Adaptive hyperviscosity stabilisation for the RBF-FD method in solving advection-dominated transport equations

Miha Rot[a,b,*], Žiga Vaupotič[a,c], Andrej Kolar-Požun[a,c], Gregor Kosec[a]

[a] *Parallel and Distributed Systems Laboratory, Jožef Stefan Institute, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[b] *Jozef Stefan International Postgraduate School, Jamova cesta 39, 1000 Ljubljana, Slovenia*
[c] *Faculty of Mathematics and Physics, University of Ljubljana, Jadranska ulica 19, 1000 Ljubljana, Slovenia*

Abstract

This paper presents an adaptive hyperviscosity stabilisation procedure for the Radial Basis Function-generated Finite Difference (RBF-FD) method, aimed at solving linear and non-linear advection-dominated transport equations on domains without a boundary. The approach employs an equation independent algorithm that adaptively determines the hyperviscosity constant based on the spectral radius of the RBF-FD evolution matrix. The proposed procedure supports general node layouts and is not tailored for specific equations, avoiding the limitations of empirical tuning and von Neumann-based estimates. To reduce computational cost, it is shown that lower monomial augmentation in the approximation of the hyperviscosity operator can still ensure consistent stabilisation, enabling the use of smaller stencils and improving overall efficiency. A hybrid strategy employing different spline orders for the advection and hyperviscosity operators is also implemented to enhance stability. The method is evaluated on pure linear advection and non-linear Burgers' equation, demonstrating stable performance with limited numerical dissipation. The two main contributions are: (1) a general hyperviscosity RBF-FD solution procedure demonstrated on both linear and

*Corresponding author
Email addresses: miha.rot@ijs.si (Miha Rot), gregor.kosec@ijs.si (Gregor Kosec)

non-linear advection-dominated problems, and (2) an in-depth analysis of the behaviour of hyperviscosity within the RBF-FD framework, addressing the interplay between key free parameters and their influence on numerical results.

Keywords: RBF-FD, hyperviscosity, stabilisation scheme, eigenvalue stability, meshless, advection, Burgers' equation

1. Introduction

Transport Partial Differential Equations (PDE) arise from fundamental conservation principles and provide a mathematical framework to describe how conserved quantities move and evolve over time and space. Examples of such PDEs include the advection equation for the transport of a quantity by a moving fluid, the diffusion equation for describing the spreading of a substance due to random motion, the Navier-Stokes equation governing fluid motion through momentum conservation, and its simplified variant, Burgers' equation, capturing key non-linear and viscous effects while omitting pressure, to name a few. Solving these equations is crucial for understanding physical processes in various fields ranging from pure scientific research, to environmental modelling and real-world engineering applications. Transport equations are typically solved numerically due to their complexity. Nevertheless, even numerical simulations, especially in advection-dominated regimes, can be challenging. If not managed correctly, they may produce non-physical oscillations, which can potentially lead to divergence [1].

In most numerical simulations, mesh-based methods such as the Finite Volume Method (FVM), or the Finite Element Method (FEM) are used. However, due to meshing limitations an alternative class of meshless methods emerged in 1970s [2]. The fundamental distinction between traditional mesh-based methods and meshless methods lies in how they establish relationships between computational nodes. Unlike mesh-based approaches, meshless methods define these relationships solely based on relative node positions, allowing them to operate on scattered nodes without relying on any predefined structure or mesh [3]. Although achieving a stable meshless approximation requires scattered nodes to follow specific guidelines [4, 5], the process of positioning these nodes is far less complex than traditional meshing [6]. This makes meshless methods highly flexible for handling complex geometries, including those supplied by Computer-Aided Design (CAD)

models [7, 8], as well as for applications in h-adaptivity [9, 10, 11] and moving boundary problems [12, 13]. Scattered node positioning algorithms can also be implemented in a dimension-independent manner [4, 14] that can be combined with the elegant formulation of meshless approximation techniques to facilitate solution procedures for high-dimensional problems [15].

Among emerging meshless methods, the Radial Basis Function-generated Finite Differences (RBF-FD)[16] has in the last two decades gained significant attention for its adaptability and robustness in solving various PDEs. RBF-FD most commonly employs polyharmonic spline (PHS) radial basis functions (RBFs) augmented with monomials [17, 18] for differential operator approximation, which helps prevent stagnation errors and allows control over the convergence rate. Besides improved stability, the flexibility in the order of approximation has recently led to the development of the first truly meshless hp-adaptive method [19], high-order solutions for various problems [20, 21, 15], and the implementation of spatially variable operator approximations to improve computational performance [22].

Nonetheless, despite the potential advantages of the RBF-FD method, its differentiation matrices inherently contain spurious eigenvalues [23, 24]. This effect remained theoretically unexplained until recently [25], when Tominec et al. provided a mathematical foundation for it within the framework of energy methods, a common approach in finite element analysis. The authors demonstrated that instabilities stem from integration errors and discontinuities in global cardinal functions, which were already discussed in [26]. In that work, the authors introduced an oversampling technique, a method for stabilising RBF partition of unity method (RBF-PUM) [27] and Kansa formulations [25, 28], where global cardinal functions do not exhibit jumps. On the other hand, the jumps in global cardinal functions present in the RBF-FD method can be reduced by increasing the stencil size [25], a conditionally useful approach due to its high computational cost.

In practice, dissipative numerical schemes are used for stabilisation. Arguably the most widely used dissipative stabilisation technique is upwind [29]. Upwind is a directional biasing approach in which, when computing numerical derivatives, greater weight is assigned to values from the direction where information is coming from. Upwind schemes are widely used in computational fluid dynamics and numerical simulations where convective transport is dominant [30]. Nonetheless, while upwind stabilises numerical simulations that would otherwise diverge, it also introduces significant artificial diffusion [30], which consequently smears out sharp features in the final so-

lution [31].

In an alternative stabilisation technique, referred to as hyperviscosity, a higher-order diffusion term, i.e. fourth or higher order derivatives, is added to the governing equations. Such hyperviscosity operator selectively damps small-scale oscillations while preserving large-scale structures, i.e. it controls high-frequency instabilities. The concept of a hyperviscosity operator dates back to the 1990s [32, 33] and was later adapted to RBF-FD methods [24], where authors demonstrated its effectiveness in shifting the eigenvalues of the differentiation matrix for convective PDEs to a stable region. In the context of RBF-FD, hyperviscosity has been recently also demonstrated on advection-diffusion PDEs [34] and advection-diffusion-reaction PDEs on manifolds [35].

The main problem with the hyperviscosity scheme is its inherent constant, typically denoted by γ, which must be tuned specifically for each case. Although an empirical estimate for this constant was proposed [36, 37], user input was still required to determine its precise value. It was not until recently that the first attempt to theoretically determine an appropriate γ was introduced by Shankar et al. [34], using a quasi-analytical approach based on von Neumann analysis. However, in addition to standard limitations of von Neumann analysis, such as its validity only for periodic or infinite domains, linear PDEs, uniform spatial discretisation, and its disregard for boundary effects, the proposed approach also requires an estimate of an additional growth term to account for scattered nodes and RBF-FD interpolation. This term is not well-defined, which may lead to inaccurate results and, consequently, an ill-determined γ.

This paper advances the applicability of hyperviscosity by refining the understanding and determination of hyperviscosity constants. The primary goal is to develop a PDE-independent approach for optimising the parameters of the hyperviscosity scheme on domains without a boundary and to enhance the understanding of their impact on accuracy and stability. We demonstrate that an appropriate hyperviscosity constant can be estimated through the eigenvalues of the evolution matrix, as indicated by Fornberg et al. [24], by using a relatively simple iterative algorithm. The proposed approach is not restricted by the type of PDE, domain geometry, RBF-FD setup, or the layout of scattered nodes.

Another challenge of hyperviscosity is its high computational cost, as increasing its order significantly enlarges the required stencil. For example, a stable computation of fourth derivatives, as required by the second-order

hyperviscosity operator in 2D, necessitates augmentation with fourth-order monomials and a corresponding stencil of 30 support nodes [17]. Here, we demonstrate that the hyperviscosity operator exhibits consistent behaviour even by using a smaller monomial augmentation in constructing the approximation. This finding suggests that computational costs can be reduced by employing only second-order augmentation, regardless of the order of the hyperviscosity operator. The developed methodology is demonstrated on a linear advection-dominated problem [38, 34] and a non-linear Burgers' problem [39, 40], where we clearly illustrate its efficiency.

The paper is organised in the following order: In section 2 we provide a description of the RBF-FD method and use it to discretise linear and non-linear Cauchy problems in the matrix-vector format. In section 3 we construct a hyperviscosity operator and theoretically analyse its parameters. The hyperviscosity scheme and parameters are first numerically verified on simple linear advection case in subsection 4.1. These results are extended to include the non-linear Burgers' equation in subsection 4.2. We finish the paper with some remarks on the future work in section 5.

2. The RBF-FD method

First, we discuss the RBF-FD approach to discretising differential operators on scattered nodes spanning the domain Ω and therefore constructing the appropriate differentiation matrices.

2.1. RBF approximation

The domain $\Omega \subseteq \mathbb{R}^d$ is first discretised using a set of quasi-uniformly distributed scattered nodes $X = \{\boldsymbol{x}_i\}_{i=1}^N \subset \Omega$ that are generated using a dedicated advancing front algorithm [4]. Each node $\boldsymbol{x}_i \in X$ is assigned a subset $X^{(i)} \subseteq X$ of the closest nodes n (commonly referred to as a stencil or support domain) that are used for construction of a local interpolation in the vicinity of that node. Using RBFs $\phi_i(\boldsymbol{x}) = \phi(\|\boldsymbol{x}_i - \boldsymbol{x}\|), i = 1, \dots, n$ and a multivariate monomial basis $\{m_j\}_{j=1}^q$, the local interpolation reads

$$\begin{aligned} u_h(\boldsymbol{x}) &= \sum_{i=1}^{n} w_i \phi_i(\boldsymbol{x}) + \sum_{i=1}^{q} \beta_i m_i(\boldsymbol{x}), \\ \text{subject to:} \quad & \sum_{i=1}^{n} w_i m_j(\boldsymbol{x}_i) = 0 \quad j = 1, \dots, q, \end{aligned} \tag{1}$$

where w_i are the RBF interpolation weights and β_i are the Lagrange multipliers. To obtain the weights a linear system is assembled enforcing the pointwise interpolation in equation (1) for each point in the stencil:

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{P} \\ \boldsymbol{P}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{w} \\ \boldsymbol{\beta} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u_h} \\ 0 \end{bmatrix}, \tag{2}$$

where $\boldsymbol{w} = [w_i]_{i=1}^n$ is the interpolation weight vector, $\boldsymbol{\beta} = [\beta_i]_{i=1}^q$ is the Lagrange multipliers vector, $\boldsymbol{u_h} = [u_h(\boldsymbol{x}_i)]_{i=1}^n$ is a vector of stencil nodal values, and $\boldsymbol{A}_{i,j} = \phi_i(\boldsymbol{x}_j)$ and $\boldsymbol{P}_{i,j} = m_j(\boldsymbol{x}_i)$ are interpolation matrices. In this paper, we use conditionally positive definite RBFs, more specifically, polyharmonic splines $\phi(r) = r^k$, where $r = \|\boldsymbol{x}_j - \boldsymbol{x}\|_2$ is the Euclidean distance to the centre and $k \in \mathbb{N}$ is odd. Monomials are added to the local interpolation (1) to ensure the unisolvency of the local interpolation system (2) for conditionally positive definite RBFs [41, 42] and to improve the convergence characteristics [15, 43]. To prove unisolvency, we require a monomial tensor basis $\{m_j\}_{j=1}^q$ of size $q = \binom{m+d}{m}$ to span the polynomial space of order m, where $m \geq \lceil \frac{k}{2} \rceil - 1$ [42]. The recommended stencil size n for 2D systems is $n = 2q$ [41].

In subsection 3.2, where we discuss the consistency of the hyperviscosity scheme, we will use the concept of global cardinal functions. To approximate the linear differential operator $\mathcal{L}$ acting on function $u : \Omega \to \mathbb{R}^d$, we use the global cardinal functions $\Psi_i : \Omega \to \mathbb{R}, i = 1, \ldots, N$ with the following ansatz

$$u_h(\boldsymbol{x}) = \sum_{i=1}^{N} \Psi_i(\boldsymbol{x}) u_h(\boldsymbol{x}_i) \qquad (\mathcal{L}u_h)(\boldsymbol{x}) = \sum_{i=1}^{N} \mathcal{L}\Psi_i(\boldsymbol{x}) u_h(\boldsymbol{x}_i), \tag{3}$$

where $u_h(\boldsymbol{x}_1), \ldots, u_h(\boldsymbol{x}_N)$ are the degrees of freedom. The global cardinal functions are constructed by blending the local interpolation systems for every stencil in the domain. A local interpolation system for node i is (re)introduced via local cardinal functions $\psi_j^{(i)} : \Omega \to \mathbb{R}$ and values $u_h(\boldsymbol{x}_j)$ in stencil nodes $\boldsymbol{x}_j \in X^{(i)}$

$$u_h^{(i)}(\boldsymbol{x}) = \sum_{x_j \in X^{(i)}} \psi_j^{(i)}(\boldsymbol{x}) u_h(\boldsymbol{x}_j) \qquad (\mathcal{L}u_h^{(i)})(\boldsymbol{x}) = \sum_{x_j \in X^{(i)}} \mathcal{L}\psi_j^{(i)}(\boldsymbol{x}) u_h(\boldsymbol{x}_j). \tag{4}$$

The equation (1) can now be rewritten using the local cardinal functions (4) as

$$
\begin{aligned}
u_h^{(i)}(\boldsymbol{x}) &= \begin{bmatrix}\boldsymbol{\phi}^{(i)}(\boldsymbol{x}) & \boldsymbol{m}^{(i)}(\boldsymbol{x})\end{bmatrix} \begin{bmatrix}\boldsymbol{w}^{(i)} \\ \boldsymbol{\beta}^{(i)}\end{bmatrix} \\
&= \begin{bmatrix}\boldsymbol{\phi}^{(i)}(\boldsymbol{x}) & \boldsymbol{m}^{(i)}(\boldsymbol{x})\end{bmatrix} \begin{bmatrix}\boldsymbol{A}^{(i)} & \boldsymbol{P}^{(i)} \\ \left(\boldsymbol{P}^{(i)}\right)^T & 0\end{bmatrix}^{-1} \begin{bmatrix}\boldsymbol{u}_h^{(i)} \\ 0\end{bmatrix} \\
&= \begin{bmatrix}\boldsymbol{\psi}^{(i)}(\boldsymbol{x}) & \boldsymbol{\xi}^{(i)}(\boldsymbol{x})\end{bmatrix} \begin{bmatrix}\boldsymbol{u}_h^{(i)} \\ 0\end{bmatrix},
\end{aligned} \tag{5}
$$

where $\boldsymbol{\psi}^{(i)}(\boldsymbol{x}) = [\psi_j^{(i)}(\boldsymbol{x})]_{\boldsymbol{x}_j \in X^{(i)}}$ is the local cardinal function vector.

2.2. Spatial discretisation of PDEs

To construct the global cardinal functions, for every computational point in the domain we merge all the local cardinal functions used for interpolating that point in local stencils

$$
\Psi_i(\boldsymbol{x}) = \begin{cases} \psi_i^{(\sigma(\boldsymbol{x}))}(\boldsymbol{x}), & \boldsymbol{x}_i \in X^{(\sigma(\boldsymbol{x}))} \\ 0, & \boldsymbol{x}_i \notin X^{(\sigma(\boldsymbol{x}))} \end{cases}, \tag{6}
$$

where the function $\sigma(\boldsymbol{x})$ returns the index of the closest computational point to any point $\boldsymbol{x} \in \Omega$. Both global and local cardinal functions have the Kronecker delta property

$$
u_h(\boldsymbol{x}_j) = \sum_{i=1}^{N} \Psi_i(\boldsymbol{x}_j) u_h(\boldsymbol{x}_i) = \sum_{i=1}^{N} \delta_{i,j} u_h(\boldsymbol{x}_i). \tag{7}
$$

This ensures that the evaluation matrix for sampling the pointwise values of the function is an identity matrix. This also means that the mass matrix does not need to be inverted, which is a considerable overhead in the finite element method. To construct the global differentiation matrix, we assemble the evaluated derivatives of global cardinal functions at every $\boldsymbol{x}_i \in X$ in an $N \times N$ matrix. In practice, the evaluated derivatives of cardinal functions are calculated by solving the local interpolation system (2) with $\begin{bmatrix}\mathcal{L}\boldsymbol{\phi}(\boldsymbol{x}_i) & \mathcal{L}\boldsymbol{m}(\boldsymbol{x}_i)\end{bmatrix}$ as the right hand side [43].

To put the above-mentioned into perspective, we consider the semidiscrete scheme. Here, we are especially interested in solving linear and non-linear advection, however, for the sake of generality, we consider a Cauchy problem

on a domain Ω without a boundary. Specifically, find $u : \Omega \times [0, T] \to \mathbb{R}^m$ such that

$$\begin{cases} \partial_t u(\boldsymbol{x}, t) + \mathcal{N}(u(\boldsymbol{x}, t)) = 0 & \text{in } \Omega \times (0, T] \\ u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) & \text{in } \Omega, \end{cases} \tag{8}$$

where $\mathcal{N}$ is a (non)linear operator. If $\mathcal{N}$ is nonlinear, we work with its linearisation $\mathcal{L}$. To spatially discretise the equation (8) we employ the ansatz (3) for each point $\boldsymbol{x} \in X$ and obtain a set of N equations

$$\begin{aligned} \partial_t \sum_{i=1}^{N} \Psi_i(\boldsymbol{x}_1) u_h(\boldsymbol{x}_i) &= -\sum_{i=1}^{N} \mathcal{L}\Psi_i(\boldsymbol{x}_1) u_h(\boldsymbol{x}_i), \\ &\vdots \\ \partial_t \sum_{i=1}^{N} \Psi_i(\boldsymbol{x}_N) u_h(x_i) &= -\sum_{i=1}^{N} \mathcal{L}\Psi_i(\boldsymbol{x}_N) u_h(\boldsymbol{x}_i). \end{aligned} \tag{9}$$

We introduce the differentiation matrix

$$(\boldsymbol{D}_h^{\mathcal{L}})_{j,i} = \mathcal{L}\Psi_i(x_j) \tag{10}$$

to rewrite the obtained set of equations in the vector-matrix form

$$\partial_t \boldsymbol{u_h}(t) = -\boldsymbol{D}_h \boldsymbol{u_h}(t). \tag{11}$$

To integrate the obtained system of differential equations (11), we primarily use the backwards Euler scheme. This particular scheme will be useful due to its fundamental unconditional stability. Given a timestep size of $\Delta t > 0$, we discretise the time domain and obtain

$$\boldsymbol{u_h}(t_{n+1}) = \boldsymbol{u_h}(t_n) - \Delta t \boldsymbol{D}_h \boldsymbol{u_h}(t_{n+1}). \tag{12}$$

In practice, when solving the system (12), we rewrite it as $(I + \Delta t \boldsymbol{D}_h)\boldsymbol{u_h}(t_{n+1}) = \boldsymbol{u_h}(t_n)$ and solve the system via LU-decomposition. In the later section, we will also enforce Dirichlet boundary conditions. This is done by replacing the offending rows and columns of the boundary points with the identity matrix.

3. Hyperviscosity stabilisation

In this section, we discuss the hyperviscosity stabilisation scheme. As previously stated RBF-FD differentiation matrices tend to have spurious eigenvalues. Put simply, the RBF-FD solution exhibits numerical artefacts that may lead to instability and divergence of the time-dependent systems. To address this issue, an artificial higher-order Laplacian operator is added to the equation (8) [35, 32, 34, 44]

$$\partial_t u(\boldsymbol{x}, t) + \mathcal{N}(u(\boldsymbol{x}, t)) = (-1)^{\alpha+1}\gamma\Delta^{\alpha}u(\boldsymbol{x}, t), \tag{13}$$

or equivalently in the semi-discrete scheme (11),

$$\partial_t \boldsymbol{u_h}(t) = -\boldsymbol{D}_h\boldsymbol{u_h}(t) + (-1)^{\alpha+1}\gamma\boldsymbol{D}_h^{hip}\boldsymbol{u_h}(t). \tag{14}$$

This operator is of particular interest to meshless methods, as it allows us to control the stability of the scheme without relying on a structured mesh. Additionally, the higher order Laplace operator allows us the preserve the order with which our numerical approximation converges to the right solution.

However, as previously discussed, there is no clear consensus in the literature regarding the selection of the constant γ and the order α. In the following sections, we will discuss the importance of those two parameters and how to select them to satisfy the conditions of the Lax-equivalence theorem. The stability of the scheme is achieved by correctly choosing γ and α terms, which is described in subsection 3.1. On the other hand, the scheme is consistent if the RBF-FD parameters are correctly selected; this, along with the optimality of the parameters, is described in subsection 3.1 and subsection 3.3.

Throughout this section, we will support the discussion with numerical experiments on the stabilised advection operator

$$\hat{\boldsymbol{D}}_h = -\boldsymbol{D}_h + (-1)^{\alpha+1}\gamma\boldsymbol{D}_h^{hip} \tag{15}$$

or in some cases on the pure advection operator $\boldsymbol{D}_h$.

3.1. Note on the parameter γ and how to iteratively determine it

The estimation of the gamma term has been the subject of numerous papers [37, 36, 35, 34, 38] that provided various estimates. Authors of [37] introduced additional scaling that is also used throughout this paper

$$\gamma(c) = ch^{2\alpha}, \tag{16}$$

and provided the estimate $c \in [0.1, 1]$ for the Navier-Stokes equation; it is **assumed** that c is independent or very loosely dependent on h. This estimate was later discussed in [44]. In [45] another estimate $\gamma = 2^{-6}h^6$ was given and later generalised by Shankar as $\gamma = 2^{-2\alpha}h^{2\alpha}$ [34]. Both estimates relied on the Nyquist cut-off wavenumber $k \approx 2h^{-1}$ or what is most commonly referred to as the grid resolution. This general estimate is sometimes sufficient as we aim to target high-order modes with wavenumbers close to the grid resolution. Unfortunately, it only works for simple PDEs and can only be utilised under the assumption that the node set reproduces a grid-like structure. In our case, we primarily choose $h^{2\alpha}$ scaling of the γ term to be consistent with the Fourier theory for the hyperviscosity operator [32, 46].

To choose c, we will introduce a more general approach that relies purely on the eigenvalue decomposition of the discrete evolution matrix. For studying stability it is natural to consider the following weighted L^2-like norm [25]

$$\|u\|_h^2 = \frac{|\Omega|}{N}\sum_{i=1}^{N} u(x_i)u(x_i) \quad \|\boldsymbol{A}\|_h = \sup_{\|x\|\neq 0}\frac{\|\boldsymbol{A}x\|_h}{\|x\|_h} = \|\boldsymbol{A}\|_2, \tag{17}$$

where $|\Omega|/N \approx h^2$. We must show that our solution is stable with respect to this norm. Firstly, we construct a discrete RBF-FD evolution matrix $\boldsymbol{G}_h \in \mathbb{R}^{N\times N}$, that is

$$\boldsymbol{u_h}(t_{n+1}) = \boldsymbol{G}_h\boldsymbol{u_h}(t_n). \tag{18}$$

Here we assume that $\mathcal{N}$ is linear for simplicity. Note that the hyperviscosity term is directly included in the evolution matrix. For the implicit Euler method, this matrix is

$$\boldsymbol{u_h}(t_{n+1}) = \boldsymbol{G}_h\boldsymbol{u_h}(t_n) = \left[I - \Delta t\hat{\boldsymbol{D}}_h\right]^{-1}\boldsymbol{u_h}(t_n), \tag{19}$$

as discussed regarding the system (12). Recall that a discretisation of a linear Cauchy problem (19) is said to be *(Lax)*-stable if there exists a constant $C > 0$ independent of h, such that for all $n \in \mathbb{N}$ the following holds

$$\|\boldsymbol{u_h}(t_n)\|_h \leq C\|\boldsymbol{u_h}(0)\|_h. \tag{20}$$

This notion of stability is weaker than the more appropriate energy or entropy stability. However, it still ensures continuous dependence on the initial data. First notice that $\boldsymbol{u_h}(t_n) = \boldsymbol{G}_h^n\boldsymbol{u_h}(0)$. Assume now that $\boldsymbol{G}_h$ is diagonalizable.

Then there exists a matrix $\boldsymbol{U}_h \in \mathbb{C}^{N\times N}$ and a diagonal matrix $\boldsymbol{D} \in \mathbb{C}^{N\times N}$ such that $\boldsymbol{G}_h = \boldsymbol{U}_h \boldsymbol{D}_h \boldsymbol{U}_h^{-1}$, this allows us to estimate

$$\|\boldsymbol{G}_h^n \boldsymbol{u_h}(0)\|_h = \|\boldsymbol{U}_h \boldsymbol{D}_h^n \boldsymbol{U}_h^{-1} \boldsymbol{u_h}(0)\|_h \leq \|\boldsymbol{U}_h\|_h \|\boldsymbol{D}_h^n\|_h \|\boldsymbol{U}_h^{-1}\|_h \|\boldsymbol{u_h}(0)\|_h. \quad (21)$$

We denote the condition number by $\kappa(\boldsymbol{U}_h) = \|\boldsymbol{U}_h\|_h \|\boldsymbol{U}_h^{-1}\|_h$ and let $\rho : \mathbb{C}^{N\times N} \to \mathbb{R}^+$ be the spectral radius, i.e. ρ returns the maximal magnitude of the eigenvalues of the matrix. It follows that

$$\|\boldsymbol{G}_h^n \boldsymbol{u_h}(0)\|_h \leq \kappa(\boldsymbol{U}_h) \max_i |\lambda_i^n| \, \|\boldsymbol{u_h}(0)\|_h = \kappa(\boldsymbol{U}_h) \rho(\boldsymbol{G}_h)^n \|\boldsymbol{u_h}(0)\|_h. \quad (22)$$

If $\rho(\boldsymbol{G}_h) \leq 1$, then the system satisfies the stability condition for a fixed h. To extend this bound to the Lax-stability criteria, the condition number $\kappa(\boldsymbol{U}_h)$ must be uniformly bounded for a sufficiently small h. Furthermore, if $\kappa(\boldsymbol{U}_h)$ is not close to 1 then nonphysical growth (and decay) is still possible; hence, we must further assume that $\kappa(\boldsymbol{U}_h)$ is close to 1. We note that RBF-FD matrices are typically not normal and therefore we have no control over the condition number of this matrix. A similar condition in terms of the spectral radius can be made for the case where $\boldsymbol{G}_h$ is not diagonalizable. In this case, we can work with its Jordan canonical form $\boldsymbol{G}_h = \boldsymbol{U}_h' \boldsymbol{J}_h \boldsymbol{U}_h'^{-1}$ and again estimate

$$\|\boldsymbol{G}_h^n \boldsymbol{u_h}(0)\|_h \leq \kappa(\boldsymbol{U}_h') \|\boldsymbol{J}_h^n\|_h \|\boldsymbol{u_h}(0)\|_h \leq \kappa(\boldsymbol{U}_h') \|\boldsymbol{J}_h^n\|_F \|\boldsymbol{u_h}(0)\|_h. \quad (23)$$

We have used $\|\cdot\|_2 \leq \|\cdot\|_F$, where $\|\cdot\|_F$ is the Frobenius norm, which is bounded for all n if $\rho(\boldsymbol{G}_h) < 1$ (technically, eigenvalues that correspond to 1×1 Jordan blocks may have unit modulus) [47, Theorem 3.2.5.2.]. While this guarantees *Lax*-stability, the upper bound does not admit a nice expression in terms of $\rho(\boldsymbol{G}_h)$, unlike in the diagonalizable case. In particular this implies that we do not have any control over the potential nonphysical growth of the solution. Note that from a numerical point of view this may not be an issue as a generic matrix tends to be diagonalizable. Furthermore, due to round-off errors the conditions $\rho(\boldsymbol{G}_h) < 1$ and $\rho(\boldsymbol{G}_h) \leq 1$ are practically equivalent, justifying the usage of $\rho(\boldsymbol{G}_h) \leq 1$ as a stability condition for a general case.

We expect the eigenvalues to progressively shift into the stable region as we increase the constant c. This is demonstrated in Figure 1, where eigenvalue spectra of the advection operator $\hat{\boldsymbol{D}}_h$ (15) using $N \approx 10^3$ nodes are shown for different values of c. As expected, we observe that as c increases, more eigenvalues shift into the stable region, where c_{opt} stands for the optimal

shift determined by the following algorithm. However, we also note that the spectra become more scattered, indicating the dissipative properties of the hyperviscosity operator. Consequently, it can be anticipated that as c becomes larger, more physical properties of the system are lost.

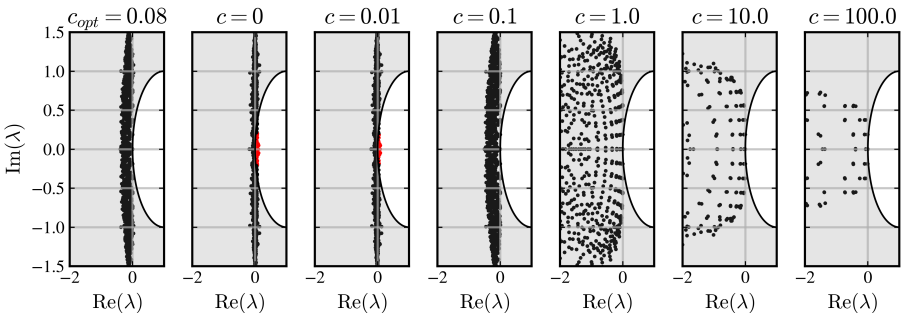

Figure 1: Eigenvalue spectra of the stabilised advection operator $\hat{\boldsymbol{D}}_h$ (15) using $N \approx 10^3$ with respect to different values of c for $\alpha = 2$. The shaded region in the figure is the implicit Euler stability region. The black dots represent stable eigenvalues, whereas the red dots represent eigenvalues that lie outside of the stability region. Eigenvalues are scaled with h for ease of visualisation.

Finally, we take a more detailed look at the relationship between the spectral radius $\rho(\boldsymbol{G}_h(\gamma(c)))$ and c shown in Figure 2 for different hyperviscosity orders α and internodal distances h. At this point we focus on the solid lines, while the dotted lines show the reduced monomial order results discussed in subsection 3.3. The stabilising effect of hyperviscosity appears to be well behaved for the pure advection case i.e. once c is large enough to remove spurious eigenvalues the system remains stable with $\rho \lesssim 1$. Problems only occur in the $\alpha = 4$ case with ρ diverging as we increase c. The fact that this is exacerbated by decreasing h indicates that the issue might be caused by increasingly computationally ill-conditioned approximation of the high order derivatives. Decreasing h also increases the required c to reach $\rho \lesssim 1$, which is somewhat unexpected since h scaling is already included in $\gamma(c)$. Further examination is required to determine whether this is caused by the stabilising effect of additional numerical diffusion inherent to larger internodal distances and whether additional scaling for γ would be beneficial.

3.1.1. How to determine the parameter c in practice

The system is stable if $\rho(\boldsymbol{G}_h(\gamma(c))) \leq 1$. We aim to introduce the least amount of numerical diffusion while maintaining stability. This implies that

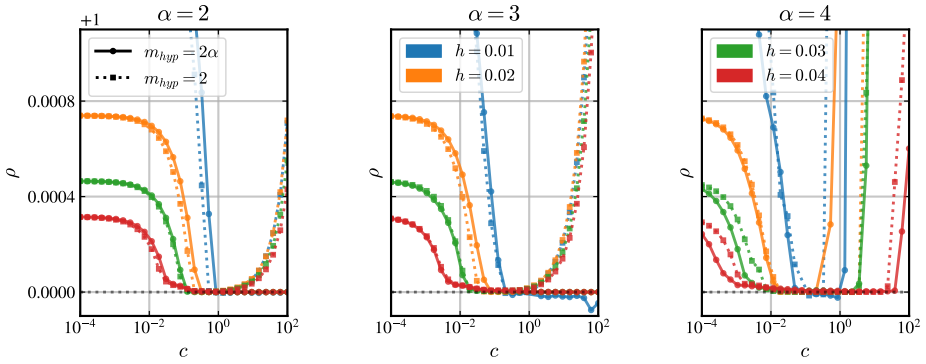

Figure 2: Spectral radius ρ of the evolution matrix $\boldsymbol{G}_h$ (19) at $\Delta t = 10^{-4}$ with respect to $c \in [10^{-4}, 10^2]$ for internodal distances $h \in \{0.01, 0.02, 0.03, 0.04\}$ and orders of hyperviscosity $\alpha \in \{2, 3, 4\}$. The solid line shows ρ calculated with the full order monomial augmentation as suggested for the RBF-FD approximation while the dotted line displays effects of using a reduced monomial order of $m = 2$ with stencil size $n = 30$.

we are searching for the lowest c (denoted as c_{opt}) that produces a stable system

$$c = \min\{c \geq 0 \, : \, \rho(\boldsymbol{G}_h(\gamma(c))) \leq 1\}. \tag{24}$$

Therefore, in the most simplified form, we are solving $\rho(\boldsymbol{G}_h(\gamma(c))) - 1 = 0$.

To determine c_{opt}, one must solve equation (24). The simplest approach is to use the bisection method in log-scale, with the lower bound selected so that $\rho(\boldsymbol{G}_h(\gamma(c_{lo}))) > 1$ and $\rho(\boldsymbol{G}_h(\gamma(c_{hi}))) \leq 1$ for the upper bound. The bounds were determined with a coarse sweep around an initial guess but the selection is not trivial in practice for cases with narrow stability regions requiring future work to devise a more optimal method. The most computationally intense part of the algorithm is the eigendecomposition of the evolution matrix. Since we are only interested in computing the eigenvalue with the largest magnitude, the Implicitly Restarted Arnoldi Method (IRAM) algorithm [48] is employed to improve efficiency but the computational cost of evaluating ρ remains high. We use the iterative algorithm to calculate the 40 largest eigenvalues by magnitude to be reasonably certain that the maximal value was captured since the algorithm does not guarantee convergence to the extremal (largest or smallest) eigenvalue.

3.2. Consistency and order of hyperviscosity

Next, we would like to investigate the consistency of the scheme. That is, our approximation of the solution must converge to the correct solution. For simplicity, again assume that $\mathcal{N}$ is a linear operator. Let u be sufficiently regular and solve the posed problem, that is

$$\partial_t u(\boldsymbol{x}, t) + \mathcal{N}(u(\boldsymbol{x}, t)) = 0. \tag{25}$$

Then it also holds for many classical PDEs that as $h \to 0$ the family of solutions $\{u_\gamma\}$ satisfying (13) converges strongly $u_\gamma \to u$ since $\gamma = ch^{2\alpha}$. This result extends the classical result (for artificial viscosity), see for example [49, 50]. Furthermore, it ensures the consistency of stabilisation in a continuous sense. To preserve the order of approximation of the discrete scheme, the stabilisation term must be consistent and vanish faster than the expected convergence rate. Assume that our base convergence rate is $\mathcal{O}(h^p)$, then the hyperviscosity term must vanish with at least the same order to preserve the order of accuracy. To study consistency, we introduce the local interpolation error estimate on a Voronoi region K_i around $\boldsymbol{x}_i \in X$. Fix $t \in [0, T]$ and define $u(\cdot) = u(\cdot, t)$. We define an interpolation operator that maps our functions to the trial space $V_h = \operatorname{span}\{\Psi_1, \dots, \Psi_N\}$ by $I_h : C(\Omega) \to V_h$ and $I_h u = \sum_{i=1}^N \Psi_i u(\boldsymbol{x}_i)$. Consider a polyharmonic spline approximation of order $k > 2\alpha$, augmented with monomials of degree m as explained in section 2. Provided that the interpolant exists, we have the following error estimate for sufficiently small h and $u \in W_\infty^{m+1}(\Omega) \cap C^{2\alpha}(\Omega)$:

$$\|\Delta^\alpha (I_h u - u)\|_{L^\infty(K_i)} \leq C_i h^{m+1-2\alpha} \left|u\right|_{W_\infty^{m+1}(K_i)}. \tag{26}$$

For the $m \geq 2\alpha$ case, this is a well-known result [26]. If $u \in W_\infty^{2\alpha+1}(\Omega) \cap C^{2\alpha}(\Omega)$, then the above also holds for $m < 2\alpha$ as proven in the Appendix Appendix A.

Since we would like to study consistency in the $\|\cdot\|_h$ norm, we must relate it to the $\|\cdot\|_{L^\infty(\Omega)}$ norm. This is simple for $w : \Omega \to \mathbb{R}$, for which the essential supremum is equal to the supremum:

$$\|w\|_h^2 = \frac{|\Omega|}{N} \sum_{i=1}^N w(x_i)^2 \leq \frac{|\Omega|}{N} N \max_{1 \leq i \leq N} w(x_i)^2 = |\Omega| \|w\|_\infty^2 \leq C \|w\|_{L^\infty(\Omega)}^2. \tag{27}$$

Our local estimates $w = \Delta^\alpha (I_h u - u)$ are only defined on $\overline{K_i}$ but the measure of Voronoi region boundaries is 0 and therefore $\|w\|_{L^\infty(\Omega)} = \max_{1 \leq i \leq N} \|w\|_{L^\infty\left(\overline{K_i}\right)}$.

Since w is continuous, we have pointwise control via L^∞ norm inside each Voronoi region K_i. Hence we again see that $\|w\|_h \leq C\|w\|_{L^\infty(\Omega)}$ holds. This now allows us to estimate the norm of the stabilisation term at time t. Let u solve (13) and $u_h \in V_h$ be its stable numerical approximation of order h^p, furthermore, assume that $m \geq p-1$. We infer by triangle inequality that

$$\begin{aligned}\|\gamma\Delta^\alpha u_h\|_h &= \gamma\|\Delta^\alpha u_h + \Delta^\alpha u - \Delta^\alpha u + \Delta^\alpha I_h u - \Delta^\alpha I_h u\|_h \\ &\leq ch^{2\alpha}(\|\Delta^\alpha u_h - \Delta^\alpha I_h u\|_h + \|\Delta^\alpha I_h u - \Delta^\alpha u\|_h + \|\Delta^\alpha u\|_h).\end{aligned} \tag{28}$$

We can bound the first term as

$$\|\Delta^\alpha u_h - \Delta^\alpha I_h u\|_h = \|\Delta^\alpha(u_h - I_h u)\|_h \leq Ch^{-2\alpha}\|u_h - I_h u\|_h, \tag{29}$$

where an inverse type-inequality was used [51, Lemma 4.5.3], and the second according to inequalities (27, 26). Furthermore we note that by the stability assumption and timestepping consistency we have $\|u_h - I_h u\|_h \leq Ch^p$. This yields

$$\begin{aligned}\|\gamma\Delta^\alpha u_h\|_h &\leq ch^{2\alpha}(C_1 h^{p-2\alpha} + C_2 h^{m+1-2\alpha} + C_3) \\ &\leq (C_1 h^p + C_2 h^{m+1} + C_3 h^{2\alpha}).\end{aligned} \tag{30}$$

Note that the $h^{2\alpha}$ scaling for γ cancels out the $ch^{-2\alpha}$ term in operator approximation error estimate (26), thus removing the hard limit for minimal monomial order. For sufficiently large m and p, the term $h^{2\alpha}$ dominates, which means that our solution can only preserve accuracy up to 2α order. For the p-order accurate scheme, we must select $\alpha \geq p/2$ and $m \geq p-1$.

For a second-order approximation, we expect that all orders of hyperviscosity larger than $\alpha \geq 2$ will produce similar results, which we confirm on Figure 3. Here, we also note that the approximation of higher-order derivatives becomes increasingly computationally ill-conditioned, since the condition numbers of the differentiation matrix $\boldsymbol{D}_h^{hip}$ scale as $h^{-2\alpha}$, hence, lower orders are typically prefered.

3.3. Parametrisation of the RBF-FD approximation

One of the problems with hyperviscosity schemes is their large computational complexity. Typically, when approximating operators of order k, a monomial order of at least k is required for the first-order operator approximation, i.e., $m = 2\alpha$ is typically used for hyperviscosity. Since the order of the monomial term is inherently connected to the stencil size and subsequently the number of non-zero row entries in the matrix, it has a significant

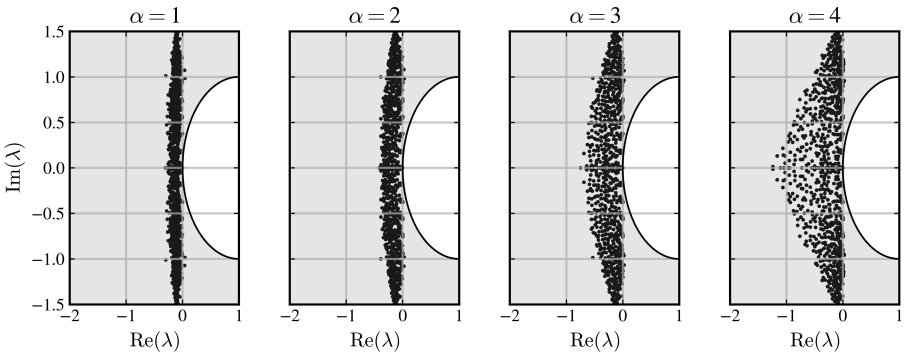

Figure 3: Eigenvalue spectra of the stabilised advection operator $\hat{\boldsymbol{D}}_h$ (15) using $N \approx 10^3$ with respect to different hyperviscosity orders α using c_{opt}. The shaded region in the figure is the implicit Euler stability region. Eigenvalues are scaled with h for ease of visualisation.

impact on the time complexity of the algorithms for both solving the system and finding the maximal eigenvalues.

In the previous section, we have shown that PHS alone can be used to control the consistency of the hyperviscosity operator, assuming that the local interpolation systems are unisolvable. This suggests that we can approximate the hyperviscosity operator with much lower monomial orders than otherwise warranted by the high-order derivatives, preserving sparsity and improving computational performance. We also believe that in practice, the bounds that are obtained from inverse inequality (A.1) are not strict as argued in subsection 3.2. The veracity of this claim and the impact of reduced order approximation for hyperviscosity is first examined in Figure 2, where the dotted lines show results with hyperviscosity operators approximated with 2nd order monomials and stencil size $n = 30$. In contrast to the full order results, displayed with solid lines, the system now becomes unstable again as c increases. Reduced order results closely match the full order results where ρ first dips below 1 with increasing c, which is the region we aim for with the bisection algorithm. Increased stencil size widens the range of c where $\rho \lesssim 1$ and can prevent successful stabilisation if chosen too small. The interplay between the reduced monomial order and stencil size is further analysed in subsection 4.1.

It is also essential to select the correct polyharmonic spline order, k, to ensure the desired stability properties of the semi-discrete system. The authors of [35] proposed a formula linking the monomial order m of the approxima-

tion to the polyharmonic spline order via $k = 2m+1$. This choice is justified by the fact that the system (2) is guaranteed to be solvable when this polyharmonic spline order is used [42]. However, in practice, the system tends to be solvable regardless of this criterion. Specifically, from our experiments, we observed no significant relationship between the monomial order and the required spline order, as demonstrated in Figure 4, where eigenvalue spectra for the advection operator are shown for various combinations of k and augmentation orders. Moreover, Figure 4 reveals that increasing k leads to greater instability. This suggests that the lowest value of k that still provides the required regularity should be used, i.e. for the advection operator this corresponds to $k = 3$. Nonetheless, when approximating hyperviscosity, it is crucial to ensure that the local interpolation system possesses sufficient regularity to compute the 2α-order derivatives. This requirement implies $k = 2\alpha + 1$.

Based on this discussion we propose using different approximations for constructing the advection and hyperviscosity operators. Reduced order approximation is used for hyperviscosity term since there appears to be no clear downside. Unless otherwise specified we use $k = 3$ and $m = 2$ with the suggested stencil size $n = 12$ for the advection operator, and $k = 2\alpha + 1$ and $m = 2$ with stencil size $n = 30$ for the hyperviscosity operator.

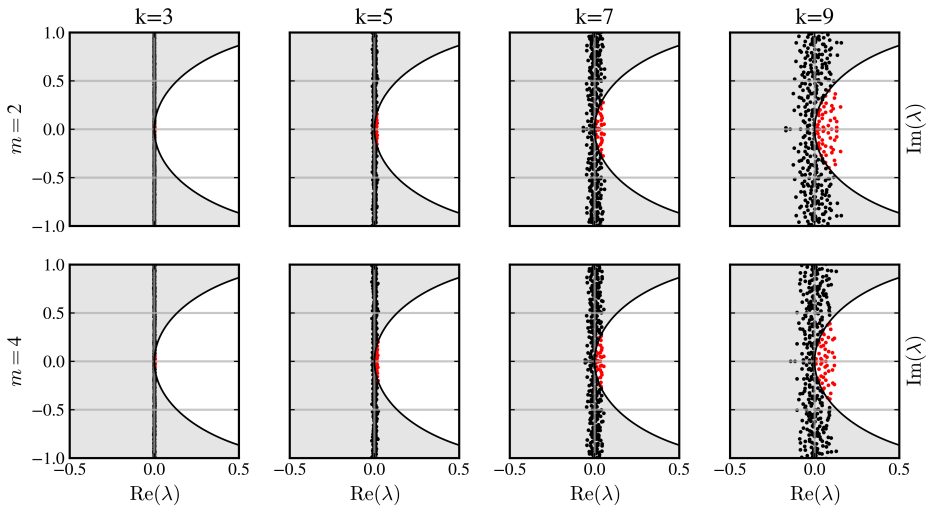

Figure 4: Eigenvalue spectra of the advection operator $\boldsymbol{D}_h$ for different orders of polyharmonic splines k at $h = 0.03$. The eigenvalues are scaled by $\lambda = h\tilde{\lambda}$. The shaded region represents the implicit Euler stability region.

4. Numerical results

In this section, we investigate the impact of hyperviscosity on two unstable problems:

1. Linear advection equation, where a $C^\infty(\Omega)$ initial condition is linearly advected under constant velocity field on $\Omega = \mathbb{T}^2$,
2. Burgers' equation, where $C^\infty(\Omega)$ initial conditions with simple Dirichlet boundary conditions are non-linearly advected.

These test cases have already been extensively used in a stabilisation research setting. A linear advection equation has also been studied in [38, 34]. However, our results focus on the influence of the c term on the hyperviscosity stabilisation. Throughout this section, we use the following error norm

$$\|e\|_2 := \frac{\|\boldsymbol{u} - \boldsymbol{u_h}\|_2}{\|\boldsymbol{u}\|_2}, \tag{31}$$

where $\boldsymbol{u}$ is a vector of analytical solution values sampled at nodes X. To further show the effects of hyperviscosity, we introduce the relative energy

$$\text{Relative energy} := \frac{\|u_h\|_h^2}{\|u\|_h^2}. \tag{32}$$

The set of computational nodes X is generated using the dedicated meshless node positioning algorithm described in [4] with a constant density throughout the computational domain parametrised with the internodal distance h.

4.1. Linear advection

In this section, we consider the classical 2D pure linear advection equation on a periodic domain $\Omega = \mathbb{T}^2([0, 1])$

$$\partial_t u(\boldsymbol{x}, t) + \boldsymbol{\beta} \cdot \nabla u(\boldsymbol{x}, t) = 0, \tag{33}$$

where we set $\boldsymbol{\beta} = (1, 0)$. To avoid any issues with the regularity of u, we consider the following initial conditions

$$u(\boldsymbol{x}, 0) = \begin{cases} \exp\left(\frac{1-R^2(r^2-R^2)}{r^2-R^2}\right), & r < R \\ 0, & r \geq R. \end{cases}, \tag{34}$$

where $R = 0.1$ and r is the distance to the centre of the domain Ω. In the fully discrete case we consider

$$\boldsymbol{u_h}(t_{n+1}) - \boldsymbol{u_h}(t_n) = \Delta t \, \left(-\boldsymbol{D}_h^{\partial_x} \boldsymbol{u_h}(t_{n+1}) + (-1)^{\alpha+1} \gamma \boldsymbol{D}_h^{hip} \boldsymbol{u_h}(t_{n+1})\right) \quad (35)$$

The regularity of $u(\boldsymbol{x}, 0) \in C^\infty(\Omega)$ allows the application of fourth-order hyperviscosity. The monomial order for approximating the derivatives is set to $m = 2$. The error norms are measured by comparing the solutions with the initial solution $u(\boldsymbol{x}, 0)$, since $u(x, t) = u(x - t, 0)$, at time $t = 1, 2, \ldots, T$. The objective of this section is to observe how the stabilised RBF-FD method performs on a pure linear advection.

We first visually examine the solution. In Figure 5, we compare the hyperviscosity stabilised solution to its non-stabilised counterpart. The stabilised solution uses hyperviscosity of order $\alpha = 2$ with $c_{opt} = 0.32$ found by the algorithm presented in subsection 3.1.

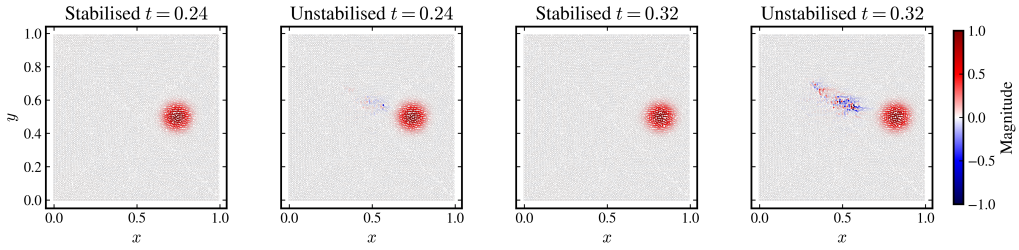

Figure 5: Solution of the linear advection equation at two different times $t \in \{0.24, 0.32\}$ with $h = 0.01$. The stabilised solution (left) admits no visible artefacts, compared to the non-stabilised solution (right) where major instabilities can be observed.

After a relatively short time $t = 0.24$, visible numerical artefacts begin to appear in the non-stabilised solution, whereas no visible artefacts are present in the stabilised solution. Once present, the erroneous field values quickly grow in magnitude, surpassing the advected solution at $t = 0.32$ and diverging into invalid numerical values shortly afterwards.

Before proceeding, we consider the convergence with respect to timestep Δt for different values of h shown in Figure 6. The diffusivity caused by the implicit Euler scheme vanishes with $\Delta t \to 0$ and the relative energy, displayed on the left graph, converges towards 1 indicating energy conservation. This

is an encouraging result, since, in general, this is not guaranteed by the stability criterion. On the right graph, showing the relative error, we can observe the convergence behaviour $\mathcal{O}(\Delta t)$. For $\Delta t > 0.01$ the spatial error is dominated by the implicit Euler diffusion, whereas for $\Delta t < 0.01$ we obtain the expected convergence rate of around 1 before plateauing for a given h. Based on this result we use $\Delta t = 10^{-4}$ for following computations unless otherwise specified.

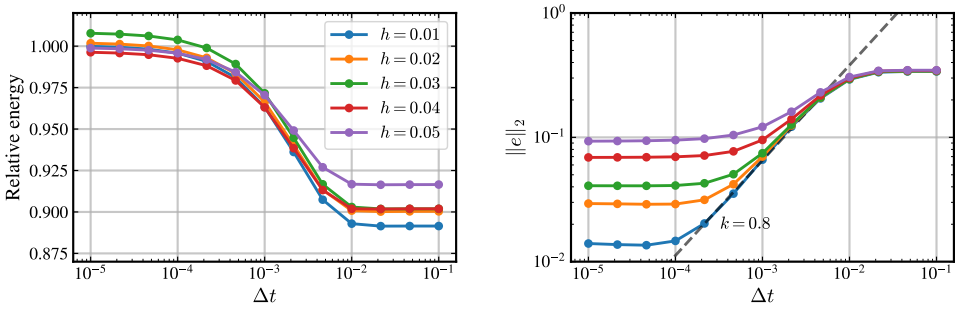

Figure 6: Relative energy and error of numerical simulation of the pure advection with respect to Δt at $t = 10$. Stabilised with hyperviscosity of order $\alpha = 2$ with a fixed $c = 1$ for clearer comparison. Different lines correspond to different spatial discretisation parameters h. The convergence order is fitted based on the middle 5 markers with $\Delta t^k + d$.

The impact of Δt is further scrutinized in Figure 7 where we look at its influence on the relationship between the relative error and c. The region of stable c shrinks as $\Delta t \to 0$ and less diffusivity is introduced by integration. The relative error is dictated by the 2nd order approximation used for the advection operator and remains similar across different orders of hyperviscosity. This provides us with the first indication that the solution is not disturbed by the stabilising term.

We now discuss the optimal selection of c. Although $c_{opt} = 0.32$ selected by the algorithm in Figure 5 noticeably stabilises the solution, the optimality of this c remains in question. To shed some light on this issue, we analyse the relative energy of the solution after 1, 3, and 10 rotations, corresponding to $t \in \{1, 3, 10\}$, for different orders of hyperviscosity α. The results of this analysis, performed with $h = 0.02$ ($N \approx 2100$), are presented in Figure 8. First, we observe that for each setup, there is a range of suitable c values where the solution remains stable with minimal dissipation. This region narrows as smaller and smaller instabilities become apparent with passing

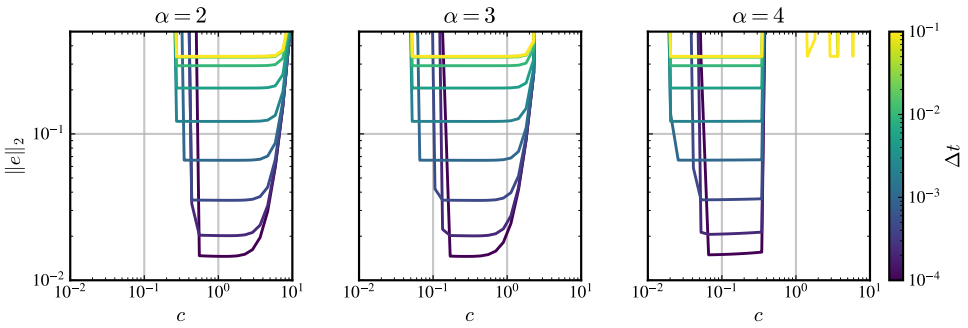

Figure 7: Relative error of pure advection equation at $t = 10$ for $h = 0.01$. Different lines correspond to different timestep Δt ranging from 0.1 to 10^{-4}. Different columns correspond to different orders of hyperviscosity.

time and cause solutions to diverge. Most importantly, the c_{opt} (denoted by the vertical dashed line) falls within the stable range in all cases, which is exactly what we hoped for. We note that the $\alpha = 1$ case, analogous to normal diffusion, exhibits significantly stronger drop in relative energy with increasing c when compared to higher orders of Laplacian operator.

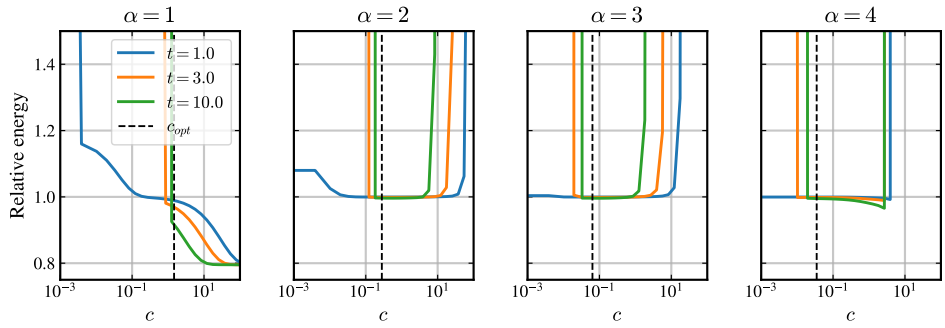

Figure 8: Relative energy of the hyperviscosity stabilized solution discretised with $h = 0.02$ with respect to c at $t \in [1, 3, 10]$ for different orders of hyperviscosity.

To verify that the hyperviscosity operator does not interfere with convergence rates of the scheme, we consider convergence under h-refinement for different orders of monomial augmentation m used in approximating the advection term. In Figure 9 we observe that the convergence rates stay relatively stable across different orders of hyperviscosity. In the last two markers for the high order approximations, the effects of timestepping error already dominate over the spatial error. For $m \in \{2, 4\}$, we obtain 1 order of conver-

gence less than is typically expected. The observed effect is similar to [52]. The overall convergence rates agree with the observed convergence rates in the RBF-FD setting, most importantly, the diffusion caused by hyperviscosity does not affect the convergence rate.

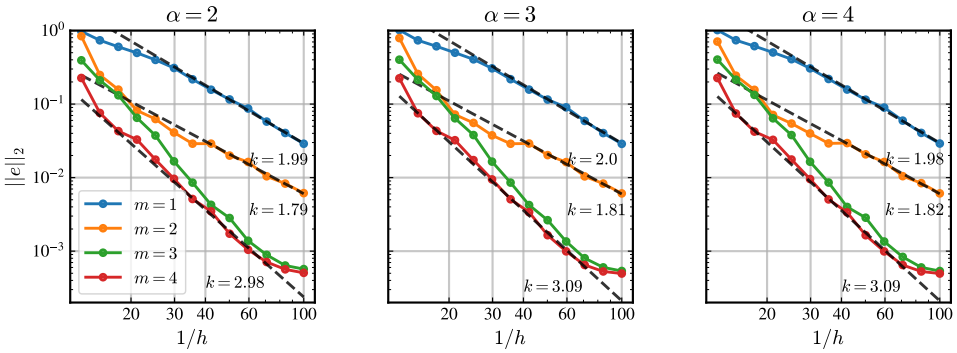

Figure 9: Error convergence under h-refinement for linear advection equation at $t = 5$ with $\Delta t = 10^{-5}$. Different lines represent different orders of monomial augmentation, while columns represent the order of the hyperviscosity operator. The black lines are fitted based on the ending 8 markers without the last 2 markers with $h^k + d$.

Next, we take a look at how the stencil size and different orders of polynomial augmentation for the hyperviscosity operator affect the error of the solution. This is shown in Figure 10. We notice that the results are in strong agreement with the theory from subsection 2.1 and subsection 3.2. We are able to successfully stabilise advection with augmentation orders that are much lower than the approximated derivative orders as indicated by the hyperviscosity consistency estimate (30). This comes at the cost of increased stencil size though. Stable solutions require stencil size that is roughly in line with the suggested stencil size for $m = \lceil \frac{k}{2} \rceil - 1$ required to prove unisolvency for PHS with $k = 2\alpha + 1$. Even with this increased stencil size requirement there are still significant computational benefits. In $\alpha = 4$ case we require $n \approx 30$ instead of $n \approx 90$ that is suggested for $m = 8$. The error with $m = 1$ tends to dominate the overall error, which is also consistent with the theory, since for that case $m = p - 1$ and the $\mathcal{O}(h^{m+1})$ term in our hyperviscosity consistency estimate becomes relevant.

At last, we take a look at the consistency of the hyperviscosity operator by varying the monomial order in the approximation of the hyperviscosity operator. According to Figure 10, we use stencil size $n = 35$ to be able to

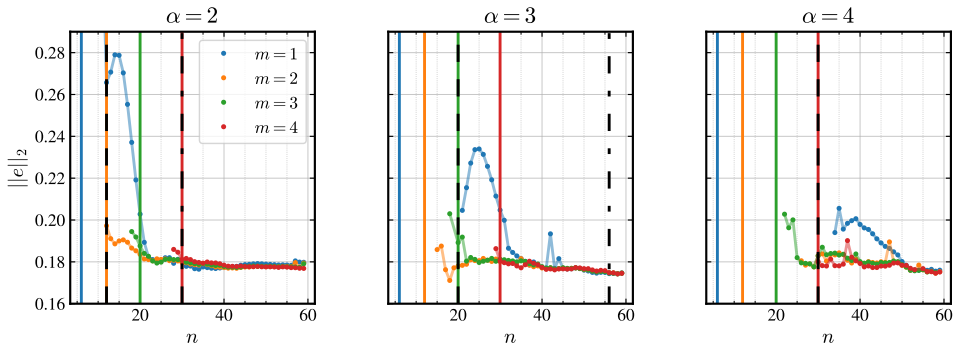

Figure 10: Error of pure advection numerical simulation with respect to stencil size of the hyperviscosity operator at $t = 10$ and $h = 0.02$. Different colours correspond to different orders of hyperviscosity monomial augmentation m. Coloured vertical lines display the recommended [41] stencil size for that augmentation order. Black vertical lines show the recommended stencil sizes based on α. Dash-dotted for the order appropriate $m = 2\alpha$ and dashed for $m = \lceil \frac{k}{2} \rceil - 1$, with $k = 2\alpha + 1$ based on unisolvency bounds.

fairly compare all of the considered orders. We set $\Delta t = 10^{-5}$ and perform a similar experiment to the above. In Figure 11 we can see that as $h \to 0$ the errors for different approximations (monomial orders) of the hyperviscosity operator stay relatively similar, i.e. the error of approximation stays consistent regardless of the order of monomial augmentation. The only apparent outlier is the severely reduced order $m = 1$ approximation for $\alpha = 4$.

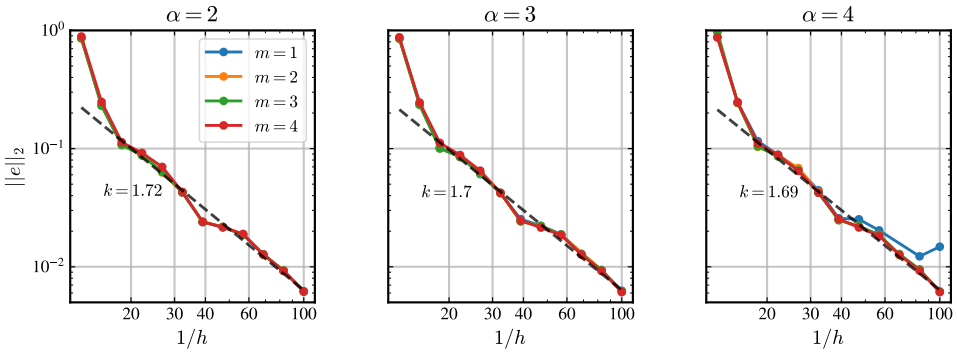

Figure 11: Error of pure advection numerical simulation with respect to $1/h$ at $t = 5$. Different lines correspond to different orders of hyperviscosity monomial augmentation m.

4.2. Non-linear advection: Burgers' equation

In this section, we consider a 2D non-linear advection-diffusion equation, the viscid Burgers' equation

$$\partial_t \boldsymbol{u}(\boldsymbol{x}, t) + \boldsymbol{u}(\boldsymbol{x}) \cdot \nabla \boldsymbol{u}(\boldsymbol{x}, t) = \mathrm{Re}^{-1} \Delta \boldsymbol{u}, \tag{36}$$

where $\boldsymbol{u}$, Re stand for velocity and the Reynold's number, respectively. The problem is considered on a unit square $\Omega = [0, 1] \times [0, 1]$ with Dirichlet boundary conditions taken from closed form solution $\boldsymbol{f} : \Omega \times [0, T] \to \mathbb{R}^2$ from [39, 40],

$$\boldsymbol{f}(\boldsymbol{x}, t) = \begin{bmatrix} \frac{3}{4} + \frac{1}{(4(1+\exp \frac{\mathrm{Re}}{32}(-t-4x_1+4x_2)))} \\ \frac{3}{4} - \frac{1}{(4(1+\exp \frac{\mathrm{Re}}{32}(-t-4x_1+4x_2)))} \end{bmatrix}. \tag{37}$$

Again, we solve the problem using implicit time stepping, where the problem is numerically linearised by using the previous time step velocity in the advection term

$$\begin{aligned} \boldsymbol{u_h}(t_{n+1}) - \boldsymbol{u_h}(t_n) = &\Delta t(-\boldsymbol{u_h}(t_n) \cdot \boldsymbol{D}_h^{\nabla} \boldsymbol{u_h}(t_{n+1}) + \mathrm{Re}^{-1} \boldsymbol{D}_h^{\Delta} \boldsymbol{u_h}(t_{n+1}) \\ &+ (-1)^{\alpha+1} \gamma \boldsymbol{D}_h^{hip} \boldsymbol{u_h}(t_{n+1})), \end{aligned} \tag{38}$$

The linearisation of the advection term causes the matrix $\boldsymbol{G}_h$ to become time-dependent; i.e., c is also time-dependent, and it should therefore be recomputed multiple times during the simulation. We also note that this is not a linear Cauchy problem, and the Lax-stability is not a correct notion of stability for this problem. However, we assume that boundary conditions and nonlinearity do not play a crucial role and still use it as a valid criterion. The equation becomes more unstable with increasing Re, as advection begins to dominate over diffusion. Moreover, the solution exhibits increasingly pronounced shock-like fronts with thin boundary layers near the shocks, where the Gibbs phenomenon may occur. Here, we aim to demonstrate that in the moderately diffusive regime, a stable solution can be obtained using hyperviscous RBF-FD, whereas the non-stabilised RBF-FD diverges.

In the following discussions, if not specifically stated otherwise, we recompute c_{opt} every $\chi = 10$ time-steps.

To begin with, we visually compare the stabilised and non-stabilised solutions of the Burgers' equation in Figure 12. The parameters of the scheme are $h = 0.01$ and $\mathrm{Re} = 5000$. Although the initial conditions are smooth and

the effects of non-linear advection are still relatively small, the non-stabilised solution exhibits considerable numerical artefacts arising from the interplay of the Gibbs phenomenon, advection operator instability, and the inherent characteristics of RBF-FD [25], whereas these artefacts are less present in the stabilised solution.

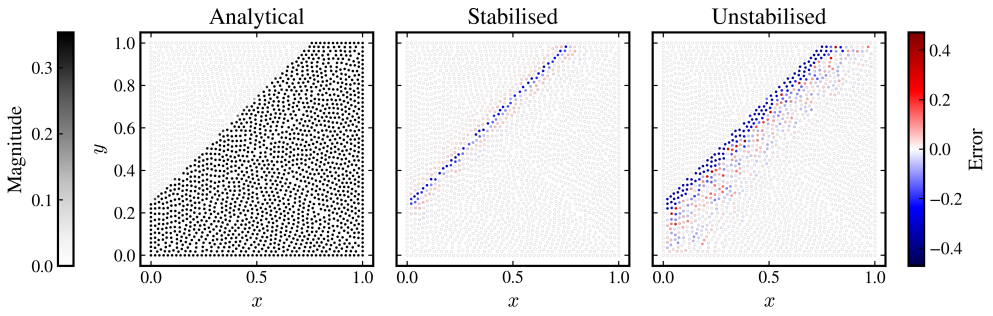

Figure 12: Comparison between stabilised (centre) and non-stabilised (right) Burgers' equation at $t = 1$. The error is measured using a pointwise ℓ_1 norm. The simulation was performed at $\mathrm{Re} = 5000$ with $h = 0.01$ ($N \approx 10^4$) with $\Delta t = 0.01$.

Akin to the previous section, we begin with the timestep analysis. We again consider the error with respect to c for different Δt. We choose an unstable case of $\mathrm{Re} = 10000$, where shocks already form. The results are shown in Figure 13. This time, the stability area does not shrink as $\Delta t \to 0$, however, it does shrink with the hyperviscosity order α. The order $\alpha = 2$ results in the lowest error and the largest stability area, which is consistent with the fact that shocks need lower-order smoothing. This prompts us to use the order $\alpha = 2$ for the remaining results. We also observe that there exists a c where the error achieves its minimum and this is no longer the lowest stable c that we search for with our algorithm.

In Figure 14 we examine the relation between the relative energy and c for different Reynolds numbers at different times of the simulation. Again, we see that there is an interval of suitable c values where the solution remains stable with minimal dissipation. Note that in contrast with the linear advection case shown in Figure 8 the smallest stable c no longer leads to an optimal result in terms of the relative energy conservation. This loss is caused by insufficient damping of the numerical artefacts that appear when evolving

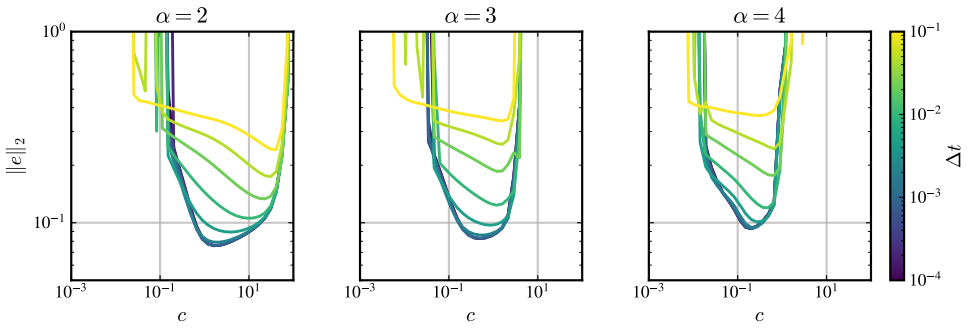

Figure 13: Error for the case of Burgers' equation at $t = 1$ for $h = 0.02$. Different lines correspond to different timesteps Δt ranging from 0.1 to 10^{-4}. Different columns correspond to different orders of hyperviscosity.

the sharp initial condition. This is most apparent in the intermediate range of Re where the initial condition is sharp enough to cause significant artefacts, while the case is still sufficiently mild to not outright diverge.

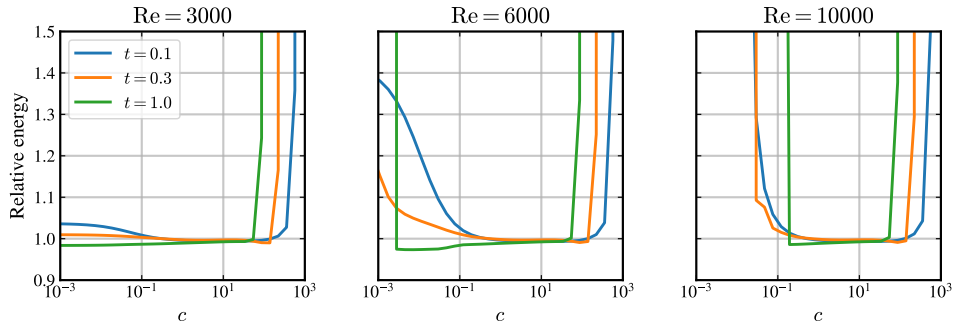

Figure 14: Relative energy with respect to c at $h = 0.02$ and $\Delta t = 0.01$. Different lines represent sampling at different times of the simulation. The columns show simulation at different Reynolds numbers.

As the linearisation of the advection term causes the matrix $\boldsymbol{G}_h(t_n)$ to become time-dependent, a natural question arises regarding the required frequency of c_{opt} re-computation. We can assume that the number of times that c_{opt} is recomputed is proportional to the system's dynamics, i.e. the higher the Reynolds number, the more frequent c_{opt} recomputation is required for optimal stabilisation. This is demonstrated in Figure 15, where one can see that for Reynolds numbers below 5000, the result quality is weakly coupled with the re-computation frequency and remains similar to the non-stabilised

case. This highlights one of the downsides of the algorithm as the cases that do not exhibit exponential growth in their eigenvalues remain un-stabilised even in the regime where a moderate stabilisation displayed with the dash-dotted line would prove beneficial. However, with increasing Reynolds number, the effect of more frequent c_{opt} re-computation becomes evident. Times in the top row of Figure 15 were selected to show the behaviour 5 iterations after the first re-compute for different frequencies while the bottom row displays the rest of the evolution to $t = 1$. We can see that the error of the solution with the newly re-computed c_{opt} clearly drops from the overlap of others that are still stabilised with the original value determined at the first step. Note that the benefits of the more frequent re-computation are still present at $t = 1$ after many lower frequency re-computations indicating that a sufficient re-computation frequency in the initial stages of the evolution is critical.

Finally, in Figure 16, the convergence of the presented solution is shown. We get the expected 2nd order of convergence for low Reynolds numbers $\mathrm{Re} \leq 3000$ and weak convergence above that is caused by the aforementioned numerical artefacts degrading the increasingly sharp initial wave.

Note on implementation

The numerical procedure is implemented in Medusa C++ library [53]. For the computation of the eigenvalues, we employ the Krylov subspace algorithms [48] offered by the Spectra C++ library [54] for sparse and the algorithms based on Schur decomposition provided within the function *eigs* in the Eigen C++ library [55] for dense matrices. The code was compiled using g++ (GCC) 11.3.0 for Linux with -O3 -DNDEBUG flags on AMD EPYC 7702 64-Core Processor computer. OpenMP API has been used to run parts of the solution procedure in parallel on shared memory. Post-processing was done using Python 3.10.6.

5. Final remarks

This paper discusses the hyperviscosity stabilisation scheme in the RBF-FD context from various points of view. The presented analysis was primarily focused on the computation of adherent hyperviscosity constant c (or γ) to satisfy the Lax equivalence theorem. In particular, we constructed an algorithm that uses the maximal eigenvalue magnitude of the evolution matrix to

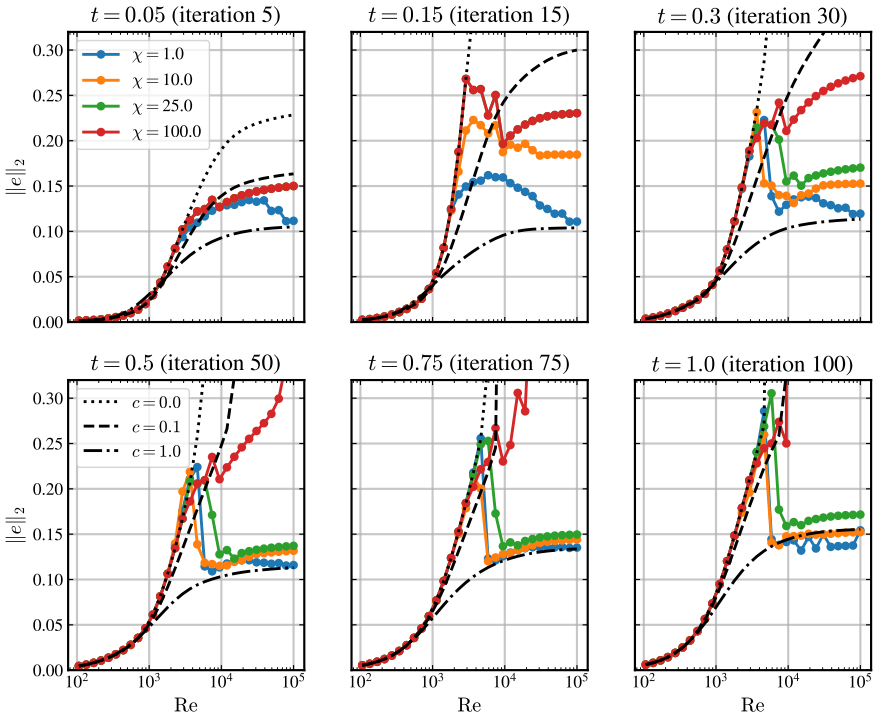

Figure 15: Time evolution of relative error of Burgers' equation at $\Delta t = 0.01$ and $h = 0.02$ with respect to Reynolds number (Re). Different colours of lines correspond to c_{opt} at different recomputation frequency χ. The black lines correspond to statically chosen hyperviscosity parameters.

iteratively obtain the constant that produces a Lax-stable scheme in the sense that all eigenvalues of the evolution matrix lie inside the stability region.

Besides the parameter c, which effectively governs the magnitude of hyperviscosity, another important parameter, the order of hyperviscosity α, determines the vanishing order. As α increases, the order of the derivatives involved increases as 2α. Even for low values of α, this becomes computationally expensive due to the large stencil requirements. To mitigate this issue, we considered the consistency estimate for the hyperviscosity operator. Specifically, we demonstrated that any monomial order can be used in the RBF-FD approximation of the hyperviscosity operator to produce a consis-

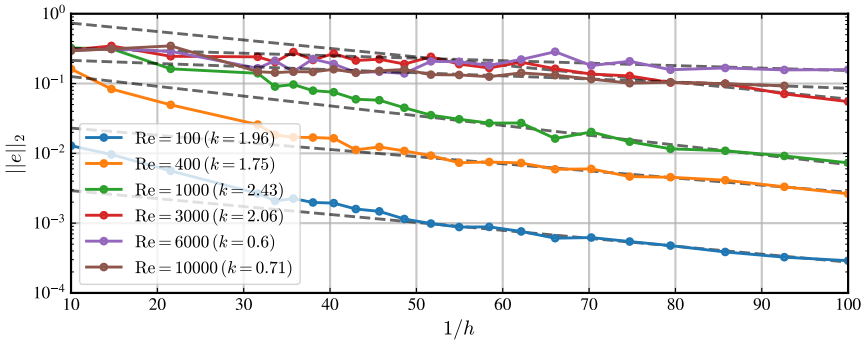

Figure 16: Convergence of the approximation error as a function of $1/h$ for different Reynolds numbers (Re) with $\Delta t = \max\{0.01h, 0.001\}$. The error is captured at $t = 1$ with c_{opt} recomputed every 10 iterations. The lines are fitted based on the last 10 markers with $h^k + d$.

tent scheme. This, in turn, allows for the use of smaller stencil sizes in the RBF-FD approximation.

In terms of the RBF-FD parametrisation, we showed that increasing the polyharmonic spline order k also increases the instability of the system. On the other hand, when approximating hyperviscosity, sufficient interpolation regularity is required to compute the 2α-order derivatives, specifically, $k = 2\alpha + 1$. Therefore, we propose using different polyharmonic spline orders for constructing the advection and hyperviscosity operators, namely, $k = 3$ for the advection operator and $k = 2\alpha + 1$ for the hyperviscosity operator.

In the numerical demonstration, we first tackled a linear advection problem. Although the case is relatively simple, its numerical solution exhibits instability, making it an ideal case for studying stabilisation schemes. We initially demonstrated that the proposed scheme provides significantly more stable results compared to the non-stabilised RBF-FD counterpart. We also investigated the interplay between α and c, clearly showing that there exists a range of c values for which the solution remains stable while introducing relatively low dissipation. Moreover, the presented algorithm successfully identifies c_{opt} within this range. The advection case is also used to demonstrate the consistency of the hyperviscosity operator and to verify the theoretical result regarding the use of low monomial orders for high order hyperviscosity.

To extend the analysis, we addressed a more complex non-linear case, namely the Burgers' problem. First, we repeated similar analyses as in the linear advection case to confirm the adequacy of the proposed solution proce-

dure. However, the main focus of the analysis in this case is the non-linearity of the problem, which makes the parameter c_{opt} time-dependent. We demonstrated how c_{opt} evolves throughout the simulation and to what extent the frequency of its re-computation affects the simulation results.

To sum up, this paper adds another piece to the understanding and development of general hyperviscosity stabilisation. It discusses parameter selection and their influence on the numerical simulation, as well, the core concepts of hyperviscosity itself. Although this paper pushes the hyperviscosity stabilisation one step closer to the practical implementation, there are still several unanswered questions and open topics for future work. Starting with the c selection algorithm, more efficient techniques for determining the largest eigenvalues should be investigated, and the bisection algorithm for root finding could be replaced with a more sophisticated approach to reduce the required number of eigenvalue re-computations. In future work we will also consider recently provided energy method framework [25] to potentially determine c more effectively.

In time-dependent problems, an adaptive c recomputation could be devised that would follow the dynamics of the solution and recompute c accordingly. For example, in highly unstable cases, it might be practical to periodically compute the eigenvalue with the largest magnitude and then assess whether to reapply the algorithm or not.

Acknowledgements

The authors would like to acknowledge the financial support from the Slovenian Research and Innovation Agency (ARIS) in the framework of the research core funding No. P2-0095 and research projects No. J2-3048 and N2-0275, as well as The Young Researcher programmes PR-12347 and PR-10468. Funded by National Science Centre, Poland under the OPUS call in the Weave programme 2021/43/I/ST3/00228. This research was funded in whole or in part by National Science Centre (2021/43/I/ST3/00228). For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

Appendix A. Proof of error estimate for $m < 2\alpha$

For convenience let us concisely repeat the proposition from the main text:

Proposition 1. *Let $u \in W_\infty^{2\alpha+1}(\Omega) \cap C^{2\alpha}(\Omega)$ and assume its polyharmonic spline interpolant $I_h u$ exists. Additionally assume that the order of the polyharmonic splines is $k > 2\alpha$ and the approximation is augmented with monomials of degree $m < 2\alpha$. For sufficiently small h it holds that*

$$\|\Delta^\alpha (I_h u - u)\|_{L^\infty(K_i)} \leq C_i h^{m+1-2\alpha} \left|u\right|_{W_\infty^{m+1}(K_i)}. \tag{A.1}$$

Proof. We observe that polyharmonic splines are in $C^{k-1}(K_i)$, hence $I_h u \in C^{2\alpha}(K_i)$, since $k > 2\alpha$. We now consider an auxiliary interpolation operator $I'_h : C(\Omega) \to V'_h$, where V'_h is obtained by augmentation of degree $m' = 2\alpha$. By the canonical splitting

$$\Delta^\alpha I_h u - \Delta^\alpha u = \Delta^\alpha I_h u - \Delta^\alpha I'_h u + \Delta^\alpha I'_h u - \Delta^\alpha u \tag{A.2}$$

Notice that $I_h u - I'_h u$ lies on a finite dimensional space $V_h + V'_h \subset W_\infty^{2\alpha}(K_i)$. This, combined with the fact that our Voronoi regions are shape-regular (due to the fact that our discretisation is quasi-uniform) allows us to use the inverse inequalities [51, Lemma (4.5.3)] to estimate

$$\|\Delta^\alpha I_h u - \Delta^\alpha I'_h u\|_{L^\infty(K_i)} \leq C_1 h^{-2\alpha} \|I_h u - I'_h u\|_{L^\infty(K_i)}. \tag{A.3}$$

The factor appearing on the right hand side can be further bound as

$$\begin{aligned} \|I_h u - I'_h u\|_{L^\infty(K_i)} &\leq \|I_h u - u\|_{L^\infty(K_i)} + \|I'_h u - u\|_{L^\infty(K_i)} && \text{(A.4)} \\ &\leq C_2 h^{m+1} \left|u\right|_{W_\infty^{m+1}(K_i)} + C_3 h^{m'+1} \left|u\right|_{W_\infty^{m'+1}(K_i)}, && \text{(A.5)} \end{aligned}$$

where the h^{m+1} term dominates for small h, since $m' > m$. Putting it all together we get

$$\|\Delta^\alpha I_h u - \Delta^\alpha I'_h u\|_{L^\infty(K_i)} \leq C_4 h^{-2\alpha+m+1} \left|u\right|_{W_\infty^{m+1}(K_i)}. \tag{A.6}$$

To estimate the other term in equation (A.2), we use the $m' \geq 2\alpha$ result to get

$$\|\Delta^\alpha I'_h u - \Delta^\alpha u\|_{L^\infty(K_i)} \leq C_5 h^{m'+1-2\alpha} \left|u\right|_{W_\infty^{m'+1}(K_i)}. \tag{A.7}$$

Again, the h^{m+1} term dominates, which concludes the proof:

$$\|\Delta^\alpha (I_h u - u)\|_{L^\infty(K_i)} \leq C_6 h^{m+1-2\alpha} \left|u\right|_{W_\infty^{m+1}(K_i)}. \tag{A.8}$$

References

[1] S. Širca, M. Horvat, Computational Methods in Physics, Graduate Texts in Physics, Springer International Publishing, Cham, 2018. doi:10.1007/978-3-319-78619-3.
URL http://link.springer.com/10.1007/978-3-319-78619-3

[2] G.-R. Liu, Mesh free methods: moving beyond the finite element method, CRC Press, 2002. doi:10.1201/9781420040586.

[3] V. P. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: a review and computer implementation aspects, Math. Comput. Simul 79 (3) (2008) 763–813. doi:10.1016/j.matcom.2008.01.003.
URL https://sourceforge.net/projects/elemfregalerkin/

[4] J. Slak, G. Kosec, On generation of node distributions for meshless pde discretizations, SIAM Journal on Scientific Computing 41 (5) (2019) A3202–A3229. doi:10.1137/18m1231456.
URL http://dx.doi.org/10.1137/18M1231456

[5] K. van der Sande, B. Fornberg, Fast variable density 3-D node generation, SIAM Journal on Scientific Computing 43 (1) (2021) A242–A257, arXiv:1906.00636 [cs, math]. doi:10.1137/20M1337016.
URL http://arxiv.org/abs/1906.00636

[6] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The finite element method: its basis and fundamentals, Elsevier, 2005.

[7] S. M. Mirfatah, B. Boroomand, E. Soleimanifar, On the solution of 3d problems in physics: from the geometry definition in cad to the solution by a meshless method, Journal of Computational Physics 393 (2019) 351–374.

[8] U. Duh, V. Shankar, G. Kosec, Discretization of Non-uniform Rational B-Spline (NURBS) Models for Meshless Isogeometric Analysis, Journal of Scientific Computing 100 (2) (2024) 51. doi:10.1007/s10915-024-02597-z.
URL https://link.springer.com/10.1007/s10915-024-02597-z

[9] J. A. Reeger, Adaptivity in local kernel based methods for approximating the action of linear operators, SIAM Journal on Scientific Computing

46 (4) (2024) A2683–A2708. doi:10.1137/23M1598052.
URL https://doi.org/10.1137/23M1598052

[10] D. T. Oanh, O. Davydov, H. X. Phu, Adaptive rbf-fd method for elliptic problems with point singularities in 2d, Applied Mathematics and Computation 313 (2017) 474–497.

[11] J. Slak, G. Kosec, Adaptive radial basis function–generated finite differences method for contact problems, International Journal for Numerical Methods in Engineering 119 (7) (2019) 661–686.

[12] E. Ortega, E. Oñate, S. Idelsohn, R. Flores, A meshless finite point method for three-dimensional analysis of compressible flow problems involving moving boundaries and adaptivity, International Journal for Numerical Methods in Fluids 73 (4) (2013) 323–343.

[13] M. Jančič, M. Založnik, G. Kosec, Meshless interface tracking for the simulation of dendrite envelope growth, Journal of Computational Physics 507 (2024) 112973. doi:10.1016/j.jcp.2024.112973.
URL https://linkinghub.elsevier.com/retrieve/pii/S0021999124002225

[14] V. Shankar, R. M. Kirby, A. L. Fogelson, Robust node generation for mesh-free discretizations on irregular domains and surfaces, SIAM Journal on Scientific Computing 40 (4) (2018) A2584–A2608.

[15] M. Jančič, J. Slak, G. Kosec, Monomial augmentation guidelines for rbf-fd from accuracy versus computational time perspective, Journal of Scientific Computing 87 (1) (2021) 9. doi:10.1007/s10915-020-01401-y.

[16] A. Tolstykh, D. Shirobokov, On using radial basis functions in a "finite difference mode" with applications to elasticity problems, Computational Mechanics 33 (1) (2003) 68–79.

[17] V. Bayona, N. Flyer, B. Fornberg, G. A. Barnett, On the role of polynomials in rbf-fd approximations: Ii. numerical solution of elliptic pdes, Journal of Computational Physics 332 (2017) 257–273. doi:10.1016/j.jcp.2016.12.008.

[18] V. Bayona, An insight into rbf-fd approximations augmented with polynomials, Computers & Mathematics with Applications 77 (9) (2019) 2337–2353.

[19] M. Jančič, G. Kosec, Strong form mesh-free hp-adaptive solution of linear elasticity problem, Engineering with Computers (May 2024). doi:10.1007/s00366-023-01843-6.
URL https://link.springer.com/10.1007/s00366-023-01843-6

[20] S. Shahane, A. Radhakrishnan, S. P. Vanka, A high-order accurate meshless method for solution of incompressible fluid flow problems, Journal of Computational Physics 445 (2021) 110623.

[21] M. E. Nielsen, B. Fornberg, High-order numerical method for solving elliptic partial differential equations on unfitted node sets, arXiv preprint arXiv:2407.15825 (2024).

[22] M. Rot, M. Jančič, G. Kosec, Spatially dependent node regularity in meshless approximation of partial differential equations, Journal of Computational Science 79 (2024) 102306.

[23] R. Platte, T. Driscoll, Eigenvalue stability of radial basis function discretizations for time-dependent problems, Mathematics with Applications 51 (8) (2006) 1251–1268. doi:10.1016/j.camwa.2006.04.007.
URL http://dx.doi.org/10.1016/j.camwa.2006.04.007

[24] B. Fornberg, E. Lehto, Stabilization of rbf-generated finite difference methods for convective pdes, Journal of Computational Physics 230 (6) (2011) 2270–2285. doi:10.1016/j.jcp.2010.12.014.
URL http://dx.doi.org/10.1016/j.jcp.2010.12.014

[25] I. Tominec, M. Nazarov, E. Larsson, Stability estimates for radial basis function methods applied to linear scalar conservation laws, Applied Mathematics and Computation 485 (2025) 129020. doi:10.1016/j.amc.2024.129020.
URL http://dx.doi.org/10.1016/j.amc.2024.129020

[26] I. Tominec, E. Larsson, A. Heryudono, A least squares radial basis function finite difference method with improved stability properties, SIAM Journal on Scientific Computing 43 (2) (2021) A1441–A1471.

doi:10.1137/20m1320079.
URL http://dx.doi.org/10.1137/20M1320079

[27] E. Larsson, V. Shcherbakov, A. Heryudono, A least squares radial basis function partition of unity method for solving pdes, SIAM Journal on Scientific Computing 39 (6) (2017) A2538–A2563. arXiv:https://doi.org/10.1137/17M1118087, doi:10.1137/17M1118087.
URL https://doi.org/10.1137/17M1118087

[28] J. M. Martel, R. B. Platte, Stability of radial basis function methods for convection problems on the circle and sphere, Journal of Scientific Computing 69 (2) (2016) 487–505. doi:10.1007/s10915-016-0206-9.
URL http://dx.doi.org/10.1007/s10915-016-0206-9

[29] A. Golbabai, et al., Analysis on the upwind local radial basis functions method to solve convection dominated problems and it's application for mhd flow, Engineering Analysis with Boundary Elements 100 (2019) 59–67.

[30] P. Lyra, K. Morgan, A review and comparative study of upwind biased schemes for compressible flow computation. part i: 1—d first—order schemes, Archives of Computational Methods in Engineering 7 (2000) 19–55.

[31] G. Kosec, B. Šarler, Simulation of macrosegregation with mesosegregates in binary metallic casts by a meshless method, Engineering Analysis with Boundary Elements 45 (2014) 36–44. doi:10.1016/j.enganabound.2014.01.016.
URL https://linkinghub.elsevier.com/retrieve/pii/S0955799714000290

[32] H. Ma, Chebyshev–legendre spectral viscosity method for nonlinear conservation laws, SIAM Journal on Numerical Analysis 35 (3) (1998) 869–892. doi:10.1137/s0036142995293900.
URL http://dx.doi.org/10.1137/S0036142995293900

[33] J. P. Boyd, Hyperviscous shock layers and diffusion zones: Monotonicity, spectral viscosity, and pseudospectral methods for very high order differential equations, Journal of Scientific Computing 9 (1) (1994) 81–106. doi:10.1007/BF01573179.

[34] V. Shankar, A. L. Fogelson, Hyperviscosity-based stabilization for radial basis function-finite difference (rbf-fd) discretizations of advection–diffusion equations, Journal of Computational Physics 372 (2018) 616–639. doi:10.1016/j.jcp.2018.06.036.

[35] V. Shankar, G. B. Wright, A. Narayan, A robust hyperviscosity formulation for stable rbf-fd discretizations of advection-diffusion-reaction equations on manifolds, SIAM Journal on Scientific Computing 42 (4) (2020) A2371–A2401. doi:10.1137/19m1288747.

[36] N. Flyer, G. A. Barnett, L. J. Wicker, Enhancing finite differences with radial basis functions: Experiments on the navier–stokes equations, Journal of Computational Physics 316 (2016) 39–62. doi:10.1016/j.jcp.2016.02.078.
URL http://dx.doi.org/10.1016/j.jcp.2016.02.078

[37] N. Flyer, E. Lehto, S. Blaise, G. B. Wright, A. St-Cyr, A guide to rbf-generated finite differences for nonlinear transport: Shallow water simulations on a sphere, Journal of Computational Physics 231 (11) (2012) 4078–4095. doi:10.1016/j.jcp.2012.01.028.
URL http://dx.doi.org/10.1016/j.jcp.2012.01.028

[38] I. Tominec, M. Nazarov, Residual viscosity stabilized rbf-fd methods for solving nonlinear conservation laws, Journal of Scientific Computing 94 (1) (Dec. 2022). doi:10.1007/s10915-022-02055-8.
URL http://dx.doi.org/10.1007/s10915-022-02055-8

[39] C. A. J. Fletcher, Generating exact solutions of the two-dimensional burgers' equations, International Journal for Numerical Methods in Fluids 3 (3) (1983) 213–216. doi:10.1002/fld.1650030302.
URL http://dx.doi.org/10.1002/fld.1650030302

[40] H. Zhu, H. Shu, M. Ding, Numerical solutions of two-dimensional burgers' equations by discrete adomian decomposition method, Mathematics with Applications 60 (3) (2010) 840–848. doi:10.1016/j.camwa.2010.05.031.
URL http://dx.doi.org/10.1016/j.camwa.2010.05.031

[41] V. Bayona, An insight into rbf-fd approximations augmented with polynomials, Mathematics with Applications 77 (9) (2019) 2337–2353.

doi:10.1016/j.camwa.2018.12.029.
URL http://dx.doi.org/10.1016/j.camwa.2018.12.029

[42] G. E. Fasshauer, Meshfree Approximation Methods with Matlab: (With CD-ROM), WORLD SCIENTIFIC, 2007. doi:10.1142/6437.
URL http://dx.doi.org/10.1142/6437

[43] S. Le Borne, W. Leinen, Guidelines for rbf-fd discretization: Numerical experiments on the interplay of a multitude of parameter choices, Journal of Scientific Computing 95 (1) (2023) 8. doi:10.1007/s10915-023-02123-7.
URL https://doi.org/10.1007/s10915-023-02123-7

[44] Ž. Vaupotič, M. Rot, G. Kosec, Hyperviscosity stabilisation of the rbf-fd solution to natural convection, Journal of Physics: Conference Series 2766 (1) (2024) 012160. doi:10.1088/1742-6596/2766/1/012160.
URL http://dx.doi.org/10.1088/1742-6596/2766/1/012160

[45] N. Flyer, B. Fornberg, V. Bayona, G. A. Barnett, On the role of polynomials in rbf-fd approximations: I. interpolation and accuracy, Journal of Computational Physics 321 (2016) 21–38. doi:10.1016/j.jcp.2016.05.026.
URL http://dx.doi.org/10.1016/j.jcp.2016.05.026

[46] H.-p. Ma, H.-y. Li, Super spectral viscosity method for nonlinear conservation laws, Journal of Shanghai University (English Edition) 10 (1) (2006) 9–14. doi:10.1007/s11741-006-0098-2.
URL http://dx.doi.org/10.1007/s11741-006-0098-2

[47] R. Horn, C. Johnson, Matrix Analysis, Cambridge University Press, 2012.

[48] Y. Saad, Variations on arnoldi's method for computing eigenelements of large unsymmetric matrices, Linear Algebra and its Applications 34 (1980) 269–295. doi:10.1016/0024-3795(80)90169-x.
URL http://dx.doi.org/10.1016/0024-3795(80)90169-X

[49] E. Tadmor, Burgers' equation with vanishing hyper-viscosity, Communications in Mathematical Sciences 2 (2) (2004) 317–324. doi:10.4310/cms.2004.v2.n2.a9.
URL http://dx.doi.org/10.4310/CMS.2004.v2.n2.a9

[50] A. Hussein, Partial and full hyper-viscosity for navier-stokes and primitive equations, Journal of Differential Equations 269 (4) (2020) 3003–3030. doi:10.1016/j.jde.2020.02.019.
URL http://dx.doi.org/10.1016/j.jde.2020.02.019

[51] S. C. Brenner, L. R. Scott, The Mathematical Theory of Finite Element Methods, Springer New York, 2008. doi:10.1007/978-0-387-75934-0.
URL http://dx.doi.org/10.1007/978-0-387-75934-0

[52] A. Kolar-Požun, M. Jančič, G. Kosec, A superconvergence result in the rbf-fd method, Journal of Physics: Conference Series 2766 (1) (2024) 012161. doi:10.1088/1742-6596/2766/1/012161.
URL http://dx.doi.org/10.1088/1742-6596/2766/1/012161

[53] J. Slak, G. Kosec, Medusa: A C++ Library for Solving PDEs Using Strong Form Mesh-free Methods, ACM Transactions on Mathematical Software 47 (3) (2021) 1–25. doi:10.1145/3450966.
URL https://dl.acm.org/doi/10.1145/3450966

[54] Y. Qiu, Spectra 1.1.0, a c++ library for large scale eigenvalue problems, https://spectralib.org/ (2025).

[55] G. Gunnebaud, B. Jacob, et al., Eigen v3, http://eigen.tuxfamily.org (2010).