A New Class of High-Order Methods for Fluid Dynamics Simulations Using Gaussian Process Modeling: One-Dimensional Case

Adam Reyes¹ · Dongwook Lee² · Carlo Graziani³ · Petros Tzeferacos^{3,4}

Received: 11 May 2017 / Revised: 17 November 2017 / Accepted: 30 November 2017 /
Published online: 6 December 2017
© The Author(s) 2017

Abstract We introduce an entirely new class of high-order methods for computational fluid dynamics based on the Gaussian process (GP) family of stochastic functions. Our approach is to use kernel-based GP prediction methods to interpolate/reconstruct high-order approximations for solving hyperbolic PDEs. We present a new high-order formulation to solve (magneto)hydrodynamic equations using the GP approach that furnishes an alternative to conventional polynomial-based approaches.

Keywords Gaussian processes · Stochastic models · High-order methods · Finite volume method · Gas dynamics · Magnetohydrodynamics

1 Introduction

Cutting edge simulations of gas dynamics and magnetohydrodynamics (MHD) have been among the headliner applications of scientific high-performance computing (HPC) [12, 13, 32, 61]. They are expected to remain important as new HPC architectures with ever more

✉ Dongwook Lee
dlee79@ucsc.edu

Adam Reyes
acreyes@ucsc.edu

Carlo Graziani
carlo@oddjob.uchicago.edu

Petros Tzeferacos
petros.tzeferacos@flash.uchicago.edu

¹ Department of Physics, University of California, Santa Cruz, CA, USA

² Applied Mathematics and Statistics, University of California, Santa Cruz, CA, USA

³ Department of Astronomy and Astrophysics, Flash Center for Computational Science, University of Chicago, Chicago, IL, USA

⁴ Department of Physics, Oxford University, Oxford, UK

powerful capabilities come online in the decades to come. A notable trend in recent HPC developments concerns the hardware design of the newer architectures. It is expected that newer HPC architectures will feature a radical change in the balance between computation and memory resources, with memory per compute core declining dramatically from current levels [2, 12, 61]. This trend tells us that new algorithmic strategies will be required to meet the goals of saving memory and accommodating increased computation. This paradigm shift in designing scientific algorithms has become a great importance in HPC applications. In the context of numerical methods for computational fluid dynamics (CFD), one desirable approach is to design high-order accurate methods [61] that, in contrast to low-order methods, can achieve an increased target solution accuracy more efficiently and more quickly by computing increased higher-order floating-point approximations on a given grid resolution [25, 36, 37]. This approach embodies in a concrete manner the desired tradeoff between memory and computation by exercising more computation per memory—or equivalently, the equal amount of computation with less memory.

Within the broad framework of finite difference method (FDM) and finite volume method (FVM) discretizations, discrete algorithms of data interpolation and reconstruction play a key role in numerical methods for PDE integration [36, 37, 62]. They are frequently the limiting factor in the convergence rate, efficiency, and algorithmic complexity of a numerical scheme in FDM [26, 43] and in FVM [7, 35, 36, 42, 55, 62]. More generally, interpolation and reconstruction are not only essential for estimating high-order accurate approximations for fluxes at quadrature points on each cell but also for interface tracking; for prolonging states from coarse zones to corresponding refined zones in adaptive-mesh refinement (AMR) schemes; and for various other contexts associated with high-order solutions. In CFD simulations, these interpolation and reconstruction algorithms must be carried out as accurately as possible, because, to a large extent, their accuracy is one of the key factors that determine the overall accuracy of the simulation.

Polynomial-based approaches are the most successful and popular among interpolation/reconstruction methods in this field. There are a couple of convincing reasons for this state of affairs. First, they are easily relatable to Taylor expansion, the most familiar of function approximations. Second, the nominal N -th order accuracy of polynomial interpolation/reconstruction is derived from using polynomials of degree $(N - 1)$, bearing a leading term of the error that scales with $\mathcal{O}(\Delta^N)$ as the local grid spacing Δ approaches to zero [36, 37, 62]. However, the simplicity of polynomial interpolation/reconstruction comes at a price: the polynomial approach is notoriously prone to oscillations in data fitting, especially with discontinuous data [20]; furthermore, in many practical situations the high-order polynomial interpolation/reconstruction must be carried out on a *fixed* size of stencils, whereby there is a one-to-one relationship between the order of the interpolation/reconstruction and the size of the stencils. This becomes a restriction in particular when unstructured meshes are considered in multiple spatial dimensions. Lastly, another related major issue lies in the fact that the algorithmic complexity of such polynomial based schemes typically grows with an order of accuracy [18], as well as with a spatial dimensionality [7, 42, 73] in FVM.

To overcome the aforementioned issues in polynomial methods, practitioners have developed the so-called “non-polynomial” interpolation/reconstruction based on the mesh-free Radial Basis Function (RBF) approximations in the last decades. The core idea is to replace the polynomial interpolants with RBFs, which is a part of a very general class of approximants from the field known as *Optimal Recovery* (OR) [68]. Several interpolation techniques in OR have shown practicable in the framework of solving hyperbolic PDEs [31, 46, 58], parabolic PDEs [44, 45], diffusion and reaction-diffusion PDEs [53], boundary value problems of elliptic PDEs [40], interpolations on irregular domains [9, 24, 41], and also interpolations on a

more general set of scattered data [17]. Historically, RBFs were introduced to seek exact function interpolations [47]. Recently, the RBF approximations have been combined with the key ideas of handling discontinuities in the ENO [23] and WENO [26, 39] methods. Such approaches, termed as ENO/WENO-RBF, have been extended to solve nonlinear scalar equations and the Euler equations in the FDM [29] and the FVM [21] frameworks. These studies focused on designing their RBF methods with the use of adaptive shape parameters to control local errors. Also in [3], two types of multiquadrics and polyharmonic spline RBFs were used to model the Euler equations with a strategy of selecting optimal shape parameters for different RBF orders. Stability analysis on the fully discretized hyperbolic PDEs in both space and time using the multiquadrics RBF is reported in [8]. Admittedly, while there exist a few conceptual resemblances between these RBF approaches and our new GP method, there are fundamental differences that distinguish the two approaches, which will be discussed later in this paper.

The goal of this article is to develop a new class of high-order methods that overcome the aforementioned difficulties in polynomial approaches by exploiting the alternative perspective afforded by GP modeling, a methodology borrowed from the field of statistical data modeling. In view of the novelty of our approach, which bridges the two distinct research fields of statistical modeling and CFD, it is our intention in this paper to first construct a mathematical formulation in a 1D framework. The current study will serve as a theoretical foundation for multidimensional extensions of the GP modeling for CFD applications, the topics of which will be studied in our future work.

Our GP method is a class of high-order schemes primarily designed for the numerical evolution of hyperbolic PDEs, $q_t + \nabla \cdot F(q) = 0$. To this end, we describe the new high-order Gaussian Process (GP) approximation strategies in two steps:

1. GP *interpolation* that works on pointwise values of $q(x_i)$ as both inputs and outputs, and
2. GP *reconstruction* that works on volume-averaged values $\langle q_i \rangle = \frac{1}{\Delta x} \int_{\Delta x} q(x, t^n) dx$ as inputs, reconstructing pointwise values as outputs.

The GP interpolation will provide a “baseline” formulation of using GP as a new high-order interpolator operating on the *same* data type, while the GP reconstruction will serve as a high-order reconstructor operating on two *different* types of data.

2 Gaussian Process Modeling

The theory of GP, and more generally of stochastic functions, dates back to the work of Wiener [69] and Kolmogorov [33]. Modern-day applications are numerous. Just in the physical sciences, GP prediction is in common use in meteorology, geology, and time-series analysis [4, 48, 67], and in cosmology, where GP models furnish the standard description of the Cosmic Microwave Background [5]. Applications abound in many other fields, in particular, wherever spatial or time-series data requires “nonparametric” modeling [4, 48, 59]. From the perspective of CFD applications, our specific goal in this study is to use the predictive GP modeling that is processed by training observed data, (e.g., cell-averaged fluid variables at cell centers) to produce a “data-informed” prediction (e.g., pointwise Riemann state values at cell interfaces). In what follows we give a brief overview on GP from the statistical perspective (Sect. 2.1), followed by our strategies of tuning GP for high-order interpolation (Sect. 2.2) and reconstruction (Sect. 2.3) in CFD applications. Readers who wish to pursue the subject in greater detail are referred to “Appendix A” toward the end of this paper as well as [4, 48, 59]. Readers who are more interested in our new GP interpolation and reconstruction

algorithms in the framework of FDM and FVM than in the basic Bayesian theory may start reading our paper from Sect. 2.2 after briefly reviewing Eqs. (1) and (3) in Sect. 2.1.

2.1 GP—Statistical Perspective

GP is a class of stochastic processes, i.e., processes that sample functions (rather than points) from an infinite dimensional function space. Formally, a GP is a collection of random variables, any finite collection of which has a joint Gaussian distribution [4, 48]. For the purposes of interpolation/reconstruction, this would be the collection of input data, either pointwise values $q(x_i)$ or the volume averaged $\langle q_i \rangle$. In the language of GP, the input data form a joint Gaussian distribution with function values that are to be interpolated/reconstructed. In this way, the known input data (e.g., $q(x_i)$ or $\langle q_i \rangle$) can be used to “train” a probability distribution for the unknown function values (e.g., $q(x_*)$ at any arbitrary point x_*), with a *posterior* mean and uncertainty that are compatible with the observed data. The updated mean value is our target interpolation/reconstruction for FDM and FVM.

A GP is fully defined by two functions:

- a mean function $\bar{f}(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ over $\mathbb{R}^N$, and
- a covariance kernel function which is a symmetric, positive-definite integral kernel $K(\mathbf{x}, \mathbf{y}) = \mathbb{E}[(f(\mathbf{x}) - \bar{f}(\mathbf{x}))(f(\mathbf{y}) - \bar{f}(\mathbf{y}))]$ over $\mathbb{R}^N \times \mathbb{R}^N$.

The mean function incorporates prior information about the interpolating data. Typically the mean function is expressed using explicit basis functions with some parameters that can be inferred from the input data (see [4, 48]). One can even use a constant mean function, $\bar{f}(x) = f_0$, where the optimal choice of f_0 can be determined from the data (see “Appendix B”). In practice, and throughout this paper, a choice of constant zero mean, $f_0 = 0$, works well and saves additional computational cost in the reconstruction procedure.

The covariance kernel function, $K(\mathbf{x}, \mathbf{y})$, sets the relationship between pairs of function values, and often contain additional parameters called *hyperparameters*, that are entirely independent of the interpolation/reconstruction conditioning procedure. In this paper, we use the “Squared Exponential” (SE) kernel for the default choice of $K(\mathbf{x}, \mathbf{y})$,

$$K(\mathbf{x}, \mathbf{y}) = K_{\text{SE}}(\mathbf{x}, \mathbf{y}) = \Sigma^2 \exp \left[-\frac{(\mathbf{x} - \mathbf{y})^2}{2\ell^2} \right]. \quad (1)$$

More discussion on this choice of covariance kernel is in Sect. 2.4. The SE kernel contains the length hyperparameter ℓ , which controls the characteristic length scale of the functions that GP samples. It plays a critical role in the accuracy of the resultant interpolation/reconstruction. An in-depth discussion of the role of ℓ is contained in Sect. 5.2.

Such functions f , drawn randomly from this distribution, are said to be sampled from a Gaussian Process with a mean function $\bar{f}(\mathbf{x})$ and a covariance function $K(\mathbf{x}, \mathbf{y})$, and we write $f \sim \mathcal{GP}(\bar{f}, K)$. Both $\bar{f}(x)$ and $K(\mathbf{x}, \mathbf{y})$ then control the character (e.g., length scales, differentiability, oscillation strength) of “likely” functions. Now that N “training” points $\mathbf{x}_i$, $i = 1, \dots, N$ at which the function values $f(\mathbf{x}_i)$ are given, we may calculate the likelihood $\mathcal{L}$ (the probability of $\mathbf{f}$ given the GP model) of the data vector $\mathbf{f} \equiv [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ (e.g., N many pointwise values of density ρ at $\mathbf{x}_i$, $i = 1, \dots, N$) by

$$\mathcal{L} \equiv P(\mathbf{f}) = (2\pi)^{-\frac{N}{2}} \det |\mathbf{K}|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{f} - \bar{\mathbf{f}})^T \mathbf{K}^{-1} (\mathbf{f} - \bar{\mathbf{f}}) \right], \quad (2)$$

where $\mathbf{K} = [K_{ij}]_{i,j=1,\dots,N}$ with $K_{ij} \equiv K(\mathbf{x}_i, \mathbf{x}_j)$. Here $\mathcal{L}$ is a measure of the compatibility of the data $\mathbf{f}$ and the GP model, being specified by $\bar{f}(\mathbf{x})$ and $K(\mathbf{x}, \mathbf{y})$.

Given the function samples $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ obtained at spatial points $\mathbf{x}_i$, $i = 1, \dots, N$, GP predictions aim to make a probabilistic statement about the value $f_* \equiv f(\mathbf{x}_*)$ of the unknown function $f \sim \mathcal{GP}(\bar{f}, K)$ at a new spatial point $\mathbf{x}_*$. This is particularly of interest to us from the perspectives of FDM and FVM, because we can use GP to predict an unknown function value at cell interfaces (e.g., $q_{i \pm \frac{1}{2}}$ in 1D) where both FDM and FVM require estimates of flux functions.

We arrive at a conclusion now: we can accomplish this by utilizing the conditioning property of GP from the theory of Bayesian inference [4, 48] (details of this construction can be found in “Appendix A”) to obtain the result of an updated *posterior mean function*,

$$\tilde{f}_* \equiv \bar{f}(\mathbf{x}_*) + \mathbf{k}_*^T \mathbf{K}^{-1} \cdot (\mathbf{f} - \bar{\mathbf{f}}), \quad (3)$$

where $k_{*,i} = K(\mathbf{x}_*, \mathbf{x}_i)$. If we take a constant zero mean, $\bar{f}(x) = 0$, as mentioned earlier, Eq. (3) becomes

$$\tilde{f}_* = \mathbf{k}_*^T \mathbf{K}^{-1} \mathbf{f}. \quad (4)$$

Equations (3) and (4) reveal the main structure of GP prediction, which is the term $\mathbf{k}_*^T \mathbf{K}^{-1}$, that is specified entirely by the choice of covariance kernel function, without the need to seek any fixed number of parameters that may depend on the datasets under consideration.

This provides a clear and important conceptual difference between the GP predictions and the RBF approximations. RBF models in the context of interpolations and reconstructions follow the parametric modeling paradigm, requiring to solve a linear system $\mathbf{A}\boldsymbol{\lambda} = \mathbf{F}$ (see e.g., [3, 31]) whose size is determined and *fixed* by the number of desired N interpolation points $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ and by the associated coefficients (or parameters) λ_n and c_m of an RBF interpolating function $s(\mathbf{x})$,

$$s(\mathbf{x}) = \sum_{n=1}^N \lambda_n \Phi(\|\mathbf{x} - \mathbf{x}_n\|, \epsilon_n) + \sum_{m=1}^M c_m p_m(\mathbf{x}). \quad (5)$$

The components of the solution vector $\boldsymbol{\lambda}$ are λ_n and c_m , $\mathbf{A}$ is the matrix whose components are set by the values of RBFs $\Phi_{i,j} = \Phi(\|\mathbf{x}_i - \mathbf{x}_j\|, \epsilon_j)$ at two points $\mathbf{x}_i$ and $\mathbf{x}_j$ and by the shape parameter $\epsilon_j > 0$, $\mathbf{F}$ is the vector with entries of $\mathbf{f}$, and the last term in Eq. (5) is a polynomial (at most) M -th degree with m -th standard basis polynomials p_m . Unlike GP, a probability distribution (if needed) is not directly imposed on $s(\mathbf{x})$, rather, it can only be induced by a prior distribution defined on the coefficients. Even though the two approaches resemble each other in terms of using “kernels” (i.e., RBFs and covariance kernel functions), they are fundamentally different from these statistical viewpoints.

2.2 High-Order GP Interpolation for CFD

In this paper, we are most interested in developing a high-order reconstruction method for FVM in which the function samples $\mathbf{f}$ are given as volume-averaged data $\langle q_i \rangle$. However, we first consider an interpolation method which uses pointwise data q_i as function samples $\mathbf{f}$. An algorithmic design of GP interpolation using pointwise data will provide a good mathematical foundation for FVM which reconstructs pointwise values from volume-averaged data.

The mean $\tilde{f}_*$ of the distribution given in Eq. (3) is our interpolation of the function f at the point $\mathbf{x}_* \in \mathbb{R}^D$, $D = 1, 2, 3$, where f is any given fluid variable such as density, pressure, velocity fields, magnetic fields, etc. For the purpose of exposition, let us use q to denote one of such fluid variables (e.g., density $\rho = \rho(\mathbf{x}, t^n)$). Also, the mathematical descriptions of both interpolation and reconstruction will be considered in 1D hereafter.

We are interested in seeking a high-order interpolation of q at $\mathbf{x}_* = x_* = x_{i \pm \frac{1}{2}}$ on a stencil $\cup_{k=i-p}^{i+r} I_k$,

$$q_{i \pm \frac{1}{2}} = \mathcal{I}_{GP}(q_{i-p}, \dots, q_{i+r}), \quad (6)$$

where $\mathcal{I}_{GP}(\cdot)$ is the GP interpolation given in Eq. (3). We define

$$\mathbf{f} = [q_{i-p}, \dots, q_{i+r}]^T. \quad (7)$$

Furthermore, if we simply assume a constant mean f_0 for the data in Eq. (7), we get

$$\bar{\mathbf{f}} = f_0 \mathbf{1}_{r+p+1}, \quad \bar{f}(x_{i \pm \frac{1}{2}}) = f_0, \quad (8)$$

where $\mathbf{1}_{r+p+1} = [1, \dots, 1]^T$ is an $(r + p + 1)$ -dimensional one-vector. In case of a zero mean $f_0 = 0$, the GP interpolation scheme simply becomes

$$q_{i \pm \frac{1}{2}} = \mathbf{k}_{i \pm \frac{1}{2}}^T \mathbf{K}^{-1} \cdot \mathbf{f}. \quad (9)$$

As shown in Eqs. (3) and (9), the interpolant $\tilde{f}_* = q_{i \pm \frac{1}{2}}$ is a simple linear combination of the observed data $\mathbf{f}$ and the covariance kernels $\mathbf{k}_*$ and $\mathbf{K}$, anchored by one of its arguments to one of the data points, x_* , $x_{i-p}, \dots, x_{i+r}$. The second term in Eq. (3) can also be cast as an inner product between a vector of weights $\mathbf{w}^T \equiv \mathbf{k}_*^T \mathbf{K}^{-1}$ and a vector of data residuals $(\mathbf{f} - \bar{\mathbf{f}})$.

The weights $\mathbf{w}$ are independent of the data values $\mathbf{f}$ and they depend only on the locations of the data points x_i and on the desired interpolation point x_* . This is useful because, in hydrodynamic simulations, the training point locations (viz. a GP stencil) and the interpolation points (viz. cell interfaces) are often known in advance, as is the case with static Cartesian grid configurations. In such cases, the weight vector $\mathbf{w}$ can be computed and stored in advance at an initialization step and remain constant throughout the simulation. When an adaptive mesh refinement (AMR) configuration is considered, $\mathbf{w}$ can be computed for all possible grid refinement levels and stored a priori for later use. The GP interpolations then come at the cost of the remaining inexpensive inner product operation between $\mathbf{w}^T$ and $(\mathbf{f} - \bar{\mathbf{f}})$ in Eq. (3), whose operation count is linearly proportional to the number of points in the stencil. Typically in 1D, the size of the stencil is one for the first order Godunov (FOG) method [19, 62]; three for the 2nd order piecewise linear methods (PLM) [62, 66]; five for both the 3rd order piecewise parabolic method (PPM) [10] and the 5th order Weighted Essentially Non-oscillatory (WENO) method [26]. Given such stencil sizes, the cost of the linear solves required by Eq. (3) is minor. For instance, $\mathbf{K}$ is a 5×5 matrix when using a stencil of five grid points, namely, the 5-point GP stencil centered at x_i with a radius of two-cell length $S_2 = \cup_{k=i-2}^{i+2} I_k$. In general, we define a stencil S_R centered at x_i with a radius R cell-length,

$$S_R = \bigcup_{k=i-R}^{i+R} I_k. \quad (10)$$

Note that the interpolation point x_* may be anywhere in the continuous interval S_R .

Since the matrix $\mathbf{K}$ is symmetric positive-definite, the inversion of $\mathbf{K}$ can be obtained efficiently by Cholesky decomposition which is about a factor 2 faster than the usual LU decomposition. The decomposition is needed only once, at initialization time, for the calculation of the vector of weights $\mathbf{w}$.

An important feature of GP interpolation is that it naturally supports multidimensional stencil configurations. The reason for this is that there are many valid covariance functions

over $\mathbf{R}^D$ that are isotropic, and therefore do not bias interpolations and reconstructions along any special direction. The possibility of directionally-unbiased reconstruction over multidimensional stencils is a qualitative advantage of GP interpolation, especially when designing high-order algorithms [7, 42, 73]. Such high-order multidimensional GP methods will be studied in our future work.

2.3 High-Order GP Reconstruction for CFD

In FVM, the fluid variables to be evolved are not pointwise values but rather are volume-averaged integral quantities, $\langle q_i \rangle = \frac{1}{\Delta V_i} \int_{\Delta V_i} q(\mathbf{x}, t^n) dV$. In FDM, on the other hand, the main task is to find a high-order approximation to the interface flux values $\hat{F}_{i+\frac{1}{2}}$, given the fact that their integral quantities $F(q_i) = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{F}(\xi) d\xi$ are known via the analytic evaluations of the flux function F at the given pointwise data set q_i . The conservative FDM updates in 1D then readily proceed by solving

$$\frac{dq_i}{dt} + \frac{1}{\Delta x} \left(\hat{F}_{i+\frac{1}{2}} - \hat{F}_{i-\frac{1}{2}} \right) = 0. \quad (11)$$

In both cases, we see that there is a *change in data types*—hence the name “reconstruction”—between input [e.g., $\langle q_i \rangle$ for FVM; $F(q_i)$ for FDM] and output (e.g., $q(\mathbf{x}, t^n)$ for FVM; $\hat{F}_{i+\frac{1}{2}}$ for FDM) pairs in such a way that high-order approximations are applied to the integral quantities (inputs) and produce corresponding pointwise values (outputs), with high accuracy.

The GP interpolation method outlined in Sect. 2.2 should, therefore, be modified so that reconstruction may account for such data type changes in both FDM and FVM. Note that the integral averages over a grid cell constitute “linear” operations on a function $f(\mathbf{x})$. Similar to the case with ordinary finite-dimensional multivariate Gaussian distributions where linear operations on Gaussian random variables result in new Gaussian random variables with linearly transformed means and covariances, a set of N linear functionals operating on a GP-distributed function f has an N -dimensional Gaussian distribution with mean and covariance that are linear functionals of the GP mean function and the covariance function.

Suppose, for example, in an FVM sense, we consider a GP reconstruction on a GP stencil $\cup_{k=i-p}^{i+r} I_k$. First we define $r+p+1$ measures $dg_k(\mathbf{x})$, $k = i-p, \dots, i+r$, defining $r+p+1$ linear functionals,

$$G_k \equiv \int f(\mathbf{x}) dg_k(\mathbf{x}), \quad k = i-p, \dots, i+r. \quad (12)$$

For the sake of a reconstruction scheme on a mesh of control volumes, we choose the measures $dg_k(\mathbf{x})$ to be the cell volume-average measures,

$$dg_k(\mathbf{x}) = \begin{cases} d^3\mathbf{x} \times \prod_{d=x,y,z} \frac{1}{\Delta^{(d)}} & \text{if } \mathbf{x} \in I_k, \\ 0 & \text{if } \mathbf{x} \notin I_k, \end{cases} \quad (13)$$

where $\Delta^{(d)}$ is the grid spacing in the d -direction, and the k -th cell I_k is given by $I_k = \prod_{d=x,y,z} I_k^{(d)}$ with 1D cells $I_k^{(d)} = [x_k^{(d)} - \frac{\Delta^{(d)}}{2}, x_k^{(d)} + \frac{\Delta^{(d)}}{2}]$ for each d -direction. For the purpose of the current discussion, we will assume a locally-uniform rectilinear grid of cubical cells I_k of uniform size $\Delta = \Delta^{(x)} = \Delta^{(y)} = \Delta^{(z)}$.

Then, the vector $\mathbf{G} = [G_{i-p}, \dots, G_{i+r}]^T$ is normally distributed with mean $\bar{\mathbf{G}} = [\bar{G}_{i-p}, \dots, \bar{G}_{i+r}]^T$ and covariance matrix $\mathbf{C} = [\mathbf{C}_{kh}]_{k,h=i-p,\dots,i+r}$, where

$$\bar{G}_k = \mathbb{E}[G_k] = \int \mathbb{E}[f(\mathbf{x})] dg_k(\mathbf{x}) = \int \bar{f}(\mathbf{x}) dg_k(\mathbf{x}), \quad (14)$$

and

$$\begin{aligned} \mathbf{C}_{kh} &= \mathbb{E}[(G_k - \bar{G}_k)(G_h - \bar{G}_h)] \\ &= \int \mathbb{E}[(f(\mathbf{x}) - \bar{f}(\mathbf{x}))(f(\mathbf{y}) - \bar{f}(\mathbf{y}))] dg_k(\mathbf{x}) dg_h(\mathbf{y}) \\ &= \int K(\mathbf{x}, \mathbf{y}) dg_k(\mathbf{x}) dg_h(\mathbf{y}). \end{aligned} \quad (15)$$

Thus, we see that the GP distribution on the function f leads to a multivariate Gaussian distribution on any $(r+p+1)$ -dimensional vector $\mathbf{G}$ of linear functionals of f . This Gaussian distribution can be used for likelihood maximization in a manner completely analogous to the case of interpolation, where training is performed using pointwise data.

For reconstruction, it is also not hard to generalize Eq. (3). We first define the $(r+p+1)$ -dimensional *prediction vector* $\mathbf{T}_* = [\mathbf{T}_{*,k}]_{k=i-p,\dots,i+r}$ at $\mathbf{x}_*$ by

$$\mathbf{T}_{*,k} \equiv \int K(\mathbf{x}, \mathbf{x}_*) dg_k(\mathbf{x}). \quad (16)$$

Then the pointwise function value $f(\mathbf{x}_*)$ at the point $\mathbf{x}_*$, reconstructed from the volume-averaged data $\mathbf{G}$, is given by

$$\tilde{f}_* = \bar{f}(\mathbf{x}_*) + \mathbf{T}_*^T \mathbf{C}^{-1} (\mathbf{G} - \bar{\mathbf{G}}). \quad (17)$$

Notice that the sample data $\mathbf{G}$ now holds volume-averaged quantities on a stencil $\cup_{k=i-p}^{i+r} I_k$,

$$\mathbf{G} = [\langle q_{i-p} \rangle, \dots, \langle q_{i+r} \rangle]^T. \quad (18)$$

The $(r+p+1)$ -dimensional vector $\mathbf{T}_*$ is the covariance between cell average quantities $\langle q_k \rangle$, $k = i-p, \dots, i+r$, and the pointwise value q_* at $\mathbf{x}_*$, $\mathbf{C}$ is the $(r+p+1) \times (r+p+1)$ covariance kernel matrix between cell average values $\langle q_k \rangle$, and $\mathbf{z} = \mathbf{C}^{-1} \mathbf{T}_*$ is the $(r+p+1)$ -dimensional vector of weights for the data $(\mathbf{G} - \bar{\mathbf{G}})$. Equation (17) is a straightforward generalization of Eq. (3), producing an integral version of it.

In a simple 1D case with a constant mean f_0 and $\mathbf{x}_* = x_{i \pm \frac{1}{2}}$, the mean vector $\bar{\mathbf{G}}$ is written as

$$\bar{\mathbf{G}} = f_0 \mathbf{1}_{r+p+1}, \quad (19)$$

and the GP reconstruction in Eq. (17) becomes

$$\begin{aligned} \tilde{f}_{i \pm \frac{1}{2}} &= f_0 + \mathbf{T}_{i \pm \frac{1}{2}}^T \mathbf{C}^{-1} (\mathbf{G} - f_0 \mathbf{1}_{r+p+1}) \\ &= f_0 + \mathbf{z}^T (\mathbf{G} - f_0 \mathbf{1}_{r+p+1}). \end{aligned} \quad (20)$$

Analogous considerations arise in the RBF reconstruction methods for volume-averaged data. In [3] the RBF function is integrated and is used for reconstruction, a procedure that corresponds closely to integrating the kernel function in Eqs. (15) and (16).

2.4 The GP-SE Model

One of the most widely-used kernels in GP modeling is the “Squared Exponential (SE)” covariance kernel function [4, 48] which has been introduced in Eq. (1),

$$K(\mathbf{x}, \mathbf{y}) \equiv \Sigma^2 \exp \left[-\frac{(\mathbf{x} - \mathbf{y})^2}{2\ell^2} \right]. \quad (21)$$

This GP-SE model features three free parameters, f_0 , Σ^2 , and ℓ . Often, in Eqs. (3) and (17), a constant mean function $\tilde{f}(\mathbf{x}) = f_0 \mathbf{1}_N$ is adopted for simplicity, where $\mathbf{1}_N = [1, \dots, 1]^T$ is an N -dimensional one-vector. The constant f_0 can be analytically determined from the data as part of the interpolation process by maximizing the likelihood function in Eq. (2). See “Appendix B” for more details.

The latter two, Σ^2 and ℓ , are called “hyperparameters” which are the parameters that are built into the kernel. The hyperparameter Σ^2 has no effect on the posterior mean function, so one can set $\Sigma^2 = 1$ for simplicity. On the other hand, the hyperparameter ℓ is the correlation length scale of the model. It determines the length scale of variation preferred by the GP model. Our GP predictions for interpolation/reconstruction, which necessarily agrees with the observed values of the function at the training points $\mathbf{x}_k$, may “wiggle” on this scale between training points. In this sense, ℓ is a “rigidity”, controlling the curvature scales of the prediction, that should correspond to the physical length scales of the features GP is to resolve. Since we want function interpolations/reconstructions that are smooth on the scale of the grid, we certainly want $\ell > \Delta$, and would also prefer $\ell \geq R$. As just mentioned, the choice of ℓ requires a balance between the physical length scales in the problem and the grid scale. This implies that different values of ℓ can be employed depending on differing length scales on different regions of the computational domain. For this paper, we select a constant value for ℓ , which has a direct impact on the solution accuracy, as shown in Fig. 3c. The accuracy of the scheme is intimately tied to the scales of the flow and ideally, would need to be recalculated throughout a simulation to best capture the features specific to the problem at hand. Such a scheme would require additional mathematical and algorithmic considerations and we leave investigation on best practice for determining such an optimal ℓ within the context of FDM/FVM reconstructions to our forthcoming papers. We have found that values of $\ell/\Delta = 6$ or 12 to perform robustly and accurately under the grid and stencil sizes considered in this paper.

The relationship often leads to the *trade-off principle* in that there is a conflicting balance between the obtainable accuracy and the numerical stability due to a large condition number of the kernel matrix when a grid is highly refined. For instance, in our GP-SE model, SE suffers a notorious singularity issue where the kernel is prone to yield nearly singular matrices when the distance between any two points $\mathbf{x}$ and $\mathbf{y}$ becomes progressively smaller (or equivalently, the grid becomes progressively refined in CFD).

In the statistical modeling community, a practical and well-known fix for this problem is to add a “nugget” [11], i.e., a small perturbation $c_0 \mathbf{I}$ (where c_0 is a small positive constant, usually chosen to be 10–100 times the floating-point machine precision) is added to $\mathbf{K}$, where $\mathbf{I}$ is the identity matrix. Unfortunately, this trick does not resolve the issue in the desired way because it often results in less accurate data predictions in GP. In the RBF community, the trade-off principle also occurs in terms of the shape parameters of the RBFs and it has been investigated by several authors [15, 17, 22, 49]. More recent studies [3, 21, 29] have focused on finding their optimal values of the shape parameters in the combined ENO/WENO-RBF framework. On the other hand, in a series of studies by Wright and Fornberg [16, 71], the authors proposed an

alternative way called Contour-Padé algorithm of evaluating RBF approximation methods. The approach circumvents the direct evaluation of the RBF functions (RBF-Direct) which suffers from severely ill-conditioned linear systems due to large condition numbers when the associated RBF functions become flat. Recently, a new improved Contour-Padé algorithm has been proposed by using rational approximations of vector-valued analytic functions [72]. Certainly, the Contour-Padé approach can be employed in modeling our GP approach to improve the analogous situations of ill-conditioned GP covariance function matrices in Eqs. (3) and (17). A careful investigation of which will be discussed in our forthcoming papers. Our current strategy of this issue is given in Sect. 5.2.

The SE covariance function has two desirable properties [4, 11, 48]. First, it has the property of having a native space of C^∞ functions, which implies that the resulting interpolants converge exponentially with stencil size, for data sampled from smooth functions. Second, the SE kernel facilitates dimensional factorization, which is useful in multidimensional cases.

On the negative side, SE performs poorly when the dataset contains discontinuities, such as shocks and contact discontinuities in a CFD context. An intuitive resolution to this issue, from a statistical modeling perspective, would be to use other types of non-smooth kernel functions instead such as the Matérn class, exponential-types, rational quadratic functions, Wendland, etc. [4, 48, 59]. They are known to be better-suited for discontinuous datasets than SE by relaxing the strong smoothness assumptions of SE. By construction, the latent (or prior) sample function spaces generated by these non-smooth kernels possess finite-order of differentiability. As a result, we have seen that the posterior GP predictions of interpolations and reconstructions using such kernels turn out to be very noisy and rough, and their solutions often suffer from reduced solution-accuracy and numerical instability. A covariance kernel function such as the Gibbs covariance or the neural-network covariance [48] would be more appropriate for resolving discontinuities. The possibility of using these GP kernels for high-order interpolations and reconstructions is under investigation and will be studied in our future work. To deal with discontinuous solutions in this paper, we introduce yet another new approach to design an improved GP reconstruction algorithm, termed as “GP-WENO”, which is described in Sect. 2.5. This new approach formulates a new set of non-polynomial, GP-based smoothness indicators for discontinuous flows.

Although we are interested in describing one-dimensional formulations of GP in this paper, we first discuss a 2D case that best illustrates a general strategy of GP to compute a single interpolation and reconstruction procedure. Consider a list of N cells I_i , $i = 1, \dots, N$ in 2D. The most natural way to construct the list of cells (i.e., the stencil) over which the GP interpolation/reconstruction will take place is to pick a radius¹ R , and add to the list those cells I_{i_k} whose cell centers $\mathbf{x}_{i_k}$ are within the distance R from a local cell I_{i_0} under consideration. The result is a “blocky sphere” that ensures isotropy of the interpolation. See Fig. 1. We can adjust R to regulate N , thus making R a performance/accuracy tradeoff tuning parameter.

As mentioned, an important practical feature of the SE covariance function is that it provides its dimensional factorization analytically. Therefore, the volume averages in Eqs. (14) and (15) simplify to iterated integrals, and in fact, they can be expressed *analytically* in terms of a pre-computed list of error functions of arguments proportional to one-dimensional cell center differences. Eqs. (15) and (16) become

¹ In the current study R is an integer multiple of Δ for simplicity, which needs not be the case in general.

Fig. 1 An example of a 2D GP stencil with a radius R . It consists of a collection grid cells I_{i_k} (blue cells) that are included in a multidimensional blocky sphere of a radius R from the local cell I_{i_0} , in which the GP interpolation/reconstruction takes place (Color figure online)

$$\begin{aligned} C_{kh} = & \sqrt{\pi} \left(\frac{\ell}{\Delta} \right)^2 \left\{ \left(\frac{\Delta_{kh} + 1}{\sqrt{2}\ell/\Delta} \operatorname{erf} \left[\frac{\Delta_{kh} + 1}{\sqrt{2}\ell/\Delta} \right] + \frac{\Delta_{kh} - 1}{\sqrt{2}\ell/\Delta} \operatorname{erf} \left[\frac{\Delta_{kh} - 1}{\sqrt{2}\ell/\Delta} \right] \right) \right. \\ & + \frac{1}{\sqrt{\pi}} \left(\exp \left[-\frac{(\Delta_{kh} + 1)^2}{2(\ell/\Delta)^2} \right] + \exp \left[-\frac{(\Delta_{kh} - 1)^2}{2(\ell/\Delta)^2} \right] \right) \\ & \left. - 2 \left(\frac{\Delta_{kh}}{\sqrt{2}\ell/\Delta} \operatorname{erf} \left[\frac{\Delta_{kh}}{\sqrt{2}\ell/\Delta} \right] + \frac{1}{\sqrt{\pi}} \exp \left[-\frac{\Delta_{kh}^2}{2(\ell/\Delta)^2} \right] \right) \right\}, \end{aligned} \quad (22)$$

and

$$\mathbf{T}_{*,k} = \sqrt{\frac{\pi}{2}} \frac{\ell}{\Delta} \left\{ \operatorname{erf} \left[\frac{\Delta_{k*} + 1/2}{\sqrt{2}\ell/\Delta} \right] - \operatorname{erf} \left[\frac{\Delta_{k*} - 1/2}{\sqrt{2}\ell/\Delta} \right] \right\}, \quad (23)$$

using the SE kernel, where $\Delta_{kh} = (x_k - x_h)/\Delta$. Other choices of covariance kernel functions are also available [11, 59], while such choices would require numerical approximations for Eqs. (16) and (15), rather than a possible analytical form as shown in Eqs. (22) and (23) using SE.

2.5 GP-WENO: New GP-Based Smoothness Indicators for Non-Smooth Flows

For smooth flows the GP linear prediction in Eq. (17) with the SE covariance kernel Eq. (21) furnishes a high-order GP reconstruction algorithm without any extra controls on numerical stability. However, for non-smooth flows, the unmodified GP-SE reconstruction suffers from unphysical oscillations that originate at discontinuities such as shocks. To handle such issues with non-smooth flows, we adopt the principal idea of employing the nonlinear weights in the Weighted Essentially Non-oscillatory (WENO) methods [26], by which we adaptively change the size of the reconstruction stencil to avoid interpolating through a discontinuity, while retaining high-order properties in smooth flow regions. A traditional WENO takes the weighted combination of candidate stencils based on the local smoothness of the individual sub-stencils. The weights are chosen so that they are optimal in smooth regions in the sense that they are equivalent to an approximation using the global stencil that is the union of the candidate sub-stencils.

Fig. 2 A schematic example of the 5-point GP stencil with $R = 2$ centered at x_i . Two grid locations indicated as x_* are the target locations where we seek for high-order GP approximations of pointwise value (e.g., ρ_*) using the given cell-centered volume average data in the 5-point stencil neighborhood (e.g., $\langle \rho_s \rangle$, $s = i - 2, \dots, i + 2$)

For a stencil S_R of a radius R , with $2R + 1$ points centered at the cell I_i , we consider $R + 1$ candidate sub-stencils $S_m \subset S_R$, each with $R + 1$ points. Similar to the traditional WENO schemes, we subdivide S_R which is given as

$$S_R = \bigcup_{m=i-R}^{i+R} I_m \quad (24)$$

into $R + 1$ sub-stencils S_m ,

$$S_m = \{I_{i-R+m-1}, \dots, I_i, \dots, I_{i+m-1}\}, \quad m = 1, \dots, R + 1. \quad (25)$$

They satisfy (Fig. 2)

$$S_R = \bigcup_{m=1}^{R+1} S_m \text{ and } I_i = \bigcap_{m=1}^{R+1} S_m. \quad (26)$$

Using Eq. (17) on each sub-stencil S_m , the GP approximation for a pointwise value at a cell location x_* from the m -th sub-stencil S_m is

$$\tilde{f}_*^m = f_0 + \mathbf{z}_m^T (\mathbf{G}_m - f_0 \mathbf{1}_{R+1}). \quad (27)$$

On the other hand, from the global $2R + 1$ stencil S_R we also have a pointwise GP approximation at the same location

$$\tilde{f}_* = f_0 + \mathbf{z}^T (\mathbf{G} - f_0 \mathbf{1}_{2R+1}). \quad (28)$$

Here, the cell location “*” is where we want to obtain the desired high-order reconstruction from GP, e.g., the cell interface $x_* = x_{i \pm 1/2}$ is of particularly of interest to the Riemann problems in our case. Furthermore, f_0 is a constant mean function that is the same over all the candidate sub-stencils, $\mathbf{z}_{*,m}^T = \mathbf{T}_{*,m}^T \mathbf{C}_m^{-1}$ is the $(R + 1)$ -dimensional vector of weights, $\mathbf{G}_m$ is the $(R + 1)$ -dimensional vector of volume averaged data in either primitive, conservative, or characteristic variables on S_m , and $\mathbf{1}_{R+1} = [1, \dots, 1]^T$ is the $(R + 1)$ -dimensional one-vector. Likewise, $\mathbf{z}^T = \mathbf{T}_*^T \mathbf{C}^{-1}$, $\mathbf{G}$, and $\mathbf{1}_{2R+1}$ are $(2R + 1)$ -dimensional vectors defined in the same way. We choose $\mathbf{G}$ and $\mathbf{G}_m$ the volume averaged quantities in characteristic variables with the use of eigen-decompositions that help minimize oscillations near shocks and discontinuities.

We now take the weighted combination of these GP approximations as our final reconstructed value

$$\tilde{f}_* = \sum_{m=1}^{R+1} \omega_m \tilde{f}_*^m. \quad (29)$$

As in the traditional WENO approach, the weights ω_m should reduce to some optimal weights γ_m in smooth regions such that the approximation in Eq. (29) gives the GP approximation over the global $2R + 1$ point stencil. The γ_m 's then should satisfy

$$\mathbf{z}^T (\mathbf{G} - f_0 \mathbf{1}_{2R+1}) = \sum_{m=1}^{R+1} \gamma_m \mathbf{z}_m^T (\mathbf{G}_m - f_0 \mathbf{1}_{R+1}), \quad (30)$$

where $\boldsymbol{\gamma} = [\gamma_1, \dots, \gamma_{R+1}]^T$ is given by the solution to the $(2R+2) \times (R+1)$ overdetermined system

$$\left(\sum_{m=1}^{R+1} \mathbf{M} f_0 \mathbf{z}_m^T \mathbf{1}_{R+1} \right) \boldsymbol{\gamma} = \left(f_0 \mathbf{z}^T \mathbf{1}_{2R+1} \right). \quad (31)$$

The k -th column of the matrix $\mathbf{M}$ is given by $\mathbf{z}_k$ for $(R+1)$ entries and zeros for the rest:

$$\mathbf{M}_{m,k} = \begin{cases} \mathbf{z}_{m,k} & \text{if } I_m \in S_k, \\ 0 & \text{otherwise.} \end{cases} \quad (32)$$

For example, in case with $f_0 = 0$ and $R = 2$, the above system reduces to the following 5×3 overdetermined system,

$$\gamma_1 \begin{bmatrix} z_{1,1} \\ z_{2,1} \\ z_{3,1} \\ 0 \\ 0 \end{bmatrix} \Bigg\} \mathbf{z}_1 + \gamma_2 \begin{bmatrix} 0 \\ z_{1,2} \\ z_{2,2} \\ z_{3,2} \\ 0 \end{bmatrix} \Bigg\} \mathbf{z}_2 + \gamma_3 \begin{bmatrix} 0 \\ 0 \\ z_{1,3} \\ z_{2,3} \\ z_{3,3} \end{bmatrix} \Bigg\} \mathbf{z}_3 = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix} \Bigg\} \mathbf{z}, \quad (33)$$

or in a matrix form, $\mathbf{M}\boldsymbol{\gamma} = \mathbf{z}$,

$$\begin{bmatrix} z_{1,1} & 0 & 0 \\ z_{2,1} & z_{1,2} & 0 \\ z_{3,1} & z_{2,2} & z_{1,3} \\ 0 & z_{3,2} & z_{2,3} \\ 0 & 0 & z_{3,3} \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ z_4 \\ z_5 \end{bmatrix}. \quad (34)$$

The optimal weights γ_m then depend on the choice of the kernel function, but as with the vectors of GP weights, $\mathbf{z}$ and $\mathbf{z}_m$, γ_m need only be computed once. We take the γ_m 's as the least squares solution to the overdetermined system in Eq. (31), which can be determined numerically.

It remains to describe the nonlinear weights ω_m in Eq. (29). Again, they should reduce to the optimal weights γ_m in smooth regions, but more importantly, they need to serve as an indicator of the quality of data on the sub-stencil S_m . We adopt the same weights as in the WENO-JS methods [26]:

$$\omega_m = \frac{\tilde{\omega}_m}{\sum_s \tilde{\omega}_s}, \text{ where } \tilde{\omega}_m = \frac{\gamma_m}{(\epsilon + \beta_m)^p}, \quad (35)$$

where we set $p = 1$ and $\epsilon = 10^{-36}$ for our test results. These weights are based on the so-called smoothness indicators β_m , $m = 1, \dots, R + 1$, for each sub-stencil S_m . In modern

WENO schemes, the smoothness indicators β_m are calculated as a scaled sum of the square L_2 norms of all the derivatives of the reconstructed polynomial on the relevant interval I_i .

Let us now construct a new class of smoothness indicators β_m that fit within the GP framework. For now, let us use the notation ‘ m ’ to denote the objects restricted on each S_m . Consider an $(R + 1)$ -dimensional sample of cell-centered pointwise values

$$\mathbf{f}^m = [q(x_1), \dots, q(x_{R+1})]^T. \quad (36)$$

We can express $\mathbf{f}^m$ in the basis $\mathbf{v}_i^m$ as

$$\mathbf{f}^m = \sum_{i=1}^{R+1} \alpha_i^m \mathbf{v}_i^m, \text{ for each } m = 1, \dots, R + 1, \quad (37)$$

where $\mathbf{v}_i^m$ are eigenvectors of the $(R + 1) \times (R + 1)$ sub-matrix $\mathbf{K}^m$ with eigenvalues λ_i^m such that

$$\mathbf{K}^m \mathbf{v}_i^m = \lambda_i^m \mathbf{v}_i^m, \quad i = 1, \dots, R + 1. \quad (38)$$

Each element $q(x_l)$, $l = 1, \dots, R + 1$, can be reconstructed from $\mathbf{G}_m$, using a zero mean $f_0 = 0$ in Eq. (20), as

$$q(x_l) = \mathbf{z}_m^T(x_l) \mathbf{G}_m, \quad (39)$$

where $(R + 1)$ -dimensional vector $\mathbf{z}_m^T(x_l) = \mathbf{T}_{l,k}^m (\mathbf{C}^m)^{-1}$, $k = 1, \dots, R + 1$, and

$$\mathbf{G}_m = [\langle q_1 \rangle, \dots, \langle q_{R+1} \rangle]^T. \quad (40)$$

Here $\mathbf{T}_{l,k}^m$ is the prediction vector (see Eq. (23)) over the sub-stencil S_m evaluating the covariance between the cell average values $\langle q_k \rangle$, $k = 1, \dots, R + 1$, and the pointwise value q_l evaluated at each $x_l \in S_m$.

Obtaining α_i^m is straightforward from Eq. (37). Using $(\mathbf{v}_i^m)^T \mathbf{v}_j^m = \delta_{ij}$,

$$\alpha_i^m = (\mathbf{v}_i^m)^T \mathbf{f}^m. \quad (41)$$

Finally, for each m , we have our new GP-based smoothness indicators β_m for the data on the sub-stencil S_m given by the norm $\|\mathbf{f}^m\|_{\mathcal{H}}^2$ in the Hilbert space of functions in the reproducing kernel Hilbert space (RKHS) defined by the kernel function $k(x, x')$ (see Eq. 68 in “Appendix C”):

$$\beta_m = \|\mathbf{f}^m\|_{\mathcal{H}}^2 = \langle \mathbf{f}^m, \mathbf{f}^m \rangle_{\mathcal{H}} = \sum_{i=1}^{R+1} \frac{(\alpha_i^m)^2}{\lambda_i^m}. \quad (42)$$

As discussed in “Appendix C”, the new GP-based smoothness indicators, defined in Eq. (42), become small on a data set $\mathbf{f}^m$ that represents a smoothly varying profile of pointwise values $q(x_l)$, while they become large on a rapidly varying data set. Thereby, they can naturally serve the role of adjusting $\tilde{\omega}_m$ in Eq. (35) in a nonlinear way depending on the level of smoothness indicated from the data. Note in Eq. (38) that the sub-matrix $\mathbf{K}^m$ is the *pointwise* covariance kernel (e.g., see Eq. 21) rather than the *integrated* covariance kernel (e.g., see Eq. (15)). In this way, we obtain the eigenpair $(\lambda_i^m, \mathbf{v}_i^m)$ that is consistent with providing β_m which is to measure smoothness of *pointwise* values $\tilde{f}_*^m = \tilde{f}^m(x_*) = \tilde{f}^m(x_{i \pm \frac{1}{2}})$ in Eq. (29). This completes our description of devising a new GP-based smoothness indicators.

The computational efficiency of Eqs. (39) and (41) can be improved by introducing vectors $\mathbf{P}_i^m$ defined as

$$\mathbf{P}_i^m = (\mathbf{v}_i^m)^T \cdot \mathbf{Z}_m^T, \quad (43)$$

where the l -th column of the matrix $\mathbf{Z}_m$ is given by $\mathbf{z}_m(x_l)$, $l = 1, \dots, R + 1$. In this way, we precompute $\mathbf{P}_i^m$ which can directly act on the volume average quantities of $\mathbf{G}_m$, by which the α_i^m are given by

$$\alpha_i^m = \mathbf{P}_i^m \cdot \mathbf{G}_m. \quad (44)$$

To see the computational efficiency in comparison, in “Appendix D”, we provide operation counts of the two approaches referred to as “Option 1” with solving Eqs. (39) and (41), and “Option 2” with solving Eqs. (43) and (44). Our analysis shows that Option 2 is twice faster than Option 1 in terms of the associated operation counts. Option 2 was used for the results shown in Sect. 5.

It is noteworthy that there is an alternative interpretation of Eq. (42) available in terms of the log likelihood function. Notice that Eq. (42) can be written in a quadratic form

$$(\mathbf{f}^m)^T (\mathbf{K}^m)^{-1} \mathbf{f}^m, \quad (45)$$

which is again equivalent to

$$-2 \ln P(\mathbf{f}^m) + C, \quad (46)$$

assuming $f_0 = 0$ in the relation in Eq. (2) and $C = \ln(\det |\mathbf{K}^m|^{-1}) + N \ln(2\pi)$. The analogy between Eqs. (42) and (45) can be proved easily if we realize that the matrix $\mathbf{K}^m$ can be expressed as

$$\mathbf{K}^m = \sum_{i=1}^{R+1} \lambda_i^m \mathbf{v}_i^m (\mathbf{v}_i^m)^T, \quad (47)$$

and hence

$$(\mathbf{K}^m)^{-1} = \sum_{i=1}^{R+1} \frac{1}{\lambda_i^m} \mathbf{v}_i^m (\mathbf{v}_i^m)^T. \quad (48)$$

The result follows easily by plugging Eqs. (37) and (48) into Eq. (45) to get Eq. (42). So the statement that β_m is large in this sense is analogous to the statement that the likelihood of $\mathbf{f}^m$ is small. The statistical interpretation of this analogy means that a short length-scale variability in $\mathbf{f}^m$ makes its data unlikely according to the smoothness model represented by $\mathbf{K}^m$.

It was observed by Jiang and Shu [26] that it is critically important to have appropriate smoothness indicators so that in smooth regions the nonlinear weights reproduce the optimal linear weights to the design accuracy in order to maintain the high-order of accuracy in smooth solutions. Typically this is shown by Taylor expansion of smoothness indicators in Eq. (35). Here we propose to calculate the eigendecomposition in Eq. (65) numerically, and therefore we don’t provide a mathematical proof of the accuracy of our new smoothness indicators. Nonetheless, we observe the expected high order convergence expected from the polynomial WENO reconstructions on the equivalent stencils in numerical experiments provided in Sect. 5.2.

3 Step-by-Step Procedures of the GP-SE Algorithm for 1D FVM

Before we present the numerical results of the GP reconstruction algorithms outlined so far, we give a quick summary of the step-by-step procedures for 1D simulations with FVM. The 1D algorithm of GP can be summarized as below:

1. **Pre-Simulation:** The following steps are carried out before starting a simulation, and any calculations therein need only be performed *once, stored, and used* throughout the actual simulation.
 - (a) **Configure computational grid:** Determine a GP stencil radius R as well as choose the size of the hyperparameter ℓ . This determines the SE kernel function in Eq. (21) as well as the global and candidate stencils in Eqs. (24) and (25).
 - (b) **Compute GP weights:** Compute the covariance matrices, $\mathbf{C}$ and $\mathbf{C}^m$ (Eq. 15) on the stencils S_R and each of S_m , respectively. Compute the prediction vectors, $\mathbf{T}_*$ and $\mathbf{T}_*^m$ respectively on S_R and each of S_m (Eq. 16). The GP weight vectors, $\mathbf{z}^T = \mathbf{T}_* \mathbf{C}^{-1}$ on S_R and $\mathbf{z}_m^T = \mathbf{T}_*^m (\mathbf{C}^m)^{-1}$ on each of S_m , can then be stored for use in the GP reconstruction. The columns of the matrices $\mathbf{Z}_m$, $\mathbf{z}_m(x_l)$, should be computed here for use in step (d).
It is crucial in this step as well as in Step (c) to use the appropriate floating point precision to prevent the covariance matrices, $\mathbf{C}$ and $\mathbf{C}^m$, from being numerically singular. This is discussed in more detail in Sect. 5.2. We find that double precision is only suitable up to condition numbers $\kappa \sim 10^8$, whereas quadruple precision comfortably allows up to $\kappa \sim 10^{18}$ (hence our choice in this paper). The standard double precision is used except for Step (b) and Step (c).
 - (c) **Compute linear weights:** Use the GP weight vectors to calculate and store the optimal linear weights γ_m according to Eq. (31).
 - (d) **Compute kernel eigensystem:** The eigensystem for the covariance matrices used in GP-WENO are calculated using Eq. (65). The matrices, $\mathbf{C}^m$, are the same on each of the candidate stencils in the GP-WENO scheme presented here, so only one eigensystem needs to be determined. This eigensystem is then used to calculate and store the vectors $\mathbf{P}_i^{(m)}$ in Eq. (43) for use in determining the smoothness indicators in the reconstruction Step 2.

At this point, before beginning the simulation, if quadruple precision was used in Step (b) and Step (c), the GP weights and linear weights can be truncated to double precision for use in the actual reconstruction step.

2. **Reconstruction:** Start a simulation. Choose f_0 according to Eq. (62) or simply set to zero. The simplest choice $f_0 = 0$ gives good results in practice and yields $\bar{\mathbf{G}} = 0$ in Eq. (17). At each cell x_i , calculate the updated posterior mean function $\tilde{f}_*^m$ according to Eq. (17) as a high-order GP reconstructor to compute high-order pointwise Riemann state values at $x_* = x_{i \pm 1/2}$ using each of the candidate sub-stencils S_m . The smoothness indicators (Eq. 42), calculated using the eigensystem from Step (d) in conjunction with the linear weights from Step (c), form the nonlinear weights (Eq. 35). Then take the convex combination according to Eq. (29).
3. **Calculate fluxes:** Solve Riemann problems at cell interfaces using the high-order GP Riemann states in Step 2 as inputs.
4. **Temporal update:** Update the volume-averaged solutions $\langle q_i \rangle$ from t^n to t^{n+1} using the Godunov fluxes from Step 3.

4 The 1D GP Source Code on GitHub

The full 1D GP source code described in this paper is available at https://github.com/acreyes/GPID_JSC. The source code is licensed under a Creative Commons Attribution 4.0 International License.

Table 1 Shown is the relative time to solution for the four methods considered, all normalized to the GP-R2 time

Scheme	Speedup
GP-R1	0.8
GP-R2	1.0
GP-R3	1.4
WENO-JS	0.9

5 Numerical Results

In this section, we present numerical results using the GP-WENO reconstructions described in Sect. 2.5 with stencil radii $R = 1, 2, 3$ (denoted as GP-R1, GP-R2, GP-R3 respectively), applied to the 1D compressible Euler equations and the 1D equations of ideal magnetohydrodynamics (MHD). The GP-WENO method will be the default reconstruction scheme hereafter.

We compare the solutions of GP-WENO with the fifth-order WENO (referred to as WENO-JS in what follows) FVM [26,55], using the same nonlinear weights in Eq. (35). The only difference between using WENO-JS and GP-R2 is the use of polynomial based reconstruction and smoothness indicators [55] for WENO-JS, and the use of Gaussian process regression with the new GP-based smoothness indicators for GP. All reconstructions are carried out in characteristic variables to minimize unphysical oscillations in the presence of discontinuities. That is, characteristic variables are used in the data vector, $\mathbf{G}_m$ in Eq. (27). A fourth-order TVD Runge-Kutta method [54] for temporal updates and the HLLC [38,63] or Roe [51] Riemann solvers are used throughout.

5.1 Performance Comparison

We first present a performance comparison for the proposed GP scheme and the WENO-JS scheme summarized in Table 1. We see that the cost of the GP-R2 and WENO-JS methods are very nearly the same, operating on the same stencils. The minor difference can be attributed to the new GP-based smoothness indicators. It can be seen in Eq. (42) the GP smoothness indicators can be written as the sum of $R + 1$ perfect squares, which involves calculations of three terms for each β_m for GP-R2. Meanwhile, the smoothness indicators for WENO-JS involves calculations of only two terms, i.e., the first and second derivatives of the second-degree ENO polynomials $p_m(x)$. The additional cost is offset by a significant advantage of the GP-WENO solution over the WENO-JS solution in smooth regions near discontinuities, which is discussed in Sect. 5.3.

5.2 1D Smooth Advection

The test considered here involves the passive advection of a Gaussian density profile. We initialize a computational box on $[0, 1]$ with periodic boundary conditions. The initial density profile is defined by $\rho(x) = 1 + e^{-100(x-x_0)^2}$, with $x_0 = 0.5$, with constant velocity, $u = 1$, and pressure, $P = 1/\gamma$. The specific heat ratio is chosen to be $\gamma = 5/3$. The resulting profile is propagated for one period through the boundaries. At $t = 1$, the profile returns to its initial position at $x = x_0$, any deformation of the initial profile is due to either phase errors or numerical diffusion. We perform this test using a length hyperparameter of $\ell = 0.1$ for stencil radii $R = 1, 2, 3, 4$ and 5, with a fixed Courant number, $C_{\text{eff}} = 0.8$ and vary the resolution of computational box, with $N = 32, 64, 128, 256$ and 512.

The results of this study are shown in Fig. 3a. From these numerical experiments, GP reconstruction shows a convergence rate that goes as the size of the stencil, $2R + 1$. Note that this convergence rate $2R + 1$ of GP is equivalent to the convergence rate $2r - 1$ of a classic WENO method for the same stencil size. In the classic WENO method, the notation r represents the order (not degree) of polynomials on S_m which require r cells on each S_m , totaling $2r - 1$ cells on S_R . This implies $2r - 1 = 2R + 1$, or $R = r - 1$. For example, GP-R2 converges at fifth-order on the stencil $S_R = S_2$ that has five cells, so does WENO-JS with three third-order ENO polynomials (i.e., $r = 3$) on the same 5-point stencil.

In Fig. 3a, the L_1 error plateaus out at $\sim 10^{-12}$ which is a few orders of magnitude greater than the standard IEEE double-precision, $\sim 10^{16}$. This happens because at high resolution the length hyperparameter, ℓ , becomes very large relative to the grid spacing, Δ . The covariance matrix, $\mathbf{C}$ given in Eq. (22) becomes nearly singular in the regime $\ell/\Delta \gg 1$, yielding very large condition numbers for $\mathbf{C}$. We find the plateau in the L_1 error occurs for condition numbers, $\kappa \sim 10^{18}$, corresponding to the point where the errors in inverting $\mathbf{C}$ in Eq. (17) begin to dominate. This implies that the choice of floating-point precision has an immense impact on the possible ℓ/Δ , and a proper floating-point precision needs to be chosen in such a way that the condition number errors do not dominate. As mentioned in Sect. 2.4, SE suffers from singularity when the size of the dataset grows. The approach that enabled us to produce the results in Fig. 3 is to utilize quadruple-precision *only* for the calculation of $\mathbf{z}^T = \mathbf{T}_*^T \mathbf{C}^{-1}$ in Eq. (17). Otherwise, the plateau would appear at a much higher L_1 error $\sim 10^{-7}$ with double-precision, producing undesirable outcomes for all forms of grid convergence studies. This corresponds to condition numbers $\kappa \sim 10^8$, and as a point of reference, the breakdown starts to occur for $\ell/\Delta > 48$ using a GP radius $R = 2$. The weight vector $\mathbf{z}^T$ needs to be calculated only once, before starting the simulation. It can then be saved and truncated to double-precision for use in the actual reconstruction procedure. There are only four related small subroutines that need to be compiled with quadruple-precision in our code implementation. The overall performance is not affected due to this extra precision handling which is necessary for the purpose of a grid convergence study from the perspective of CFD applications.

The GP's performance is displayed in Fig. 3b in terms of CPU time per a given error. The GP scheme is computed with different radii, $R = 1, 2$ and 3 , denoted as GP-R1, GP-R2 and GP-R3, respectively. The GP performances are compared with the three most popular piecewise polynomial methods including the 2nd-order PLM, the 3rd-order PPM, and the 5th-order WENO-JS. It is observed that even though GP-R2 and WENO-JS are equivalent in terms of their nominal 5th-order accuracy, GP-R2 always reaches a given error (e.g., the dotted lines show CPU times to reach the error $\sim 10^{-4}$) 30% faster than WENO-JS. Compared to WENO-JS, the 7th-order GP-R3 is 60% faster, while the 3rd-order GP-R1 is 170% slower. The 3rd-order PPM is far slower than WENO-JS by 1170%. The 2nd-order PLM is just too slow to be evaluated in this comparison test.

Lastly, the correlational length hyperparameter ℓ provides an additional avenue to tune solution accuracy that is not present in polynomial-based methods. Fig. 3c shows how the GP errors with $R = 2$ in the smooth-advection problem changes with the choice of ℓ , compared with the error from a fifth-order WENO-JS + RK4 solution (denoted in dotted lines). At large ℓ the errors become roughly the same as in WENO-JS, while at small ℓ the errors become worse. The error finds a minimum at a value of ℓ near the half-width of the Gaussian density profile. This is in line with the idea that the optimal choice of ℓ should match the physical length scale of the feature being resolved. We will report new strategies on this topic in our future papers.

Fig. 3 **a** Convergence for smooth Gaussian density advection using different GP stencil radii, all using the length scale $\ell = 0.1$ and the HLLC Riemann solver with $C_{\text{eff}} = 0.8$. The black dotted lines show $2R + 1$ convergence rates, ranging from 3rd-order ($R = 1$) to 11th-order ($R = 5$). The red dotted line represents a plateau at an L_1 of 10^{-12} where a large condition number of the covariance matrix is obtained and no further accuracy is achievable. **b** L_1 errors versus CPU time. **c** L_1 errors as a function of the hyperparameter ℓ , using $R = 2$. The dotted lines are the errors of the fifth-order WENO-JS method on the same stencil (Color figure online)

Fig. 4 **a** The Shu-Osher problem. Density profiles at $t = 1.8$ computed using three different GP stencil radii ($R = 1, 2, 3$) on a 200 grid resolution with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 6$ and the HLLC Riemann solver. The reference solution is obtained using WENO-JS on a 2056 grid resolution. **b** A close-up view of **a** in the post-shock region (Color figure online)

5.3 1D Shu-Osher Shock Tube Problem

The second test is the Shu-Osher problem [56] to test GP's shock-capturing capability as well as to see how well GP can resolve small-scale features in the flow. The test gives a good indication of the method's numerical diffusivity and has been a popular benchmark to demonstrate numerical errors of a given method. In this problem, a (nominally) Mach 3 shock wave propagates into a constant density field with sinusoidal perturbations. As the shock advances, two sets of density features appear behind the shock. One set has the same spatial frequency as the unshocked perturbations, while in the second set the frequency is doubled and follows more closely behind the shock. The primary test of the numerical method is to accurately resolve the dynamics and strengths of the oscillations behind the shock.

The results are shown in Fig. 4. The solutions are calculated at $t = 1.8$ using a resolution of $N = 200$ and are compared to a reference solution resolved on $N = 2056$. It is evident that the GP solution using $R = 3$ provides the least diffusive solution of the methods shown, especially in capturing the amplitude of the post-shock oscillations in Fig. 4b. Of the two

Fig. 5 Closeup view of the post-shock features as in Fig. 4b using $N = 200$ for WENO-JS and GP-R2, **a** with $l/\Delta = 3$, **b** with $l/\Delta = 6$, and **c** with $l/\Delta = 50$. The reference solution is calculated using WENO-JS on 2056 grid resolution

Fig. 6 The Sod shock tube problem at $t = 0.2$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the HLLC Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

fifth-order methods, GP-R2 and WENO-JS, the GP solution has a slightly better amplitude in the post-shock oscillations compared to the WENO-JS solution, consistent with what is observed in Sect. 5.2 for the smooth advection problem.

The results in Sect. 5.2 suggest that the choice of ℓ should correspond to a length scale characteristic of the flow for optimal performance. Figure 5 compares the post-shock features on the same grid for the WENO-JS method and the GP-R2 method with $\ell/\Delta = 3, 6$ and 50. Here $\ell/\Delta = 3$ is roughly a half wavelength of the oscillations, and the GP-R2 method clearly gives a much more accurate solution compared to the WENO-JS solution. Just as can be seen in Fig. 3c, ℓ/Δ much larger than the characteristic length yields a solution much closer to that of WENO-JS. From Fig. 3c, we expect that the GP reconstructions in Eq. (27) should approach the WENO-JS reconstructions in smooth regions for $\ell \gg \Delta$. However, we see that the new GP-based smoothness indicators in Eq. (42) allow the amplitudes near the shock to be better resolved. This reflects a key advantage of the proposed GP method over polynomial-based high-order methods. The additional flexibility in GP, afforded by the kernel hyperparameters, allows its solution accuracy to be tuned to the features that are being resolved. Only at larger values of ℓ does the model become fully constrained by the data in GP, whereas the interpolating polynomials used in a classical WENO are always fully constrained by design. Analogously in the RBF theory, the shape parameters ϵ_j plays an important role in the similar context of accuracy and convergence. Several strategies have been studied recently for solving hyperbolic PDEs [3, 21, 29].

Fig. 7 The Einfeldt strong rarefaction test at $t = 0.15$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{cfl}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the HLLC Riemann solver. Solid lines show the exact solution

5.4 The Sod Shock Tube Test

The shock tube problem of Sod [57] has been one of the most popular 1D tests of a code's ability to handle shocks and contact discontinuities. The initial conditions, on the domain $[0, 1]$, are given by the left and right states

$$(\rho, u, p) = \begin{cases} (1, 0, 1), & x < 0.5, \\ (0.125, 0, 0.1), & x > 0.5, \end{cases} \quad (49)$$

with the ratio of specific heats $\gamma = 1.4$. Outflow boundary conditions are imposed at $x = 0$ and $x = 1$. The results are shown in Fig. 6.

All of the GP-WENO schemes correctly predicts the nonlinear characteristics of the flow including the rarefaction wave, contact discontinuity, and the shock. The solution using GP-R2 is very comparable to the WENO-JS solution using the same 5-point stencil. As expected, the GP-R1 solution smears out the most at both the shock and the contact discontinuity, and at the head and tail of the rarefaction. The 7th order GP-R3 also successfully demonstrates that its shock solution is physically correct without triggering any unphysical oscillation. Somewhat counter-intuitively from the perspective of 1D polynomial schemes, the smallest stencil GP-R1 shows the most oscillations near the shock. This happens because the eigendecomposition of the kernel in Eq. (65) used in the calculation the smoothness indicators for Eq. (42) better approximates $\|\mathbf{f}^m\|_{\mathcal{H}}^2$ as the size of the stencil increases. All of the GP stencils use the same kernel function, and so $\|\mathbf{f}^m\|_{\mathcal{H}}^2$ and the smoothness indicators are calculated from the same

Fig. 8 The Brio-Wu shock tube at $t = 0.1$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the Roe Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

eigenspace of the kernel function. Recall that the eigenexpansion of the kernel function in Eq. (63) can contain an infinite number of eigenvalues, and so can $\|\mathbf{f}^m\|_{\mathcal{H}}^2$ and the sum in Eq. (42). The finite approximation to the eigensystem and subsequently $\|\mathbf{f}^m\|_{\mathcal{H}}^2$ then becomes better as the smallest coefficient α_i goes to zero. Then for GP, the β_m 's best indicate the smoothness on larger sized stencils, where the sum in Eq. (42) contains more terms in the infinite sum.

5.5 The Einfeldt Strong Rarefaction Test

First described by Einfeldt et al. [14], this problem tests how satisfactorily a code can perform in a low-density region in computing physical variables, ρ , u , p , ϵ , etc. Among those, the internal energy $\epsilon = p/(\rho(\gamma - 1))$ is particularly difficult to get right due to regions where the density and pressure are very close to zero. Hence the ratio of these two small values amplifies any small errors in both ρ and p , making the errors in ϵ the largest in general [62]. The large errors in ϵ are apparent for all schemes shown in Fig. 7, where the error is largest around $x = 0.5$. It can be observed that the amount of departure in ϵ from the exact solution (the cyan solid line) decreases as the GP radius R increases. The error in GP-R2 is slightly larger for ϵ at the center than in WENO-JS. However, the peak becomes considerably smaller in amplitude and becomes slightly flatter as R increases.

Fig. 9 The RJ1b MHD shock tube at $t = 0.03$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the Roe Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

5.6 Briou-Wu MHD Shock Tube

Brio and Wu [6] studied an MHD version of Sod's shock tube problem, which has become an essential test for any MHD code. The test has since uncovered some interesting findings, such as the compound wave [6], as well as the existence of non-unique solutions [64,65]. The results of this test are shown in Fig. 8.

All of the GP methods are able to satisfactorily capture the MHD structures of the problem. In all methods, including WENO-JS, there are some observable oscillations in the post-shock regions. Lee in [34] showed that these oscillations arise as a result of the numerical nature of the slowly-moving shock [70] controlled by the strength of the transverse magnetic field. As studied by various researchers [1,27,28,30,50,60], there seems no ultimate fix for controlling such unphysical oscillations due to the slowly-moving shock. Quantitatively, the level of oscillations differs in different choices of numerical methods such as reconstruction algorithms and Riemann solvers. We see that all of the GP solutions together with the WENO-JS solution feature a comparable level of oscillations. Except for GP-R1, all solutions also suffer from a similar type of distortions in u and B_y in the right going fast rarefaction. This distortion, as well as the oscillations due to the slowly-moving shock, seem to be suppressed in the 3rd order GP-R1.

Fig. 10 The RJ2a MHD shock tube at $t = 0.03$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the HLLC Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

5.7 Ryu and Jones MHD Shock Tubes

Ryu and Jones [52] introduced a large set of MHD shock tube problems as a test of their 1D algorithm, that are now informative to run as a code verification. In what follows, we will refer to the tests as RJ followed by the corresponding figure number from [52] in which the test can be found.

5.7.1 RJ1b Shock Tube

The first of the RJ shock tubes we consider is the RJ1b problem. This test contains a left going fast and slow shock, contact discontinuity as well as a slow and fast rarefaction. In Fig. 9 that all waves are resolved in the schemes considered.

5.7.2 RJ2a Shock Tube

The RJ2a test provides an interesting test due to its initial conditions producing a discontinuity in each of the MHD wave families. The solution contains both fast and slow

Fig. 11 The RJ2b MHD shock tube at $t = 0.2$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the Roe Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

left- and right-moving magnetoacoustic shocks, left- and right-moving rotational discontinuities and a contact discontinuity. Figure 10 shows that all three of the GP schemes are able to resolve all of these discontinuities. Again, we see that the smallest stencil GP-R1 solution contains some oscillations near the shock that are not present in the other GP solutions.

5.7.3 RJ2b Shock Tube

The RJ2b shock tube creates a set of a fast shock, a rotational discontinuity and a slow shock moving to the left away from a contact discontinuity, as well as a fast rarefaction, rotational discontinuity and slow rarefaction all moving to the right. What is of interest is that, since the waves propagate at almost the same speed, at $t = 0.035$ they have still yet to separate much. So, it is important to test a code's ability to resolve all of the discontinuities despite their close separation. The results of this test for the methods considered are shown in Fig. 11. At the shown resolution of $N = 128$, the contact discontinuity and the slow shock become somewhat smeared together in the GP-R1 solution, while they are better resolved by the other methods, even though there are only a couple of grid points distributed over the range of the features.

Fig. 12 The RJ4a MHD shock tube at $t = 0.15$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the HLLC Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

5.7.4 RJ4a Shock Tube

The RJ4a shock tube yields a fast and slow rarefaction, a contact discontinuity, a slow shock, and of particular interest, the switch-on fast shock. The feature of the switch-on shock is that the magnetic field is turned on in the region behind the shock. As can be seen in Fig. 12, all of the features including the switch-on fast shock are resolved in all methods. We see that GP-R1 smears out the solution not only in resolving discontinuous flow regions but also in resolving both fast and slow rarefaction waves.

5.7.5 RJ4b Shock Tube

The RJ4b test is designed to produce only a contact discontinuity and a fast right going switch-off fast rarefaction, where the magnetic field is zero behind the rarefaction. We can see in Fig. 13 that both the contact discontinuity and the switch-off rarefaction features are captured in all of the considered methods.

Fig. 13 The RJ4b MHD shock tube at $t = 0.15$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the HLLC Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

5.7.6 RJ5b Shock Tube

The RJ5b problem is of interest because it produces a fast compound wave, as opposed to the slow compound wave in the Brio-Wu problem, in addition to a left- and right-going slow shock, contact discontinuity and fast rarefaction. At the resolution of $N = 128$ used for the results in Fig. 14, the compound wave and one of the slow shocks are smeared together in all the methods tested.

6 Conclusion

We summarize key novel features of the new high-order GP approach presented in this paper.

The new GP approach utilizes the key idea from statistical theory of GP prediction to produce accurate interpolations of fluid variables in CFD applications. We have developed a new set of numerical strategies of GP for both smooth flows and non-smooth flows to numerically solve hyperbolic systems of conservation laws.

The GP methods presented here show an extremely fast rate of variable solution accuracy in smooth advection problems by controlling a single parameter, R . Further, the additional flexibility offered by the GP model approach over the fully constrained polynomial based

Fig. 14 The RJ5b MHD shock tube at $t = 0.16$. **a** GP with stencil radius $R = 1$, **b** GP with stencil radius $R = 2$, **c** GP with stencil radius $R = 3$, and **d** WENO-JS, all using 128 grid points with $C_{\text{eff}} = 0.8$. All GP calculations use $\ell/\Delta = 12$ and the HLLC Riemann solver. Solid lines show a reference solution computed using WENO-JS on 1024 grid points

model through the kernel hyperparameter ℓ allows for added tuning of solution accuracy that is not present in traditional polynomial based high-order methods. These parameters allow the GP method to demonstrate variable orders of method accuracy as functions of the size of the GP stencil and the hyperparameter ℓ (see Eq. 21) within a single algorithmic implementation.

The new GP-based smoothness indicators introduced here are used to construct nonlinear weights that give the essentially non-oscillatory property in discontinuous flows. The new smoothness indicators show a significant advantage over traditional WENO schemes in capturing flow features at the grid resolution near discontinuities.

The GP model, by design, can easily be extended to multidimensional GP stencils. Therefore, GP can seamlessly provide a significant algorithmic advantage in solving the multidimensional PDE of CFD. This “dimensional agnosticism” is unique to GP, and not a feature of polynomial methods. We will report our ongoing developments of GP in multiple spatial dimensions in forthcoming papers.

Acknowledgements This work was supported in part by the National Science Foundation under Grant AST-0909132; and in part by the U.S. DOE NNSA ASC through the Argonne Institute for Computing in Science under field work proposal 57789. The software used in this work was developed in part by funding from the U.S. DOE NNSA-ASC and OS-OASCR to the Flash Center for Computational Science at the University of Chicago.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

A Appendix: Gaussian Process Modeling

GP is a class of stochastic processes, i.e., processes that sample functions (rather than points) from an infinite dimensional function space. Initially one specifies a so-called *prior probability distribution* over the function space. Then, given a sample of function values at some set of points, one “trains” the model by regarding the sample as data and using Bayes’ theorem to update the probability distribution over the function space. This way, one obtains a data-informed *posterior probability distribution* over the function space, adjusted with respect to the prior so as to be compatible with the observed data. The posterior distribution may be used to *predict* (probabilistically) the value of the function at points where the function has not yet been sampled. The mean value of this GP prediction is our target interpolation/reconstruction for FDM and FVM.

Formally, a GP is a collection of random variables, any finite collection of which has a joint Gaussian distribution [4, 48]. A GP is fully defined by two functions:

- a mean function $\bar{f}(\mathbf{x}) = \mathbb{E}[f(\mathbf{x})]$ over $\mathbb{R}^N$, and
- a covariance function which is a symmetric, positive-definite integral kernel $K(\mathbf{x}, \mathbf{y})$ over $\mathbb{R}^N \times \mathbb{R}^N$.

Such functions f , drawn randomly from this distribution, are said to be sampled from a Gaussian Process with mean function $\bar{f}(\mathbf{x})$ and covariance function $K(\mathbf{x}, \mathbf{y})$, and we write $f \sim \mathcal{GP}(\bar{f}, K)$. As with the case of finite-dimensional Gaussian distributions, the significance of the covariance is

$$K(\mathbf{x}, \mathbf{y}) = \mathbb{E}[(f(\mathbf{x}) - \bar{f}(\mathbf{x}))(f(\mathbf{y}) - \bar{f}(\mathbf{y}))], \quad (50)$$

where the averaging is over the GP distribution.

In standard statistical modeling practice, both $\bar{f}(\mathbf{x})$ and $K(\mathbf{x}, \mathbf{y})$ are typically parametrized functions, with parameters controlling the character (e.g., length scales, differentiability, oscillation strength) of “likely” functions. Given a GP, and given N “training” points $\mathbf{x}_i$, $i = 1, \dots, N$ at which the function values $f(\mathbf{x}_i)$ are known, we may calculate the likelihood $\mathcal{L}$ (the probability of $\mathbf{f}$ given the GP model) of the data vector $\mathbf{f} \equiv [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ (e.g., N many pointwise values of density ρ at $\mathbf{x}_i$, $i = 1, \dots, N$) by

$$\mathcal{L} \equiv P(\mathbf{f}) = (2\pi)^{-\frac{N}{2}} \det |\mathbf{K}|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{f} - \bar{\mathbf{f}})^T \mathbf{K}^{-1} (\mathbf{f} - \bar{\mathbf{f}}) \right], \quad (51)$$

where $\mathbf{K} = [K_{ij}]_{i,j=1,\dots,N}$ with $K_{ij} \equiv K(\mathbf{x}_i, \mathbf{x}_j)$.

Given the function samples $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ obtained at spatial points $\mathbf{x}_i$, $i = 1, \dots, N$, GP predictions aim to make a probabilistic statement about the value $f_* \equiv f(\mathbf{x}_*)$ of the unknown function $f \sim \mathcal{GP}(\bar{f}, K)$ at a new spatial point $\mathbf{x}_*$. In other words, from a stochastic modeling view point, we are interested in making a new prediction of GP for f at any randomly chosen point $\mathbf{x}_*$. This is particularly of interest to us from the perspectives of FDM and FVM, because we can use GP to predict an unknown function value at cell interfaces (e.g., $u_{i \pm \frac{1}{2}}$ in 1D) where both FDM and FVM require estimates of flux functions.

We can accomplish this by utilizing the conditioning property of GP from the theory of Bayesian inference [4,48]. We look at the augmented likelihood function $\mathcal{L}_*$ by considering the joint distribution of the currently available training outputs, $\mathbf{f}$, and the new test output f_* ,

$$\mathcal{L}_* \equiv P(\mathbf{f}, f_*) = (2\pi)^{-\frac{N+1}{2}} \det |\mathbf{M}|^{-\frac{1}{2}} \exp \left[-\frac{1}{2} (\mathbf{g} - \bar{\mathbf{g}})^T \mathbf{M}^{-1} (\mathbf{g} - \bar{\mathbf{g}}) \right], \quad (52)$$

where $\mathbf{g}$ and $\bar{\mathbf{g}}$ are the $(N+1)$ -dimensional vectors whose components, in partitioned form, are

$$\mathbf{g} \equiv [f_*, \mathbf{f}]^T, \quad \bar{\mathbf{g}} \equiv [\bar{f}(\mathbf{x}_*), \bar{\mathbf{f}}]^T, \quad (53)$$

and $\mathbf{M}$ is the $(N+1) \times (N+1)$ augmented covariance matrix, given in a partitioned form by

$$\mathbf{M} = \begin{pmatrix} k_{**} & \mathbf{k}_*^T \\ \mathbf{k}_* & \mathbf{K} \end{pmatrix}. \quad (54)$$

In Eq. (54), we've defined a scalar k_{**} and an N -dimensional vector $\mathbf{k}_* = [\mathbf{k}_{*,i}]_{i=1,\dots,N}$ given by

$$k_{**} \equiv K(\mathbf{x}_*, \mathbf{x}_*), \quad \mathbf{k}_{*,i} \equiv K(\mathbf{x}_*, \mathbf{x}_i). \quad (55)$$

Using Bayes' Theorem, the conditioning property applied to the joint Gaussian prior distribution on the observation $\mathbf{f}$ yields the Gaussian posterior distribution of f_* given $\mathbf{f}$. One may then straightforwardly derive [4,48]:

$$P(f_*|\mathbf{f}) = (2\pi U^2)^{-\frac{1}{2}} \exp \left[-\frac{(f_* - \tilde{f}_*)^2}{2U^2} \right], \quad (56)$$

where the newly *updated posterior mean function* is

$$\tilde{f}_* \equiv \bar{f}(\mathbf{x}_*) + \mathbf{k}_*^T \mathbf{K}^{-1} \cdot (\mathbf{f} - \bar{\mathbf{f}}), \quad (57)$$

and the newly *updated posterior covariance* is

$$U^2 \equiv k_{**} - \mathbf{k}_*^T \mathbf{K}^{-1} \cdot \mathbf{k}_*. \quad (58)$$

What has happened here is that the GP on the unknown function f , trained on the data $\mathbf{f}$, has resulted in a Gaussian posterior probability distribution on the unknown function value $f(\mathbf{x}_*)$ at a new desired location $\mathbf{x}_*$, with a mean $\tilde{f}_*$ as given in Eq. (57), and with a variance as given in Eq. (58).

Note that an information on an uncertainty in the estimate is given by U in Eq. (58). This posterior uncertainty is of crucial importance in many GP modeling applications, but it is of limited interest for the purpose of the current paper. The roles of the posterior uncertainty U will be carefully studied in our future work and we will overlook it in the current study, while focusing on the posterior mean formula given in Eq. (3).

B Appendix: Choosing the Optimal Mean f_0 Using the Maximum Likelihood Function

In most practical applications we will not have any prior information on the mean of the function samples. This makes it reasonable to take a zero mean function $f_0 = 0$ if we try to achieve the simplest and most general symmetry of any random samples. However, one can easily determine the best optimal choice of the mean value by maximizing the likelihood

of the samples. This can be done by taking the logarithm function of Eq. (2) to first get the log-likelihood function of the function samples $\mathbf{f} = [f(\mathbf{x}_1), \dots, f(\mathbf{x}_N)]^T$ that are pointwise,

$$\ln \mathcal{L} = \ln P(\mathbf{f}) = -\frac{1}{2} (\mathbf{f} - \bar{\mathbf{f}})^T \mathbf{K}^{-1} (\mathbf{f} - \bar{\mathbf{f}}) - \frac{1}{2} \ln(\det |\mathbf{K}|) - \frac{N}{2} \ln(2\pi). \quad (59)$$

To maximize the likelihood, we take the partial derivative of Eq. (59) with respect to f_0 , where $\mathbf{f} = f_0 \mathbf{1}_N$, and set the partial derivative to be zero,

$$\begin{aligned} 0 = \frac{\partial \mathcal{L}}{\partial f_0} &= -\mathbf{f}^T \mathbf{K}^{-1} \mathbf{1}_N - \mathbf{1}_N^T \mathbf{K}^{-1} \mathbf{f} + 2f_0 (\mathbf{1}_N^T \mathbf{K}^{-1} \mathbf{1}_N) \\ &= -2\mathbf{f}^T \mathbf{K}^{-1} \mathbf{1}_N + 2f_0 (\mathbf{1}_N^T \mathbf{K}^{-1} \mathbf{1}_N), \end{aligned} \quad (60)$$

where we used the symmetry of $\mathbf{K}^{-1}$ to get $\mathbf{f}^T \mathbf{K}^{-1} \mathbf{1}_N = \mathbf{1}_N^T \mathbf{K}^{-1} \mathbf{f}$. The best optimal value of f_0 is then given by

$$f_0 = \frac{\mathbf{1}_N^T \mathbf{K}^{-1} \mathbf{f}}{\mathbf{1}_N^T \mathbf{K}^{-1} \mathbf{1}_N}, \quad (61)$$

which maximizes the likelihood of the samples in Eq. (2) in the case of interpolations.

In a similar way, we obtain the best optimal value of f_0 for the case of reconstructions, given as

$$f_0 = \frac{\mathbf{1}_N^T \mathbf{C}^{-1} \mathbf{G}}{\mathbf{1}_N^T \mathbf{C}^{-1} \mathbf{1}_N}, \quad (62)$$

where $\mathbf{G} = [\langle q_1 \rangle, \dots, \langle q_N \rangle]^T$ contains N volume-averaged values and $\mathbf{C}$ is the integrated kernel in Eq. (15).

These optimal choices of f_0 are obviously more expensive in computation as they involve local calculations on each GP stencil. In all the tests we presented in this paper we do not see any significant evidence that these optimal values result in any gains in terms of solution accuracy and computational efficiency.

C Appendix: Eigenfunction Analysis of Covariance Kernels

We give a brief description of the eigenfunction analysis of covariance kernels. Gaussian process regression, as described in Sect. 2, can be equivalently viewed as Bayesian linear regression using a possibly infinite number of basis functions. One choice of basis functions is the eigenfunctions, $\phi_i(\cdot)$, $i = 1, \dots, n$, of the covariance kernel function, k . By definition, each eigenfunction ϕ_i satisfies

$$\int k(x, x') \phi_i(x) d\mu(x) = \lambda_i \phi_i(x'). \quad (63)$$

The eigenpair (λ_i, ϕ_i) consists of the eigenvalue and the eigenfunction of the covariance kernel function k with respect to the measure μ [4, 48].

Of interest is when there is a non-negative function $p(x)$ called the *density* of the measure μ so that the measure may be written as $d\mu(x) = p(x)dx$. We can approximate the integral in Eq. (63) by choosing n finite sample points, x_l , $l = 1, \dots, n$, from $p(x)$ on some stencil, which gives

$$\lambda_i \phi_i(x') \simeq \frac{1}{n} \sum_{l=1}^n k(x_l, x') \phi_i(x_l). \quad (64)$$

Plugging in $x' = x_l$ into Eq. (64) for $l = 1, \dots, n$ gives the matrix eigenproblem

$$\mathbf{K}\mathbf{v}_i = n\lambda_i\mathbf{v}_i, \quad (65)$$

where $\mathbf{K}$ is an $n \times n$ Gram matrix with entries $\mathbf{K}_{ij} = k(x_i, x_j)$, and $n\lambda_i$ is the i -th matrix eigenvalue associated with the normalized n -dimensional eigenvector $\mathbf{v}_i$ satisfying $\mathbf{v}_i^T \mathbf{v}_j = \delta_{ij}$. We see that $\phi_i(x_j) \sim \sqrt{n}\mathbf{e}_j \cdot \mathbf{v}_i$ where $\mathbf{e}_j$ is the j -th unit vector, and we get the $\sqrt{n}$ factor from differing normalizations of the eigenvector and eigenfunction. This eigendecomposition of the covariance matrix, when viewed as a principal component analysis (PCA) [4, 48], can be thought of as decomposing the covariance into the various variational modes that can be represented on the n sample points. For more discussion of eigenfunction analysis of covariance kernels, we refer readers to Section 4.3 of [48]; Section 12.3 of [4].

A set of n samples of a function $f(x)$, denoted as

$$\mathbf{f} = [f(x_1), f(x_2), \dots, f(x_n)]^T, \quad (66)$$

evaluated at the points x_l , $l = 1, \dots, n$ as in Eq. (64), can be expanded in the eigenbasis in (65) as

$$\mathbf{f} = \sum_{i=1}^n \alpha_i \mathbf{v}_i. \quad (67)$$

We can then define a norm in the Hilbert space of functions using the inner product of the function in the reproducing kernel Hilbert space (RKHS) defined by the kernel function $k(x, x')$ [48],

$$\|\mathbf{f}\|_{\mathcal{H}}^2 = \langle \mathbf{f}, \mathbf{f} \rangle_{\mathcal{H}} = \sum_{i=1}^n \frac{\alpha_i^2}{\lambda_i}. \quad (68)$$

In this expression for the norm $\|\mathbf{f}\|_{\mathcal{H}}^2$, the relative weight of a given mode in the data $\mathbf{f}$ is inversely weighted by the eigenvalue λ_i . The SE kernel has a native Hilbert space of C^∞ functions. This makes SE assign relatively small eigenvalues to eigenfunctions that change rapidly on short length scales.

An important observation we can draw from this is that the quantity $\|\mathbf{f}\|_{\mathcal{H}}^2$ thus contains large additive terms when the data embodies a discontinuity, while data corresponding to smooth regions produces smaller additive terms. This suggests that we may use $\|\mathbf{f}\|_{\mathcal{H}}^2$ to supply the basis for a GP-based smoothness indicator in the nonlinear weights in Eq. (35).

D Appendix: Operation Counts

We display possible outcomes of computing the GP-based new smoothness indicators in two different approaches described in Sect. 2.5:

- **Option 1:** Compute α_i^m using Eqs. (39) and (41),
- **Option 2:** Compute α_i^m using Eqs. (43) and (44).

Consider a 1D domain discretized with N_x grid cells. Let's say we use a GP stencil of a radius R , which is subdivided into $R + 1$ sub-stencils, S_m , $m = 1, \dots, R + 1$. Since the calculations described in Sect. 2.5 to obtain smoothness indicators need to take place on every cell, there are N_x operations involved in total, each of which has M many operation counts at each S_R level. Below, we estimate the total number of required operation counts for Option 1 and Option 2.

We first consider **Option 1**. With a computer code with successive multiplications and additions for a dot product, Eq. (39) involves $R + 1$ multiplications and $R + 1$ additions for each $m = 1, \dots, R + 1$. This gives $2(R + 1)$ operations for Eq. (39) for each m and l . Likewise, we also have $2(R + 1)$ operations for Eq. (41) for each m and i . As a result, we see there are $2m(R + 1)(l + i)$ operations involved on each S_R . Since there are N_x many S_R calculations overall, the total number of operations becomes $2m(R + 1)(l + i)N_x$ per each time step, where $m, i, l = 1, \dots, R + 1$. Assuming there are M time steps required for the run, we require $2m(R + 1)(l + i)N_x M$ operations in total.

Let us now consider **Option 2**. A major difference, in this case, is to realize the fact that Eq. (43) can be pre-computed as soon as the grid is configured. This is because unlike $f^m(x_l)$ in Eq. (39), the vector $\mathbf{P}_i^m$ has no dependency on any local data but only on the grid itself, whereby it can be pre-computed and saved for each m and i before evolving each simulation, and reused during the simulation evolutions. The typical operation count for an $(R + 1)$ -dimensional vector and an $(R + 1) \times (R + 1)$ -dimensional matrix multiplication with successive multiplications and additions are found to be $2(R + 1)^2$. Hence we see there are $2im(R + 1)^2$ total operations that can be pre-computed and saved, $m, i = 1, \dots, R + 1$. Here is an extra operation reduction available by realizing that Eq. (43) is same for all m , thereby we have $2i(R + 1)^2$ total operations for $i = 1, \dots, R + 1$. Eq. (44) however, needs to be computed locally during the run because it depends on the local data $\mathbf{G}_m$ that evolves in time. For each m and i , the dot product in Eq. (44) involves $2(R + 1)$ operations, totaling $2im(R + 1)N_x$ per each time step, $m, i = 1, \dots, R + 1$. As a result, we see there are $2i(R + 1)^2$ operations initially, plus $2im(R + 1)N_x$ during the run per each time step. The simulation with M time steps then involves $2i(R + 1)^2 + 2im(R + 1)N_x M = 2im(R + 1)[R + 1 + N_x M]$ in total.

In comparison, we see there is a factor of two performance gain in **Option 2** because the ratio of the two options becomes

$$\frac{\text{Operation No. Option 1}}{\text{Operation No. Option 2}} = \frac{2m(R + 1)(l + i)N_x M}{2i(R + 1)[R + 1 + mN_x M]} \quad (69)$$

$$= \frac{2N_x M}{1 + N_x M} \approx 2, \quad (70)$$

where we used $i, l, m = R + 1$.

References

1. Arora, M., Roe, P.L.: On postshock oscillations due to shock capturing schemes in unsteady flows. *J. Comput. Phys.* **130**(1), 25–40 (1997)
2. Attig, N., Gibbon, P., Lippert, T.: Trends in supercomputing: the European path to exascale. *Comput. Phys. Commun.* **182**(9), 2041–2046 (2011)
3. Bigoni, C., Hesthaven, J.S.: Adaptive weno methods based on radial basis functions reconstruction. Technical representative. Springer, Berlin (2016)
4. Bishop, C.: Pattern recognition and machine learning (information science and statistics), 1st edn. 2006, corr. 2nd printing edn. Springer, New York (2007)
5. Bond, J., Crittenden, R., Jaffe, A., Knox, L.: Computing challenges of the cosmic microwave background. *Comput. Sci. Eng.* **1**(2), 21–35 (1999)
6. Brio, M., Wu, C.C.: An upwind differencing scheme for the equations of ideal magnetohydrodynamics. *J. Comput. Phys.* **75**(2), 400–422 (1988)
7. Buchmüller, P., Helzel, C.: Improved accuracy of high-order WENO finite volume methods on Cartesian grids. *J. Sci. Comput.* **61**(2), 343–368 (2014)
8. Chen, X., Jung, J.H.: Matrix stability of multiquadric radial basis function methods for hyperbolic equations with uniform centers. *J. Sci. Comput.* **51**(3), 683–702 (2012)

9. Chen, Y., Gottlieb, S., Heryudono, A., Narayan, A.: A reduced radial basis function method for partial differential equations on irregular domains. *J. Sci. Comput.* **66**(1), 67–90 (2016)
10. Colella, P., Woodward, P.R.: The piecewise parabolic method (PPM) for gas-dynamical simulations. *J. Comput. Phys.* **54**(1), 174–201 (1984)
11. Cressie, N.: *Statistics for Spatial Data*. Wiley, Hoboken (2015)
12. Dongarra, J.: *On the Future of High Performance Computing: How to Think for Peta and Exascale Computing*. Hong Kong University of Science and Technology, Hong Kong (2012)
13. Dongarra, J.J., Meuer, H.W., Simon, H.D., Strohmaier, E.: Recent trends in high performance computing. *Birth Numer. Anal.* **27**, 93 (2010)
14. Einfeldt, B., Munz, C.D., Roe, P.L., Sjögreen, B.: On Godunov-type methods near low densities. *J. Comput. Phys.* **92**(2), 273–295 (1991)
15. Fasshauer, G.E., Zhang, J.G.: On choosing “optimal” shape parameters for RBF approximation. *Numer. Algorithms* **45**(1), 345–368 (2007)
16. Fornberg, B., Wright, G.: Stable computation of multiquadric interpolants for all values of the shape parameter. *Comput. Math. Appl.* **48**(5), 853–867 (2004)
17. Franke, R.: Scattered data interpolation: tests of some methods. *Math. Comput.* **38**(157), 181–200 (1982)
18. Gerolymos, G., Sénéchal, D., Vallet, I.: Very-high-order WENO schemes. *J. Comput. Phys.* **228**(23), 8481–8524 (2009). <https://doi.org/10.1016/j.jcp.2009.07.039>
19. Godunov, S.K.: A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb.* **47**(89)(3), 271–306 (1959)
20. Gottlieb, D., Shu, C.W.: On the Gibbs phenomenon and its resolution. *SIAM Rev.* **39**(4), 644–668 (1997)
21. Guo, J., Jung, J.H.: A RBF-weno finite volume method for hyperbolic conservation laws with the monotone polynomial interpolation method. *Appl. Numer. Math.* **112**, 27–50 (2017)
22. Hardy, R.L.: Multiquadric equations of topography and other irregular surfaces. *J. Geophys. Res.* **76**(8), 1905–1915 (1971)
23. Harten, A., Engquist, B., Osher, S., Chakravarthy, S.R.: Uniformly high order accurate essentially non-oscillatory schemes, iii. *J. Comput. Phys.* **71**(2), 231–303 (1987)
24. Heryudono, A.R., Driscoll, T.A.: Radial basis function interpolation on irregular domain through conformal transplantation. *J. Sci. Comput.* **44**(3), 286–300 (2010)
25. Hesthaven, J.S., Gottlieb, S., Gottlieb, D.: *Spectral Methods for Time-Dependent Problems*, vol. 21. Cambridge University Press, Cambridge (2007)
26. Jiang, G.S., Shu, C.W.: Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**(1), 202–228 (1996)
27. Jin, S., Liu, J.G.: The effects of numerical viscosities: I. slowly moving shocks. *J. Comput. Phys.* **126**(2), 373–389 (1996)
28. Johnsen, E., Lele, S.: Numerical Errors Generated in Simulations of Slowly Moving shocks, pp. 1–12. Center for Turbulence Research Annual Research Briefs, Stanford (2008)
29. Jung, J.H., Gottlieb, S., Kim, S.O., Bresten, C.L., Higgs, D.: Recovery of high order accuracy in radial basis function approximations of discontinuous problems. *J. Sci. Comput.* **45**(1), 359–381 (2010)
30. Karni, S., Čanić, S.: Computations of slowly moving shocks. *J. Comput. Phys.* **136**(1), 132–139 (1997)
31. Katz, A., Jameson, A.: A comparison of various meshless schemes within a unified algorithm. In: *AIAA Paper*, vol. 594 (2009)
32. Keyes, D.E., McInnes, L.C., Woodward, C., Gropp, W., Myra, E., Pernice, M., Bell, J., Brown, J., Clo, A., Connors, J., et al.: Multiphysics simulations challenges and opportunities. *Int. J. High Perf. Comput. Appl.* **27**(1), 4–83 (2013)
33. Kolmogorov, A.: Interpolation und Extrapolation von stationären zufälligen Folgen. *Izv. Akad. Nauk. SSSR* **5**, 3–14 (1941)
34. Lee, D.: An upwind slope limiter for PPM that preserves monotonicity in magnetohydrodynamics. In: *5th International Conference of Numerical Modeling of Space Plasma Flows (ASTRONUM 2010)*, vol. 444, p. 236 (2011)
35. Lee, D.: A solution accurate, efficient and stable unsplit staggered mesh scheme for three dimensional magnetohydrodynamics. *J. Comput. Phys.* **243**, 269–292 (2013)
36. LeVeque, R.J.: *Finite Volume Methods for Hyperbolic Problems*, vol. 31. Cambridge university press, Cambridge (2002)
37. LeVeque, R.J.: *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*. SIAM, Bangkok (2007)
38. Li, S.: An HLLC riemann solver for magneto-hydrodynamics. *J. Comput. Phys.* **203**(1), 344–357 (2005)
39. Liu, X.D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. *J. Comput. Phys.* **115**(1), 200–212 (1994)

40. Liu, X.Y., Karageorghis, A., Chen, C.: A kansa-radial basis function method for elliptic boundary value problems in annular domains. *J. Sci. Comput.* **65**(3), 1240–1269 (2015)
41. Martel, J.M., Platte, R.B.: Stability of radial basis function methods for convection problems on the circle and sphere. *J. Sci. Comput.* **69**(2), 487–505 (2016)
42. McCorquodale, P., Colella, P.: A high-order finite-volume method for conservation laws on locally refined grids. *Commun. Appl. Math. Comput. Sci.* **6**(1), 1–25 (2011)
43. Mignone, A., Tzeferacos, P., Bodo, G.: High-order conservative finite difference GLM-MHD schemes for cell-centered MHD. *J. Comput. Phys.* **229**(17), 5896–5920 (2010). <https://doi.org/10.1016/j.jcp.2010.04.013>
44. Moroney, T.J., Turner, I.W.: A finite volume method based on radial basis functions for two-dimensional nonlinear diffusion equations. *Appl. Math. Model.* **30**(10), 1118–1133 (2006)
45. Moroney, T.J., Turner, I.W.: A three-dimensional finite volume method based on radial basis functions for the accurate computational modelling of nonlinear diffusion equations. *J. Comput. Phys.* **225**(2), 1409–1426 (2007)
46. Morton, K., Sonar, T.: Finite volume methods for hyperbolic conservation laws. *Acta Numer.* **16**(1), 155–238 (2007)
47. Powell, M.J.: Radial basis functionn for multivariable interpolation: a review. In: *IMA Conference on Algorithms for the Approximation of Functions and Data*, pp. 143–167. RMCS (1985)
48. Rasmussen, C., Williams, C.: *Gaussian Processes for Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, Cambridge (2005)
49. Rippa, S.: An algorithm for selecting a good value for the parameter c in radial basis function interpolation. *Adv. Comput. Math.* **11**(2), 193–210 (1999)
50. Roberts, T.W.: The behavior of flux difference splitting schemes near slowly moving shock waves. *J. Comput. Phys.* **90**(1), 141–160 (1990)
51. Roe, P.L.: Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.* **43**(2), 357–372 (1981)
52. Ryu, D., Jones, T.: Numerical magnetohydrodynamics in astrophysics: algorithm and tests for one-dimensional flow. [arXiv:astro-ph/9404074](https://arxiv.org/abs/astro-ph/9404074) (1994)
53. Shankar, V., Wright, G.B., Kirby, R.M., Fogelson, A.L.: A radial basis function (RBF)-finite difference (FD) method for diffusion and reaction–diffusion equations on surfaces. *J. Sci. Comput.* **63**(3), 745–768 (2015)
54. Shu, C.W.: Total-variation-diminishing time discretizations. *SIAM J. Sci. Stat. Comput.* **9**(6), 1073–1084 (1988)
55. Shu, C.W.: High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM Rev.* **51**(1), 82–126 (2009)
56. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *J. Comput. Phys.* **83**(1), 32–78 (1989)
57. Sod, G.A.: A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **27**(1), 1–31 (1978)
58. Sonar, T.: Optimal recovery using thin plate splines in finite volume methods for the numerical solution of hyperbolic conservation laws. *IMA J. Numer. Anal.* **16**(4), 549–581 (1996)
59. Stein, M.: *Interpolation of Spatial Data: Some Theory for Kriging*. Springer Series in Statistics Series. Springer, New York (1999)
60. Striba, Y., Donat, R.: A numerical study of postshock oscillations in slowly moving shock waves. *Comput. Math. Appl.* **46**(5), 719–739 (2003)
61. Subcommittee, A.: Top ten exascale research challenges. US Department Of Energy Report, 2014 (2014)
62. Toro, E.: *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer, Berlin (2009)
63. Toro, E.F., Spruce, M., Speares, W.: Restoration of the contact surface in the HLL-riemann solver. *Shock Waves* **4**(1), 25–34 (1994)
64. Torrilhon, M.: Non-uniform convergence of finite volume schemes for Riemann problems of ideal magnetohydrodynamics. *J. Comput. Phys.* **192**(1), 73–94 (2003)
65. Torrilhon, M.: Uniqueness conditions for Riemann problems of ideal magnetohydrodynamics. *J. Plasma Phys.* **69**(03), 253–276 (2003)
66. Van Leer, B.: Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov’s method. *J. Comput. Phys.* **32**(1), 101–136 (1979)
67. Wahba, G., Johnson, D., Gao, F., Gong, J.: Adaptive tuning of numerical weather prediction models: randomized GCV in three- and four-dimensional data assimilation. *Mon. Weather Rev.* **123**, 3358–3369 (1995)

68. Wendland, H.: *Scattered Data Approximation*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, Cambridge (2010)
69. Wiener, N.: *Extrapolation, interpolation, and smoothing of stationary time series, with engineering applications*. Technology Press of the Massachusetts Institute of Technology, Cambridge (1949). "First published during the war as a classified report to Section D 2, National Defense Research Committee."; *Stationary time series*
70. Woodward, P., Colella, P.: The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.* **54**(1), 115–173 (1984)
71. Wright, G.B.: *Radial Basis Function Interpolation: Numerical and Analytical Developments*. University of Colorado, Boulder (2003)
72. Wright, G.B., Fornberg, B.: Stable computations with flat radial basis functions using vector-valued rational approximations. *J. Comput. Phys.* **331**, 137–156 (2017)
73. Zhang, R., Zhang, M., Shu, C.W.: On the order of accuracy and numerical performance of two classes of finite volume WENO schemes. *Commun. Comput. Phys.* **9**(03), 807–827 (2011)