the permutation operator which reorders all nodes according to ascending y -values. This step is only of a virtual book-keeping nature and is not shown explicitly in Fig. 2.
3. We can now extend the AMR interpolation operator in 1D into 2D using a Kronecker product

$$(\hat{I}_{FW} \otimes I_f).$$

This step corresponds to moving from the second to the third grid in the lower part of Fig. 2.

4. Before proceeding, we have to reorder the nodes back into block-wise lexicographical ordering. We let this permutation operation be denoted with $(\tilde{E}^{n+1})^T$.
5. Finally, the right grid is coarsened along the vertical direction to make the interface non-collocated again

$$\begin{pmatrix} \hat{I}_f^{n+1} \otimes I_f^{n+1} & \\ & \hat{I}_c^{n+1} \otimes I_{f2c} \end{pmatrix}.$$

This step corresponds to moving from the third to the fourth grid in Fig. 2.

It can easily be shown that the successive application of several IPP operators is always an IPP operation in itself, see Lemma 1 in [18]. Thus, since each step above involves the use of an IPP operator, it follows that the whole sequence is IPP with respect to the reverse operation. In the same way, the condition (10) of contractivity is also preserved if each one of the individual steps is contractive.

The complete five step procedure in both directions can be written out compactly as

$$\begin{aligned} \hat{\mathcal{I}}_{FW} &= \begin{pmatrix} \hat{I}_f^{n+1} \otimes I_f^{n+1} & \\ & \hat{I}_c^{n+1} \otimes I_{f2c} \end{pmatrix} (\tilde{E}^{n+1})^T (\hat{I}_{FW} \otimes I_f) \tilde{E}^n \begin{pmatrix} \hat{I}_f^n \otimes I_f^n & \\ & \hat{I}_c^n \otimes I_{c2f} \end{pmatrix} \\ \hat{\mathcal{I}}_{BW} &= \begin{pmatrix} \hat{I}_f^n \otimes I_f^n & \\ & \hat{I}_c^n \otimes I_{f2c} \end{pmatrix} (\tilde{E}^n)^T (\hat{I}_{BW} \otimes I_f) \tilde{E}^{n+1} \begin{pmatrix} \hat{I}_f^{n+1} \otimes I_f^{n+1} & \\ & \hat{I}_c^{n+1} \otimes I_{c2f} \end{pmatrix}, \end{aligned}$$

which describes the two operators $\mathcal{I}_{FW}$ and $\mathcal{I}_{BW}$ inside the interface region.

Fig. 3 Gaussian profile with center in the left part of the fine grid

5 Numerical Results

We test the stability and convergence properties of the new AMR framework, by conducting numerical experiments using a set of both linear and non-linear hyperbolic model problems. The IPP operator coefficients that we have used in conjunction with the novel FD methodology of Sect. 4 are listed in Appendix A. In line with Remark 2, the order of accuracy is $p = s$ in the forward direction of the schematic in Fig. 1, and $p = s - 1$ in the backward direction.

5.1 One Dimension

For our one-dimensional test cases we consider the model problem

$$\begin{aligned} u_t + u_x &= 0, \quad -2 \leq x \leq 2, \quad t \geq 0 \\ u(x, 0) &= f(x). \end{aligned} \quad (16)$$

A time dependent finite difference block discretization is applied, consisting of a moving finely resolved interior block surrounded by two coarser blocks, with a refinement ratio of $1/2$ between the coarse and the fine grids. The interior block moves with a constant speed of 1 from a starting point given by $-1 \leq x \leq 0$ at $t = 0$.

Again relating to Remark 2 and the schematic in Fig. 1, we note that the order of forward AMR interpolation $\mathcal{I}_{FW}$ (and thus of the explicit AMR implementation) is $p = s - 1$ at the left block interface, and $p = s$ at the right interface, while for implicit AMR we have $p = s - 1$ at both interfaces. In order to study the effect of AMR on solution accuracy at these two types of interfaces separately, we will consider two different Gaussian profiles as exact solutions to (16), both moving with the same unit speed as the interior block. The moving block structure as well as the two exact solution profiles are illustrated in Figs. 3 and 4 respectively, where the fine interior block is indicated with dotted red lines in the solution curves.

Fig. 4 Gaussian profile with center in the right part of the fine grid

We stress that the only purpose with the numerical tests in this section are to study the asymptotic convergence rates resulting from AMR interpolation around the two types of block interfaces. Hence, we have deliberately placed the two solution profiles such that gradients are relatively large at these two interfaces, respectively, i.e. so that the contribution from AMR interpolation dominates the error. In real world applications one would of course aim to do the exact opposite, i.e. to place active AMR interfaces as far away from steep solution gradients as possible.

5.1.1 Left Interface Treatment

In (16) we first consider the exact solution

$$u(x, t) = \exp\left(\frac{-(x + 5/7 - t)^2}{2(1/7)^2}\right),$$

which we have plotted at $t = 0$ and $t = 1$ in Fig. 3, together with the moving multi-block grid structure. Note that in the initial configuration, this Gaussian is centered at $x = -5/7$, i.e. some distance to the left of the fine block center at $x = -1/2$. By placing the Gaussian in this way, solution gradients are much steeper at the left interface, and we thus highlight the contribution to the numerical error from the left interface (i.e. where $p = s - 1$ for both the implicit and explicit AMR implementations), while the contribution from the right interface treatment can be neglected.

In Fig. 5 we show the error profile at $t = 1$ for two different mesh resolutions, using a classical SBP FD scheme with $s = 2$ (i.e. fourth order interior accuracy), together with an implicit AMR implementation. The grid parameter N denotes the number of nodes in the interior fine grid, and we have used the classical fourth order explicit Runge-Kutta scheme for time marching with $\Delta t = (1/8)(1/N)$. Consequently, AMR has been applied after each 8 time steps in order for the block interface to move with the correct speed. The same time stepping strategy will be used for all subsequent numerical tests as well.

Fig. 5 Error profiles using an implicit AMR implementation

Table 1 L_∞ norm convergence for test case 1 using explicit AMR

N	$s = 1$		$s = 2$		$s = 3$	
	$\log_{10}(\ e\ _\infty)$	p	$\log_{10}(\ e\ _\infty)$	p	$\log_{10}(\ e\ _\infty)$	p
8	-0.7079	—	—	—	—	—
16	-1.1266	1.39	-1.8895	—	-2.2601	—
32	-1.6007	1.58	-2.4240	1.78	-3.3732	3.70
64	-1.7691	0.56	-2.9714	1.82	-4.5407	3.88
128	-1.9247	0.52	-3.3877	1.38	-5.3191	2.59
256	-2.0846	0.53	-3.8184	1.43	-6.2116	2.96
512	-2.2537	0.56	-4.2470	1.42	-7.0303	2.72
1024	-2.4314	0.59	-4.6727	1.41	-7.8005	2.56
2048	-2.6156	0.61	-5.1210	1.49	-8.5434	2.47
4096	-2.8047	0.63	-5.5852	1.54	-9.2777	2.44

Already from the two plots in Fig. 5, a slower converge in the region immediately surrounding the left interface can be surmised. To study this effect in more detail, in Tables 1 and 2 we list the obtained convergence rates (using a standard definition of $p = \log_2(\|e(N)\|/\|e(N/2)\|)$ with respect to the maximum (L_∞) norm of the error for the explicit and implicit AMR implementations, respectively. Asymptotically, the convergence rates appear to stabilize at an order of $s - 1/2$ in both the explicit and the implicit case, to be compared with the general rate of $s + 1$ for the basic FD schemes, and to order s of the interpolation operators. Note that the interpolation scheme is applied $\mathcal{O}(N)$ times to track the moving solution profile. Thus, since the observed convergence rate is lower by one half order compared to the interpolation scheme itself, we conclude that this must be due to AMR interpolation errors accumulating over time, as previously predicted in Remark 1. The L_2 convergence rates in the explicit and implicit cases are shown in Tables 3 and 4, pointing to a general rate of s , i.e. a loss of one order compared to the case with no AMR interpolation.

Table 2 L_∞ norm convergence for test case 1 using implicit AMR

N	$s = 1$		$s = 2$		$s = 3$	
	$\log_{10}(\ e\ _\infty)$	p	$\log_{10}(\ e\ _\infty)$	p	$\log_{10}(\ e\ _\infty)$	p
8	-0.6267	—	—	—	—	—
16	-1.1087	1.60	-2.0419	—	-2.1739	—
32	-1.7098	2.00	-2.5973	1.84	-3.3068	3.76
64	-2.0075	0.99	-3.1928	1.98	-4.7392	4.76
128	-2.1694	0.54	-3.6464	1.51	-5.3516	2.03
256	-2.3604	0.63	-4.0788	1.44	-6.2671	3.04
512	-2.5287	0.56	-4.4944	1.38	-7.2441	3.25
1024	-2.6950	0.55	-4.9200	1.41	-8.1474	3.00
2048	-2.8698	0.58	-5.3679	1.49	-8.8913	2.47
4096	-3.0525	0.61	-5.8199	1.50	-9.5773	2.28

Table 3 L_2 norm convergence for test case 1 using explicit AMR

N	$s = 1$		$s = 2$		$s = 3$	
	$\log_{10}(\ e\ _2)$	p	$\log_{10}(\ e\ _2)$	p	$\log_{10}(\ e\ _2)$	p
8	-0.9364	—	—	—	—	—
16	-1.4298	1.64	-2.3859	—	-2.5112	—
32	-1.8914	1.53	-2.9938	2.02	-3.7695	4.18
64	-2.2462	1.18	-3.5774	1.94	-5.1472	4.58
128	-2.5438	0.99	-4.1242	1.82	-5.9822	2.77
256	-2.8317	0.96	-4.6656	1.80	-7.0343	3.50
512	-3.1217	0.96	-5.2156	1.83	-8.1000	3.54
1024	-3.4149	0.97	-5.7764	1.86	-9.0164	3.04
2048	-3.7107	0.98	-6.3469	1.90	-9.8307	2.70
4096	-4.0083	0.99	-6.9255	1.92	-10.6372	2.68

Table 4 L_2 norm convergence for test case 1 using implicit AMR

N	$s = 1$		$s = 2$		$s = 3$	
	$\log_{10}(\ e\ _2)$	p	$\log_{10}(\ e\ _2)$	p	$\log_{10}(\ e\ _2)$	p
8	-0.8984	—	—	—	—	—
16	-1.4248	1.75	-2.5224	—	-2.4272	—
32	-1.9550	1.76	-3.2191	2.31	-3.7086	4.26
64	-2.4025	1.49	-3.8334	2.04	-5.3217	5.36
128	-2.7558	1.17	-4.3845	1.83	-6.0203	2.32
256	-3.0635	1.02	-4.9242	1.79	-7.0660	3.47
512	-3.3588	0.98	-5.4711	1.82	-8.1870	3.72
1024	-3.6527	0.98	-6.0279	1.85	-9.2440	3.51
2048	-3.9480	0.98	-6.5937	1.88	-10.1347	2.96
4096	-4.2447	0.99	-7.1666	1.90	-10.9580	2.73

Fig. 6 Error profiles using an explicit AMR implementation

5.1.2 Right Interface Treatment

For the second test case, we use as exact solution in (16) the following Gaussian profile located to the right of the fine block center

$$u(x, t) = \exp\left(\frac{-(x + 2/7 - t)^2}{2(1/7)^2}\right).$$

See again Fig. 4 for an illustration. In this way, we thus highlight the contribution to the numerical error from the right interface (i.e. where $p = s - 1$ the implicit and $p = s$ for the explicit AMR implementation), while the contribution from the left interface treatment can be neglected.

In Fig. 6 we show the error profile for two different mesh resolutions, again using the classical SBP-FD method with $s = 2$ (fourth order interior accuracy), this time with an explicit AMR implementation. Moreover, we quantify the convergence rates obtained in Tables 5 and 6 in the L_∞ and L_2 norm, respectively. The asymptotic rates are not as easy to interpret in this case, but appear in general to lie somewhere between s and $s + 1/2$ for the L_∞ norm, and between $s + 1/2$ and $s + 1$ for the L_2 norm. Compared to the previous case, the accuracy is thus approximately between one half and one order higher, which reflects the higher order of AMR interpolation as noted earlier.

5.2 Two Dimensions

We generalize the 1D test case discussed above and consider the moving block mesh drawn in Fig. 7 at $t = 0$ and $t = 1$. The model problem is given by

$$\begin{aligned} u_t + u_x + u_y &= 0, \quad -2 \leq x, y \leq 2, \quad t \geq 0 \\ u(x, 0) &= f(x, y), \end{aligned}$$

again amended with a Gaussian exact solution profile (described below for the two cases).

In terms of AMR, there are essentially two new features which come into play in the 2D case. Firstly, we need a pair of interpolation operators I_{f2c} and I_{c2f} acting in the tangent direction of each mesh refinement interface. Secondly, there are no less than 8 moving inter-

Table 5 L_∞ norm convergence for test case 2 using explicit AMR

N	$s = 1$		$s = 2$		$s = 3$	
	$\log_{10}(\ e\ _\infty)$	p	$\log_{10}(\ e\ _\infty)$	p	$\log_{10}(\ e\ _\infty)$	p
8	-0.5644	—	—	—	—	—
16	-1.1096	1.81	-2.3330	—	-1.9969	—
32	-1.6926	1.94	-3.1929	2.86	-3.1661	3.88
64	-2.1521	1.53	-3.9143	2.40	-4.3855	4.05
128	-2.4851	1.11	-4.6367	2.40	-5.5514	3.87
256	-2.8402	1.18	-5.3336	2.31	-6.5031	3.16
512	-3.2145	1.24	-6.0179	2.27	-7.4869	3.27
1024	-3.5948	1.26	-6.7062	2.29	-8.5099	3.40
2048	-3.9826	1.29	-7.3857	2.26	-9.5516	3.46
4096	-4.3773	1.31	-8.0672	2.26	-10.6320	3.59

Table 6 L_2 norm convergence for test case 1 using explicit AMR

N	$s = 1$		$s = 2$		$s = 3$	
	$\log_{10}(\ e\ _2)$	p	$\log_{10}(\ e\ _2)$	p	$\log_{10}(\ e\ _2)$	p
8	-0.8305	—	—	—	—	—
16	-1.3887	1.85	-2.6349	—	-2.4522	—
32	-1.9729	1.94	-3.6212	3.28	-3.7568	4.33
64	-2.5254	1.84	-4.5347	3.03	-5.1181	4.52
128	-3.0591	1.77	-5.4397	3.01	-6.2564	3.78
256	-3.5834	1.74	-6.3005	2.86	-7.3504	3.63
512	-4.1011	1.72	-7.1211	2.73	-8.4792	3.75
1024	-4.6138	1.70	-7.9268	2.68	-9.6446	3.87
2048	-5.1229	1.69	-8.7327	2.68	-10.8412	3.98
4096	-5.6294	1.68	-9.5433	2.69	-12.0649	4.06

faces with the same mesh refinement level on both sides. Both of these features require new interpolation operators. We will only study qualitative aspects of the 2D AMR interpolation, and we restrict our attention to the classical SBP FD case with $s = 2$ (i.e. using fourth order interior accuracy). A novel pair of IPP 1D interpolation operators to handle the 8 collocated mesh interfaces is given in Appendix 1, where the accuracy is given by $p = 2$ in both directions.

5.2.1 Left/Bottom Interface Treatment

We generalize the setup from Sect. 5.1.1 and thus consider a Gaussian profile positioned close to the lower left corner of the fine interior grid

$$u(x, y, t) = \exp\left(\frac{-(x + 5/7 - t)^2 - (y + 5/7 - t)^2}{2(1/7)^2}\right).$$

Fig. 7 Block mesh

Fig. 8 Absolute error levels around the bottom left corner of the fine grid. Operators are used which are IPP but not strictly contractive. Note the different scalings of the color bars

We now compare the result of using a mildly unstable AMR procedure compared to a stable one. For this purpose, we will first use the same I_{f2c} and I_{c2f} operators that were presented in [18]. These are IPP, but do not satisfy the explicit transmission condition (10). Hence, an explicit AMR procedure using these operators should be unstable. The results for $N = 64$ are shown in Fig. 8, where the explicit case clearly shows elevated error levels compared to the stable implicit implementation. These higher levels are located at the bottom left corner of the fine grid. Since the same operators are used in both cases, the qualitatively different results observed can probably be ascribed to a developing instability in the explicit case.

In order to compare these results to a setup where also the explicit implementation is stable, we have slightly modified the operators I_{c2f} and I_{f2c} to make them both IPP and contractive, see 1. The new results are shown in Fig. 9, and this time there is no qualitative difference between the two implementations.

This example illustrates the vital importance of making sure that the AMR implementation procedure is strictly stable. Despite the inherent accuracy limitations involved with such schemes, we can not do better by ignoring the issue of stability.

Fig. 9 Absolute error levels around the bottom left corner of the fine grid. Operators are used which are both IPP and strictly contractive

Fig. 10 Result of refinement for an explicit implementation of a coarse to fine AMR case in 2D. Asymptotically, the errors are dominated by the lower accuracy of I_{f2c} and I_{c2f} at the corner point

5.2.2 Right/Top Interface Treatment

We also consider the case where the Gaussian profile is close to the upper right corner of the fine grid, i.e.

$$u(x, y, t) = \exp\left(\frac{-(x + 2/7 - t)^2 - (y + 2/7 - t)^2}{2(1/7)^2}\right).$$

Just as in the 1D case, we benefit from a more accurate explicit implementation by using only the forward AMR operator $\mathcal{I}_{FW}$. However, the operators I_{c2f} and I_{f2c} still exhibit the lower accuracy level $p = 1$ at corner points, and hence we expect the errors at the upper right corner to dominate on a fine enough grid. We confirm that this is the case by showing error plots for two different grid parameters in Fig. 10.

Table 7 L_∞ convergence at $t = 7/8$ as a function of the number of starting elements

N	$s = 2$	p
	$\log_{10}(\ e\ _\infty)$	
8	-0.8193	Inf
16	-1.6421	2.73
32	-2.4318	2.62
64	-3.1147	2.27
128	-3.8628	2.49
256	-4.6559	2.63
512	-5.4944	2.79
1024	-6.3358	2.80
2048	-7.2391	3.00
4096	-8.1336	2.97

5.3 A Non-linear Example

As a final demonstration, we study the inviscid non-linear Burger's equation with an initial sinusoidal solution,

$$\begin{aligned} u_t + uu_x &= 0, \quad -\pi \leq x \leq \pi, \quad 0 < t < 1, \\ u(x, 0) &= \sin(x). \end{aligned}$$

The initial function has a unit slope at $x = 0$, meaning that a discontinuity will develop at t approaches 1. Our goal with this experiment is to demonstrate how AMR can be used to smoothly resolve the solution up until the very moment of shock formation, using a high order, non-dissipative scheme.

To solve the non-linear problem numerically with no dissipation, we employ a skew-symmetric splitting with SBP operators, see [5] for details of this approach. As discrete operators we use Galerkin element SBP operators with Gauss-Lobatto collocation [8]. For example, using third order polynomials (corresponding to $s = 2$ in (13) and (14)), the elemental operators we use are

$$\begin{aligned} \hat{\mathcal{P}} &= \Delta x \begin{pmatrix} 1/6 & & \\ & 2/3 & \\ & & 1/6 \end{pmatrix}, \\ \hat{\mathcal{D}}_x &= \hat{\mathcal{P}}^{-1} \int_0^1 \mathcal{L} \mathcal{L}_\xi^T d\xi = \hat{\mathcal{P}}^{-1} \begin{pmatrix} -1/2 & 2/3 & -1/6 \\ -2/3 & 0 & 2/3 \\ 1/6 & -2/3 & 1/2 \end{pmatrix}, \end{aligned}$$

where $\mathcal{L}$ is the vector containing the Lagrange basis

$$\mathcal{L} = (2(\xi - 1/2)(\xi - 1) \ 4\xi(\xi - 1) \ 2\xi(\xi - 1/2))^T.$$

For inter-element couplings, we use the non-dissipative version of the inviscid interface treatment in [5]. In this way, we ensure that the scheme is strictly energy conserving.

For time integration we use the classical fourth order explicit Runge-Kutta method, with a time step size given by

$$\Delta t = (2/\pi) \Delta x.$$

Fig. 11 Top: initial sinus solution on a uniform grid with $N = 32$ elements (inter-element boundaries are not indicated in the figure). Middle: numerical solution after two full cycles of AMR refinement. Bottom: solution after 10 cycles of refinement

The AMR refinement process will involve splittings of individual elements into two new ones. Using L_2 projections [12,13], the AMR operator corresponding to locally refining a single element is as follows

$$\hat{\mathcal{I}}_{FW} = \begin{pmatrix} \hat{\mathcal{P}}^{-1} \int_0^1 \mathcal{L}(\xi) \mathcal{L}(\xi/2)^T d\xi \\ \hat{\mathcal{P}}^{-1} \int_0^1 \mathcal{L}(\xi) \mathcal{L}(1/2 + \xi/2)^T d\xi \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3/8 & 3/4 & -1/8 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ -1/8 & 3/4 & 3/8 \\ 0 & 0 & 1 \end{pmatrix},$$

where the integration sign above is carried out numerically using the quadrature weights in $\hat{\mathcal{P}}$. Note that the accuracy is given by $s = 2$ in (6), which is in line with the discussion in Remark 2. We can also verify explicitly that this interpolation operator is contractive in the sense of (10), which enables an explicit implementation. We thus define

$$\hat{\mathcal{P}}^n = \hat{\mathcal{P}}, \quad \hat{\mathcal{P}}^{n+1} = \frac{1}{2} \begin{pmatrix} \hat{\mathcal{P}} \\ \hat{\mathcal{P}} \end{pmatrix} = \frac{\Delta x}{2} \begin{pmatrix} 1/6 & & & & \\ & 2/3 & & & \\ & & 1/6 & & \\ & & & 1/6 & \\ & & & & 2/3 & \\ & & & & & 1/6 \end{pmatrix},$$

leading to

$$\hat{\mathcal{I}}_{FW}^T \hat{\mathcal{P}}^{n+1} \hat{\mathcal{I}}_{FW} - \hat{\mathcal{P}}^n = -\frac{1}{32} \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix} = -\frac{1}{32} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} (1 \ -2 \ 1) \leq 0.$$

As the solution profile starts to steepen around $x = 0$, we refine the mesh by adding more elements locally in a controlled way. Here, we are not mainly interested in investigating the best refinement strategy, but rather to demonstrate the robust implementation of a given such strategy. Since the solution behavior is predictable in a qualitative sense, we choose for simplicity to carry out refinement according to a predefined schedule, as described below.

We start from a uniform mesh as in the top graph of Fig. 11, where all element nodes are indicated with blue dots, and for simplicity the inter-element boundaries are not explicitly indicated. We describe the refinement process below only for the positive part ($x > 0$) of the domain, and note that the negative part follows from symmetry. We start by refining the element whose left boundary is at $x = 0$, i.e. where the solution gradients are steepest. After every second time step, one more element is refined, proceeding outward towards the domain boundary. This continues until $t = 1/2$, at which point exactly half of the domain has been refined. The same process is now repeated on the newly refined part of the domain, thus creates an even finer mesh on the innermost quarter of the domain.

After each cycle of refinement in space, the time step size is correspondingly reduced by one half to keep the ratio between Δt and Δx constant, and the resulting total simulation time is $t = 1/2, 3/4, 7/8$, etc. after the first, second and third cycle, etc. The result of this refinement strategy is a process where the simulation time approaches $t = 1$, but in theory could go on indefinitely using smaller and smaller time steps. See Fig. 11 for the result at a few different points in time, starting from a relatively coarse mesh. We are able to capture the solution in a smooth way with high accuracy, even as we are approaching the point of discontinuity.

In absolute terms, the error still increases as we approach the discontinuity, as can be inferred from the clearly visible jump in the discontinuous solution at $x = 0$, in the bottom graph of Fig. 11. As a final test, we confirm that the solution converges with a high order of accuracy at any point in time, i.e. after any given number of AMR refinement cycles. In Table 7 we show the convergence at $t = 7/8$, i.e. after three cycles of refinement, where we have estimated the maximum error to be the jump in solution value at $x = 0$. Asymptotically, the convergence rate is $s + 1 = 3$.

6 Conclusions

We have developed a provably stable dynamical adaptive mesh refinement procedure, where computational nodes can be added or removed as needed during a simulation. The method is based on inner product preserving interpolation, and can be applied to various classes of numerical methods. Stability was analyzed within the framework of transmission problems, coupling a sequence of semi-discrete computations in time. To illustrate and evaluate the procedure, we have designed new interpolation operators for finite difference applications using moving non-collocated block interfaces. We have also applied L_2 projection operators for Galerkin elements with Gauss-Lobatto collocation.

In terms of performance we found that, due to the accuracy limitations inherent in the inner product preserving approach, the mesh should only be modified in places where the solution gradients are significantly smaller than in the more finely resolved region of interest. Moreover, in terms of interpolation accuracy, we found that adding more computational nodes to a mesh is a less sensitive operation (by an order of magnitude) than to remove nodes. Finally, we have demonstrated with a numerical example the key importance of satisfying the transmission stability definition strictly. Hence, despite the accuracy limitations, we can not hope to do better by abandoning the inner product preserving approach.

Acknowledgements This work is based on research supported in part by the National Research Foundation of South Africa (Grant Numbers: 89916). The opinions, findings and conclusions or recommendations expressed is that of the authors alone, and the NRF accepts no liability whatsoever in this regard. Jan Nordström was supported by Vetenskapsrådet, Sweden (award number 2018-05084_VR)

Funding Open Access funding provided by Linköping University.

Compliance with ethical standards

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

A New Operators

We consider classical SBP finite difference operators satisfying (13) and (14), and we have considered values of s between 1 and 3 (i.e. using second to sixth order central differences in the block interiors). All the operators listed below are both IPP (5) and contractive (10), and hence they can be used both in an explicit and an implicit AMR implementation. Operators corresponding to the forward interpolation in the schematic of Fig. 1 are accurate to order $p = s$, and operators corresponding to the opposite direction are accurate to order $p = s - 1$.

A.1 Second Order AMR

We consider the situation illustrated in Fig. 1, i.e. the interface is between a fine (to the left) and a coarse (to the right) grid, and is moving towards the right. The simplest possible AMR interpolation operator for $s = 1$ is given by $\hat{P}^n = \Delta x \text{Diag}(0.5, 1, 2)$, $\hat{P}^{n+1} = \Delta x \text{Diag}(1, 1, 0.5, 1)$, and

$$\hat{I}_{FW} = \begin{pmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

An optimized version (in terms of leading order truncation error terms), which is the one we have used for numerical calculations, is given by $\hat{P}^n = \Delta x \text{Diag}(1, 0.5, 1, 2, 2)$, $\hat{P}^{n+1} = \Delta x \text{Diag}(1, 1, 1, 0.5, 1, 2)$, and

$$\hat{I}_{FW} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 \\ 0 & 0 & a_{33} & a_{34} & 0 \\ 0 & 0 & 0 & a_{44} & 0 \\ 0 & 0 & a_{53} & a_{54} & a_{55} \\ 0 & 0 & a_{63} & a_{64} & a_{65} \end{pmatrix},$$

where

$$\begin{aligned} a_{11} &= 0.889156241852928, & a_{23} &= 0.314821100711725, & a_{54} &= 0.761208915202490, \\ a_{12} &= -0.018913262067664, & a_{24} &= 0.055421879073538, & a_{55} &= 0.119395542398755, \\ a_{13} &= 0.185178899288274, & a_{33} &= 0.5, & a_{63} &= -0.059697771199377, \\ a_{14} &= -0.055421879073537, & a_{34} &= 0.5, & a_{64} &= 0.119395542398755, \\ a_{21} &= 0.110843758147073, & a_{44} &= 1, & a_{65} &= 0.940302228800622, \\ a_{22} &= 0.518913262067664, & a_{53} &= 0.119395542398755. \end{aligned}$$

A.2 Fourth Order AMR

In the fourth order interior case ($s = 2$), we have used

$$\begin{aligned} \hat{P}^n &= \Delta x \text{Diag} \left(\frac{49}{48}, \frac{43}{48}, \frac{59}{48}, \frac{17}{48}, \frac{17}{24}, \frac{59}{24}, \frac{43}{24}, \frac{49}{24}, 2 \right), \\ \hat{P}^{n+1} &= \Delta x \text{Diag} \left(1, 1, \frac{49}{48}, \frac{43}{48}, \frac{59}{48}, \frac{17}{48}, \frac{17}{24}, \frac{59}{24}, \frac{43}{24}, \frac{49}{24} \right), \end{aligned}$$

and

$$\hat{I}_{FW} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & 0 & 0 \\ 0 & a_{32} & a_{33} & a_{34} & a_{35} & 0 & 0 \\ 0 & 0 & 0 & a_{44} & a_{45} & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{55} & 0 & 0 \\ 0 & 0 & 0 & a_{64} & a_{65} & a_{66} & 0 \\ 0 & 0 & 0 & a_{74} & a_{75} & a_{76} & a_{77} \\ 0 & 0 & 0 & 0 & a_{85} & a_{86} & a_{87} \end{pmatrix},$$

where

$$\begin{aligned} a_{11} &= 0.946416113335337, & a_{33} &= 0.453799023431231, & a_{74} &= -0.073722490074461, \\ a_{12} &= 0.107167773329325, & a_{34} &= 0.324592674150933, & a_{75} &= 0.178934963579946, \\ a_{13} &= -0.053583886664663, & a_{35} &= 0.073869434139278, & a_{76} &= 0.863297543063492, \\ a_{21} &= 0.053583886664662, & a_{44} &= 0.5, & a_{77} &= 0.031489983431023, \\ a_{22} &= 0.745093358392117, & a_{45} &= 0.5, & a_{85} &= -0.031489983431023, \\ a_{23} &= 0.099784863233432, & a_{55} &= 1, & a_{86} &= 0.062979966862046, \\ a_{24} &= 0.175407325849068, & a_{64} &= 0.147444980148924, & a_{87} &= 0.968510016568977, \\ a_{25} &= -0.073869434139280, & a_{65} &= 0.705110039702153, \\ a_{32} &= 0.147738868278557, & a_{66} &= 0.147444980148923. \end{aligned}$$

A.3 Sixth Order AMR

Finally, in the sixth order interior case ($s = 3$), we have used

$$\begin{aligned} \hat{p}^n &= \Delta x \text{Diag} \left(\frac{583}{575}, \frac{1435}{1574}, \frac{5359}{4320}, \frac{647}{1031}, \frac{1129}{812}, \frac{521}{1649}, \frac{1042}{1649}, \frac{1129}{406}, \frac{1417}{1129}, \frac{5359}{2160}, \frac{1435}{787}, \frac{1239}{611}, 2 \right), \\ \hat{p}^{n+1} &= \Delta x \text{Diag} \left(1, \frac{583}{575}, \frac{1435}{1574}, \frac{5359}{4320}, \frac{647}{1031}, \frac{1129}{812}, \frac{521}{1649}, \frac{1042}{1649}, \frac{1129}{406}, \frac{1417}{1129}, \frac{5359}{2160}, \frac{1435}{787}, \frac{1239}{611} \right). \end{aligned}$$

The coefficients of $\hat{I}_{FW}$ are omitted here to conserve space, but are available upon request to the authors.

A.4 Operators for 2D Calculations

For the moving collocated interfaces in the 2D test cases, in the fourth order interior case ($s = 2$) we have used

$$\begin{aligned} \hat{p}^n &= \Delta x \text{Diag} \left(\frac{49}{48}, \frac{43}{48}, \frac{59}{48}, \frac{17}{48}, \frac{17}{48}, \frac{59}{48}, \frac{43}{48}, \frac{49}{48}, 1 \right), \\ \hat{p}^{n+1} &= \Delta x \text{Diag} \left(1, \frac{49}{48}, \frac{43}{48}, \frac{59}{48}, \frac{17}{48}, \frac{17}{48}, \frac{59}{48}, \frac{43}{48}, \frac{49}{48} \right), \end{aligned}$$

and

$$I_{FW} = \begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & a_{24} & 0 & 0 & 0 & 0 & 0 \\ a_{31} & a_{22} & a_{33} & a_{34} & 0 & a_{26} & a_7 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & a_{46} & a_{47} & 0 & 0 \\ 0 & a_{52} & a_{53} & a_{54} & a_{55} & a_{56} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_{65} & a_{66} & a_{67} & a_{68} & 0 \\ 0 & 0 & 0 & 0 & a_{75} & a_{76} & a_{77} & a_{78} & 0 \\ 0 & 0 & 0 & 0 & a_{85} & a_{86} & a_{87} & a_{88} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{95} & a_{96} & a_{97} & a_{98} \end{pmatrix},$$

where

$$\begin{aligned}
 a_{11} &= 1 & a_{43} &= 0.457627118644263 & a_{68} &= -0.176470588235344 \\
 a_{21} &= 0.061224489795927 & a_{44} &= 0.135593220338862 & a_{75} &= -0.152542372881395 \\
 a_{22} &= 0.816326530612189 & a_{45} &= 0.237288135593209 & a_{76} &= 0.457627118644179 \\
 a_{23} &= 0.183673469387840 & a_{46} &= 0.406779661016931 & a_{77} &= 0.542372881355828 \\
 a_{24} &= -0.061224489795957 & a_{47} &= -0.101694915254204 & a_{78} &= 0.152542372881388 \\
 a_{31} &= -0.069767441860487 & a_{52} &= 0.176470588235362 & a_{85} &= 0.069767441860485 \\
 a_{32} &= 0.209302325581550 & a_{53} &= -0.529411764706101 & a_{86} &= -0.209302325581465 \\
 a_{33} &= 0.744186046511349 & a_{54} &= 0.176470588235448 & a_{87} &= 0.209302325581476 \\
 a_{34} &= 0.209302325581544 & a_{55} &= 0.352941176470665 & a_{88} &= 0.930232558139505 \\
 a_{36} &= -0.139534883720902 & a_{56} &= 0.823529411764625 & a_{95} &= 0.020408163265312 \\
 a_{37} &= 0.046511627906946 & a_{65} &= 0.176470588235345 & a_{96} &= -0.061224489795928 \\
 a_{41} &= 0.016949152542381 & a_{66} &= 0.470588235293964 & a_{97} &= 0.061224489795923 \\
 a_{42} &= -0.152542372881442 & a_{67} &= 0.529411764706035 & a_{98} &= 0.979591836734694.
 \end{aligned}$$

Moreover, we have used the following interface interpolation operator, defining an IPP pair along the tangent direction of non-located interfaces

$$I_{f2c} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & a_{16} & a_{17} & a_{18} & a_{19} & a_{110} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & a_{26} & a_{27} & a_{28} & a_{29} & a_{210} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & a_{36} & a_{37} & a_{38} & a_{39} & a_{310} \\ & & & -1/32 & 0 & 9/32 & 1/2 & 9/32 & 0 & -1/32 \\ & & & & & -1/32 & 0 & 9/32 & 1/2 & 9/32 & 0 & -1/32 \\ & & & & & & & \ddots & & \ddots & \ddots & \ddots & \ddots \end{pmatrix},$$

where

$$\begin{aligned}
 a_{11} &= 0.495787011428777, & a_{21} &= 0.002427823922400, & a_{31} &= -0.001665600132809, \\
 a_{12} &= 0.705272991724725, & a_{22} &= 0.343571496294226, & a_{32} &= -0.064194398620458, \\
 a_{13} &= -0.018803098705387, & a_{23} &= 0.375242463660731, & a_{33} &= -0.007433783209106, \\
 a_{14} &= -0.092984662839436, & a_{24} &= 0.235258110788828, & a_{34} &= 0.319343272830920, \\
 a_{15} &= -0.013273406150234, & a_{25} &= 0.007649081510304, & a_{35} &= 0.552891909196419, \\
 a_{16} &= -0.028709629126509, & a_{26} &= -0.004112247622012, & a_{36} &= 0.289521774531380, \\
 a_{17} &= -0.067822291189219, & a_{27} &= 0.047558608481923, & a_{37} &= -0.050069277912017, \\
 a_{18} &= -0.018552692763343, & a_{28} &= 0.015458331422943, & a_{38} &= -0.055299901790159, \\
 a_{19} &= 0.007040437669086, & a_{29} &= -0.004057201368626, & a_{39} &= 0.002783428845917, \\
 a_{110} &= 0.032045339951541, & a_{210} &= -0.018996467090719, & a_{310} &= 0.014122576259912.
 \end{aligned}$$

References

- Berger, M., Colella, P.: Local adaptive mesh refinement for shock hydrodynamics. *J. Comput. Phys.* **82**(1), 64–84 (1989)
- Berger, M.J., LeVeque, R.J.: Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM J. Numer. Anal.* **35**(6), 2298–2316 (1998)
- Berger, M.J., Oliger, J.: Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.* **53**(3), 484–512 (1984)
- Del Rey Fernández, D.C., Hicken, J.E., Zingg, D.W.: Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. *Comput. Fluids* **95**, 171–196 (2014)

5. Fisher, T.C., Carpenter, M.H., Nordström, J., Yamaleev, N.K., Swanson, C.: Discretely conservative finite-difference formulations for nonlinear conservation laws in split form: theory and boundary conditions. *J. Comput. Phys.* **234**, 353–375 (2013)
6. Friedrich, L., Del Rey Fernández, D.C., Winters, A.R., Gassner, G.J., Zingg, D.W., Hicken, J.E.: Conservative and stable degree preserving SBP operators for non-conforming meshes. *J. Sci. Comput.* **75**, 657–686 (2018)
7. Friedrich, L., Winters, A.R., Del Rey Fernández, D.C., Gassner, G.J., Parsani, M., Carpenter, M.H.: An entropy stable h/p non-conforming discontinuous Galerkin method with the summation-by-parts property. *J. Sci. Comput.* **77**(2), 689–725 (2018)
8. Gassner, G.J.: A skew-symmetric discontinuous Galerkin spectral element discretization and its relation to SBP-SAT finite difference methods. *SIAM J. Sci. Comput.* **35**, A1233–A1253 (2013)
9. Gustafsson, B.: *High Order Difference Methods for Time Dependent PDE*. Springer, Berlin (2008)
10. Hicken, J.E., Zingg, D.W.: Summation-by-parts operators and high order quadrature. *J. Comput. Appl. Math.* **237**(1), 111–125 (2013)
11. Kopera, M.A., Giraldo, F.X.: Analysis of adaptive mesh refinement for IMEX discontinuous Galerkin solutions of the compressible Euler equations with application to atmospheric simulations. *J. Comput. Phys.* **275**, 92–117 (2014)
12. Kopriva, D.A.: A conservative staggered-grid Chebyshev multidomain method for compressible flows. II. A semi-structured method. *J. Comput. Phys.* **128**(2), 475–488 (1996)
13. Kopriva, D.A., Woodruff, S.L., Hussaini, M.Y.: Computation of electromagnetic scattering with a non-conforming discontinuous spectral element method. *Int. J. Numer. Methods Eng.* **53**, 105–122 (2002)
14. Kozdon, J.E., Wilcox, L.C.: Stable coupling of nonconforming, high-order finite difference methods. *SIAM J. Sci. Comput.* **38**, A923–A952 (2016)
15. Kreiss, H.O., Scherer, G.: Finite element and finite difference methods for hyperbolic partial differential equations. In: De Boor, C. (ed.) *Mathematical Aspects of Finite Elements in Partial Differential Equation*. Academic Press, New York (1974)
16. Linders, V., Lundquist, T., Nordström, J.: On the order of accuracy of finite difference operators on diagonal norm based summation-by-parts form. *SIAM J. Numer. Anal.* **56**, 1048–1063 (2018)
17. Liu, Y., Sarris, C.D.: Dynamically adaptive mesh refinement FDTD: a stable and efficient technique for time-domain simulations. In: 2006 12th International Symposium on Antenna Technology and Applied Electromagnetics and Canadian Radio Sciences Conference, pp. 1–4 (2006)
18. Lundquist, T., Malan, A., Nordström, J.: A hybrid framework for coupling arbitrary summation-by-parts schemes on general meshes. *J. Comput. Phys.* **362**, 49–68 (2018)
19. Lundquist, T., Nordström, J.: Stable and accurate filtering procedures. *J. Sci. Comput.* (2020). <https://doi.org/10.1007/s10915-019-01116-9>
20. Mattsson, K., Carpenter, M.H.: Stable and accurate interpolation operators for high-order multiblock finite difference methods. *SIAM J. Sci. Comput.* **32**, 2298–2320 (2010)
21. Mavriplis, D.J.: Nonconforming discretizations and a posteriori error estimators for adaptive spectral element techniques. PhD thesis, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics (1989)
22. Nordström, J., Linders, V.: Well-posed and stable transmission problems. *J. Comput. Phys.* **364**, 95–110 (2018)
23. Nordström, J., Lundquist, T.: Summation-by-parts in time: the second derivative. *SIAM J. Sci. Comput.* **38**(3), A1561–A1586 (2016)
24. Papoutsakis, A., Sazhin, S.S., Begg, S., Danaila, I., Luddens, F.: An efficient adaptive mesh refinement (AMR) algorithm for the discontinuous Galerkin method: applications for the computation of compressible two-phase flows. *J. Comput. Phys.* **363**, 399–427 (2018)
25. Svård, M., Nordström, J.: Review of summation-by-parts schemes for initial-boundary-value problems. *J. Comput. Phys.* **268**, 17–38 (2014)
26. Svård, M., Nordström, J.: On the convergence rates of energy-stable finite-difference schemes. *J. Comput. Phys.* **397**, 108819 (2019)
27. Verfürth, R.: A posteriori error estimation and adaptive mesh-refinement techniques. *J. Comput. Appl. Math.* **50**(1), 67–83 (1994)