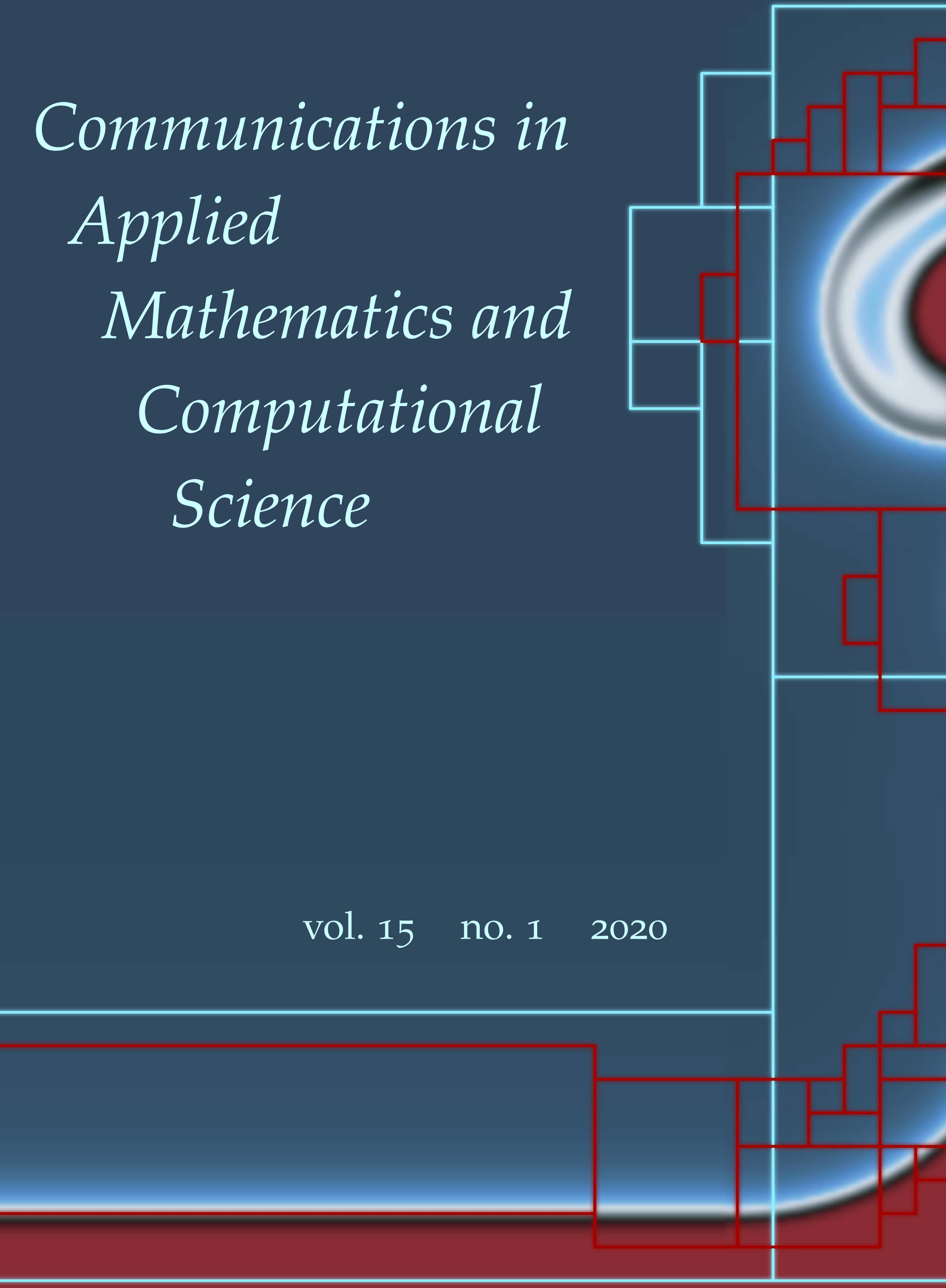
# Communications in Applied Mathematics and Computational Science



vol. 15 no. 1 2020

# Communications in Applied Mathematics and Computational Science

msp.org/camcos

See inside back cover or msp.org/camcos for submission instructions.

The subscription price for 2020 is US $110/year for the electronic version, and $165/year (+$15, if shipping outside the US) for print and electronic. Subscriptions, requests for back issues from the last three years and changes of subscriber address should be sent to MSP.

Communications in Applied Mathematics and Computational Science (ISSN 2157-5452 electronic, 1559-3940 printed) at Mathematical Sciences Publishers, 798 Evans Hall #3840, c/o University of California, Berkeley, CA 94720-3840, is published continuously online. Periodical rate postage paid at Berkeley, CA 94704, and additional mailing offices.

CAMCoS peer review and production are managed by EditFLOW® from MSP.

PUBLISHED BY

mathematical sciences publishers

nonprofit scientific publishing

http://msp.org/

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 1, 2020
dx.doi.org/10.2140/camcos.2020.15.1

# INVESTIGATION OF FINITE-VOLUME METHODS TO CAPTURE SHOCKS AND TURBULENCE SPECTRA IN COMPRESSIBLE FLOWS

EMMANUEL MOTHEAU AND JOHN WAKEFIELD

The aim of the present paper is to provide a comparison between several finite-volume methods of different numerical accuracy: the second-order Godunov method with PPM interpolation and the high-order finite-volume WENO method. The results show that while on a smooth problem the high-order method performs better than the second-order one, when the solution contains a shock all the methods collapse to first-order accuracy. In the context of the decay of compressible homogeneous isotropic turbulence with shocklets, the actual overall order of accuracy of the methods reduces to second-order, despite the use of fifth-order reconstruction schemes at cell interfaces. Most important, results in terms of turbulent spectra are similar regardless of the numerical methods employed, except that the PPM method fails to provide an accurate representation in the high-frequency range of the spectra. It is found that this specific issue comes from the slope-limiting procedure and a novel hybrid PPM/WENO method is developed that has the ability to capture the turbulent spectra with the accuracy of a high-order method, but at the cost of the second-order Godunov method. Overall, it is shown that virtually the same physical solution can be obtained much faster by refining a simulation with the second-order method and carefully chosen numerical procedures, rather than running a coarse high-order simulation. Our results demonstrate the importance of evaluating the accuracy of a numerical method in terms of its actual spectral dissipation and dispersion properties on mixed smooth/shock cases, rather than by the theoretical formal order of convergence rate.

## 1. Introduction

The utility of high-order accurate numerical methods has been a subject of discussion within the computational fluid dynamics (CFD) community for several decades. As suggested in the review paper [29], one of the myths in the debate over low- versus high-order numerical methods is the ability to get an accurate solution at a

*MSC2010:* 35L67, 65N08, 76F05, 76F50, 76F65.

*Keywords:* homogeneous isotropic turbulence, high-order numerical methods, WENO, PPM, Godunov, computational fluid dynamics, compressible turbulent flows.

reduced computational cost. High-order methods are more costly on a per point basis but can potentially obtain a solution of the desired accuracy on a coarse mesh. Low-order methods are easier to implement, less costly per point, but require a finer mesh to obtain accuracy equivalent to a high-order method.

The theoretical order of accuracy $k$ of a numerical method describes the order of the truncation error made when approximating the derivative of a function via a numerical discretization. In practice, the order of accuracy can be quantified by the asymptotic rate of convergence of the solution error $\varepsilon$ with respect to the mesh size $h$, namely $\varepsilon \propto h^k$. This type of theoretical asymptotic estimate argues for the utility of high-order methods ([29] defines high-order as $k > 3$). However, realizing this type of convergence depends on the smoothness of the solution.

In most CFD applications, particularly those involving turbulent flow, the solution is adequately resolved well before reaching the asymptotic regime of the numerical method (see the discussion in [1]). This issue is exacerbated for compressible flow. The solution can include shock waves that require local dissipation to prevent the appearance of spurious nonphysical oscillations in the solution, reducing the order of accuracy of the numerical method employed.

A more realistic way to assess a numerical method is to determine the cost needed to obtain a desired accuracy. In the context of viscous compressible turbulent flow, we can frame the question in terms of the resolution required to resolve the spectrum of the turbulent flow. Indeed, it is emphasized that the performance of a numerical method should not be defined only by the order of the convergence of the error for smooth solutions. A better measure for the actual accuracy is the ability of the numerical method to adequately resolve both the inertial range and the dissipative range of the turbulent energy spectrum.

Unfortunately the literature on the development of numerical methods often provides tests and comparisons based on canonical cases, which consist of the propagation of smooth solutions or very specific cases with discontinuities. As an alternative we propose investigating the performance of numerical schemes for resolving the spectrum of a complex turbulent flow, especially in a context where both shocks and a wide range of turbulent scales interact in the flow field. To the authors' knowledge, only a few papers [4; 14] deal with such a complete study. However, [4] only investigates incompressible flow, while in [14] the impact of the mesh resolution is not investigated and simulations are only performed on a coarse mesh. As will be shown in the present paper, refinement of the mesh allows spurious small structures to develop and may lead to inaccurate spectra in the high-frequency range. We advocate that one of the most important features of a numerical method should be its robustness to any discretization size.

Many different numerical methods exist to solve partial differential equations, and each of them present pros and cons depending on the problem investigated.

For example compact finite difference schemes [16] are very efficient to accurately capture turbulent energy spectra, but their performance quickly degrades when applied to geometries more complicated than a triply periodic cubic box, and/or if the solution is not smooth enough. In the context of the simulation of flows in engineering applications, complex geometries are often involved and multiphysics phenomena can occur. See for example [22] where simulations of flames are performed in a realistic gas turbine combustion chamber. For such complicated applications, finite-volume methods are often preferred because they are intrinsically conservative, robust, and flexible enough to handle both unstructured and structured meshes. Moreover, finite-volume methods fit naturally within the paradigm of adaptive mesh refinement (AMR) using the concept of refluxing across multigrids to achieve conservation properties.

The goal of the present paper is to compare and investigate the performance of several popular finite-volume methods for the compressible Navier–Stokes equations. Let's recall that a finite-volume method seeks to reconstruct data at the interface between cells, and then to solve a Riemann problem so as to evaluate the fluxes that cross the cells. As explained above, a flow may contain shocks. In typical compressible Navier–Stokes the associated shock profiles are so thin that they cannot for all practical purposes be represented by the points of a numerical mesh. Because from one cell to another there is a strong difference in the states of the flow, a specialized treatment is given to reconstruct fluxes that capture the discontinuities without introducing spurious oscillations. Several techniques have been proposed in the literature, but a complete review is beyond the scope of the present paper and can be found in reference textbooks [17; 26].

In the present paper, two techniques are considered. First, in the asymptotic second-order Godunov method, the classical PPM interpolation procedure [8; 20] considers several limiters to enforce the monotonicity, for example starting with the van Leer method [28]. Second, the present paper also investigates the high-order finite-volume method developed by [25], which is based on the weighted essentially nonoscillatory (WENO) schemes. There is an extensive literature on different variants of WENO schemes and a complete description is beyond the scope of the present paper; a review can be found in [24]. The basic idea of WENO schemes is to provide a high-order nonlinear reconstruction method, which effectively captures discontinuities but can also be dissipative on smooth solutions. Note that several different WENO variants were tested during the present study and it has been found that, overall, they provide similar results despite exhibiting some robustness discrepancies. Thus, for clarity purposes, only one WENO variant is employed in this paper, but we provide in Appendix C more comprehensive results to highlight the performance and robustness issues that we encountered while testing the different WENO variants.

Three test cases of increasing complexity are investigated in the present paper. First, the convection of a smooth vortex is considered, followed by the simulation of a classical shock-driven Shu–Osher problem. It is emphasized that these test cases are chosen here because they are commonly employed in the literature to assess performance of numerical schemes. Here the results show that while on a smooth problem the high-order method performs better than the second-order one, when the solution contains a shock all the methods collapse to first-order accuracy. Finally, the decay of compressible homogeneous isotropic turbulence (HIT) with shocklets is investigated. Comparisons reveal that a second-order Godunov method with the classical PPM interpolation provides essentially the same results as a fourth-order finite-volume WENO scheme but at a significantly lower cost. It is emphasized that virtually the same physical solution can be obtained much faster by refining a simulation with the second-order method, rather than running a coarse high-order simulation. However, the results also show that the refinement of the mesh presents some limits when using the second-order Godunov procedure with the classical PPM interpolation. Indeed, it is found that when the mesh is fine enough, a nonphysical pile-up of energy appears in the high-frequency range of the turbulent spectra. After an intensive trial and error process, it has been found that the limiting procedures employed by the PPM to ensure monotonicity are responsible for this pile-up of energy in the high-frequency range of the spectra.

One of the most significant innovations of the present paper is to propose replacing the interpolation and limiting procedures at cell interfaces in the classical PPM algorithm by a WENO interpolation. It is shown that the novel proposed hybrid PPM/WENO method has the ability to capture the turbulent spectra with the accuracy of a high-order method, but at the cost of the second-order Godunov method.

This study makes use of CFD software developed at the Center for Computational Sciences and Engineering (CCSE) group[1] at the Lawrence Berkeley National Laboratory in the USA. The codes are implemented in the AMReX framework,[2] which facilitates the development of a generic postprocessing chain as well as the assessment of computing costs via embedded profiling functionality. Note that while the AMReX library supports AMR applications, only single-level grids are employed in the present paper. Two codes are being compared:

- PeleC, which is based on a second-order Godunov procedure. Interpolation to evaluate data at cell faces is performed either with the original unsplit PPM [20] method, or with the hybrid PPM/WENO developed in the present paper. The diffusion operators are evaluated with a second-order finite-volume discretization.

[1] https://ccse.lbl.gov

[2] https://amrex-codes.github.io/amrex/

- RNS, which is based on a fourth-order finite-volume WENO method [25] in space. Note that RNS was originally built for the development of the adaptive multilevel spectral deferred correction (AMLSDC) method, which is a fourth-order time-integration method [9], but in the present paper the classical Runge–Kutta algorithm is employed instead. Note that the diffusion terms are discretized with a fourth-order conservative finite-volume technique. First, the cell-averaged conserved variables are used to compute fourth-order approximations to point values at cell centers using the procedure outlined by McCorquodale and Colella [19] and then explicit formulae are used to compute derivatives needed to compute the diffusive fluxes at Gauss points on the cell faces directly.

The remainder of the present paper is organized as follows. In Section 2, the set of equations solved by the codes is presented. In Section 3 the RNS code is presented, as well as a short description of the high-order finite-volume WENO scheme that is employed for the spatial discretization. Next, in Section 4 the PeleC code together with the original PPM algorithm are presented, followed in Section 4.3 by the novel hybrid PPM/WENO method developed in the present paper that captures the turbulent spectra with the accuracy of a high-order method at the cost of a second-order Godunov method. Results are then presented in Section 5. The convection of a smooth vortex and the Shu–Osher problem are investigated in Sections 5.1 and 5.2, respectively, while the decay of compressible homogeneous isotropic turbulence with shocklets is investigated in Section 5.3.

## 2. Governing equations

The software employed in the present study was initially developed for the simulation of combustion problems, and the codes solve the multicomponent reacting Navier–Stokes equations. However, only nonreacting problems with no specific mixture are investigated in the present study. Consequently, the set of equations solved are significantly simplified and are given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{u}) = 0, \tag{1}$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u}) + \nabla p = \nabla \cdot \boldsymbol{\tau}, \tag{2}$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot [(\rho E + p)\boldsymbol{u}] = \nabla \cdot (\lambda \nabla T) + \nabla \cdot (\boldsymbol{\tau} \cdot \boldsymbol{u}), \tag{3}$$

where $\rho$ is the density, $\boldsymbol{u}$ is the velocity, $p$ is the pressure, $E = e + \boldsymbol{u} \cdot \boldsymbol{u}/2$ is the total energy, $T$ is the temperature, and $\lambda$ is the thermal conductivity. The viscous stress tensor is given by

$$\boldsymbol{\tau} = \eta(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T) + \big(\varsigma - \tfrac{2}{3}\eta\big)(\nabla \cdot \boldsymbol{u})\boldsymbol{I}, \tag{4}$$

where $\eta$ and $\varsigma$ are the shear and bulk viscosities.

The system is closed by an equation of state (EOS) that specifies $p$ as a function of $\rho$ and $T$. An ideal gas mixture for the EOS is assumed:

$$p = \rho T \mathfrak{R}, \tag{5}$$

where $\mathfrak{R}$ is the specific gas constant. Here we set $C_p$ and $C_v$, the heat capacities at constant pressure and volume, respectively, to follow an ideal gas law proportional to the ratio of the specific heats $\gamma$ so that (5) is equivalent to the relation

$$e = p/(\gamma - 1)\rho \tag{6}$$

where $e$ is the specific internal energy and $\gamma$ is set to $\gamma = 1.4$.

Note that for simplicity, the system presented in (1), (2), and (3) is recast in the form of

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F} = \boldsymbol{S}, \tag{7}$$

where $\boldsymbol{U}$ is the vector of conservative variables, while $\boldsymbol{F}$ represents the convective flux vector and $\boldsymbol{S}$ contains the diffusive terms.

## 3. RNS: a high-order WENO-based finite-volume solver

The RNS code implements high-order temporal and spatial AMR integration methods for combustion applications. The major innovative feature of this code is the development of the adaptive multilevel spectral deferred correction (AMLSDC) method, which is fourth-order in time [9]. The Runge–Kutta method for AMR applications as presented in [19] is also implemented. In the present paper, the second-order explicit midpoint Runge–Kutta method is used. Note that although not shown in the present paper, the results were compared to the fourth-order Runge–Kutta and AMLSDC approaches, and results were virtually the same without impacting the spatial solutions, which is attributed to the fact that the time steps involved are small, and the spatial errors introduced at shocks dominate the solution.

The diffusion terms are discretized using standard finite-volume techniques. First, the cell-averaged conserved variables are used to compute fourth-order approximations to point values at cell centers using the procedure outlined by McCorquodale and Colella [19]. These point values of conserved quantities are then used to compute primitive variables, and explicit formulae are then used to compute derivatives needed to evaluate the diffusive fluxes at Gauss points on the cell faces directly. Similarly, diffusion coefficients are computed at cell centers using point values and are then interpolated to Gauss points.

The spatial discretization of the advection terms in the algorithm uses the conservative finite-volume WENO reconstruction presented in [25]. The following approach is repeated for each stage of the Runge–Kutta integration scheme:

(1) For each cell, the conservative equation (7) is rewritten in terms of primitive variables.

(2) The primitive variables are reconstructed at the cell interfaces with a fifth-order WENO scheme in order to provide a left and a right state for each face. For 2D and 3D cases, the variables are first reconstructed to Gauss quadrature nodes to evaluate their average value in the direction normal to the faces. This procedure is obviously computationally expensive, but as shown by [30], a midpoint rule for integrating fluxes is not sufficiently accurate to obtain fourth-order convergence. Note that although the solution is reconstructed at cell interfaces with fifth-order WENO procedures, the method is formally fourth-order accurate because a fourth-order quadrature rule is employed to integrate the flux over faces.

(3) The HLLC algorithm [27] is employed to reconstruct the fluxes through the faces.

In the present study, several different WENO schemes were investigated and results show that the so-called WENO-Z variant [5] performs the best. Some elements of the comparison results are presented in Appendix C. As the conservative finite-volume WENO method presented by [25] is based on the so-called WENO-JS scheme generalized by Jiang and Shu [13], we first review the basic principles, followed by a short description of the WENO-Z variant.

**3.1.** ***The WENO-JS method.*** For a given cell $i$, the principle of a WENO method is to provide a high-order approximation of the variable $q$ interpolated on the left and the right sides of a face, denoted $\hat{q}^L_{i+1/2}$ and $\hat{q}^R_{i-1/2}$. In the remainder of this section, the procedures to evaluate $\hat{q}^L_{i+1/2}$ are provided; $\hat{q}^R_{i-1/2}$ is evaluated analogously.

In the WENO-JS method proposed by [13], a fifth-order polynomial approximation of $\hat{q}^L_{i+1/2}$ is constructed through a convex combination of the values $\hat{q}^k_{i+1/2}$ interpolated with a third degree polynomial on a three-point stencil $k$, such that

$$\hat{q}^L_{i+1/2} = \sum_{k=0}^{2} \omega_k \hat{q}^k_{i+1/2} \tag{8}$$

with

$$\hat{q}^0_{i+1/2} = \tfrac{1}{6}(2q_{i-2} - 7q_{i-1} + 11q_i), \tag{9}$$

$$\hat{q}^1_{i+1/2} = \tfrac{1}{6}(-q_{i-1} + 5q_i + 2q_{i+1}), \tag{10}$$

$$\hat{q}^2_{i+1/2} = \tfrac{1}{6}(2q_i + 5q_{i+1} - q_{i+2}). \tag{11}$$

Here, $\omega_k$ are nonlinear weights balancing the contribution of each stencil, and the challenge is to find the best values to capture shocks the most accurately while preserving the resolution of the spectrum of a solution.

The weights $\omega_k$ are defined as

$$\omega_k = \frac{\alpha_k}{\sum_{l=0}^{2} \alpha_l}, \quad \alpha_k = \frac{d_k}{(\beta_k + \epsilon)^p}, \tag{12}$$

where $d_k$ are the so-called optimal weights because they reconstruct the fifth-order upstream central scheme for the five-point stencil, $\beta_k$ are the smoothness indicators, $\alpha_k$ are referred to as the unnormalized weights, and $\epsilon$ is a parameter set to avoid a division by zero. The parameter $p$ controls the adaptation rate. According to [3], a large value of $p$ leads to unnecessarily high dissipation in smooth regions of the flow. In the present study, the parameter is set to $p = 1$ for all the test cases. Moreover, as suggested by [3], $\epsilon$ is set to $\epsilon = 10^{-40}$.

The smoothness indicators $\beta_k$ are given by

$$\beta_0 = \tfrac{13}{12}(q_{i-2} - 2q_{i-1} + q_i)^2 + \tfrac{1}{4}(q_{i-2} - 4q_{i-1} + 3q_i)^2, \tag{13}$$

$$\beta_1 = \tfrac{13}{12}(q_{i-1} - 2q_i + q_{i+1})^2 + \tfrac{1}{4}(q_{i-1} - q_{i+1})^2, \tag{14}$$

$$\beta_2 = \tfrac{13}{12}(q_i - 2q_{i+1} + q_{i+2})^2 + \tfrac{1}{4}(3q_i - 4q_{i+1} + q_{i+2})^2. \tag{15}$$

One of the features of the conservative finite-volume WENO method is that the optimal weights as well as the formulae for the reconstructed values differ if the interpolation is performed in the normal direction at faces or at the Gauss integration points $\xi = \xi_i \pm \Delta\xi/(2\sqrt{3})$ [25].

- For the normal direction through a face, the optimal weights are

$$d_0 = \tfrac{1}{10}, \qquad d_1 = \tfrac{6}{10}, \qquad d_2 = \tfrac{3}{10}, \tag{16}$$

  and $\hat{q}^L_{i+1/2}$ is given by

$$\hat{q}^L_{i+1/2} = \tfrac{1}{6}\omega_0(2q_{i-2} - 7q_{i-1} + 11q_i) + \tfrac{1}{6}\omega_1(-q_{i-1} + 5q_i + 2q_{i+1}) + \tfrac{1}{6}\omega_2(2q_i + 5q_{i+1} - q_{i+2}). \tag{17}$$

- For the first Gaussian integration point $\xi = \xi_i + \Delta\xi/(2\sqrt{3})$, the optimal weights are

$$d_0 = \tfrac{210-\sqrt{3}}{1080}, \qquad d_1 = \tfrac{11}{18}, \qquad d_2 = \tfrac{210+\sqrt{3}}{1080}, \tag{18}$$

  and $q(\xi_i + \Delta\xi/(2\sqrt{3}))$ is given by

$$q\big(\xi_i + \tfrac{1}{2\sqrt{3}}\Delta\xi\big) = \omega_0\big[q_i - (-3q_i + 4q_{i-1} - q_{i-2})\tfrac{\sqrt{3}}{12}\big] + \omega_1\big[q_i - (q_{i-1} - q_{i+1})\tfrac{\sqrt{3}}{12}\big] + \omega_2\big[q_i - (3q_i - 4q_{i+1} + q_{i+2})\tfrac{\sqrt{3}}{12}\big]. \tag{19}$$

Recall here that a simple mirror-symmetric change to the coefficients and the formulae will provide $\hat{q}^R_{i-1/2}$ and $q(\xi_i - \Delta\xi/(2\sqrt{3}))$.

**3.2.** ***The WENO-Z method.*** A well known issue with the original WENO-JS method is that the smoothness indicators $\beta_k$ employed to compute the weights $\omega_k$ fail to recover the maximum order of the scheme at critical points when the derivatives of the flux function vanish. Borges et al. [5] propose a different approach to overcome the issues of the WENO-JS method by acting directly on the smoothness indicator $\beta_k$ with a very simple formulation. The so-called WENO-Z method is given by

$$\omega_k^{(z)} = \frac{\alpha_k^{(z)}}{\sum_{i=0}^{2} \alpha_i^{(z)}}, \quad \text{with } \alpha_k^{(z)} = d_k\left(1 + \frac{\tau_5}{\beta_k + \epsilon}\right)^p, \tag{20}$$

where

$$\tau_5 = |\beta_0 - \beta_2|. \tag{21}$$

Similarly to the WENO-JS method, the parameter $p$ controls the detection of the smoothness of the solution. In the present study, the parameter is set to $p = 1$ to reduce as much as possible the dissipation of the numerical scheme. Note that the WENO-Z method simply provides a new way to compute the nonlinear weights $\omega_k$ and can be directly implemented in the conservative finite-volume WENO method, regardless if the interpolation is performed in the normal direction at faces or at the Gauss integration points.

## 4. PeleC: the second-order Godunov-based finite-volume solver

The PeleC code is a second-order AMR finite-volume solver for reacting and nonreacting fluid simulations with complex geometry and support for Lagrangian spray particles. The simulations performed in the present paper only use a fraction of the capability of the software, namely the Godunov-based integration procedure on a single-level mesh grid. Note also that PeleC is part of the Pele Suite of codes, which are publicly available and may be freely downloaded,[3] and that all the test cases investigated in the present paper are available from the PeleC distribution and can be reproduced.

The solution is advanced from time $n$ to time $n+1$ with the second-order Godunov method

$$\boldsymbol{U}^* = \boldsymbol{U}^n - \Delta t\, \nabla \cdot \boldsymbol{F}^{n+1/2} + \Delta t\, \boldsymbol{S}^n, \tag{22}$$

$$\boldsymbol{U}^{n+1} = \boldsymbol{U}^* + \tfrac{1}{2}\Delta t (\boldsymbol{S}^* - \boldsymbol{S}^n), \tag{23}$$

where $\Delta t = t^{n+1} - t^n$ is the time step. The second step at (23) is a correction of the solution to ensure second-order accuracy by effectively time-centering the diffusion

[3] https://amrex-combustion.github.io/

source terms. The conserved state vector $\boldsymbol{U}$ is stored at cell centers, and the flux vectors are computed on cell edges.

The convective flux vector $\boldsymbol{F}$ that appears in (22) is constructed from time-centered edge states computed with a conservative, shock-capturing, unsplit Godunov method, which makes use of the piecewise parabolic method (PPM) [8], characteristic tracing, and full corner coupling [2; 20]. As the present paper proposes a modification of the PPM method, for ease of exposition the whole algorithm will be detailed in 1D for the Euler equations. It is emphasized that the algorithm can be extended to multidimensional problems and multicomponent flows. Moreover, since the publication of the original paper [8] presenting the PPM method, several modifications have been proposed in the literature [20; 7; 6]. Consequently, the algorithm implemented in the code PeleC incorporates some of the variants, but it is emphasized that these changes only differ slightly from the original PPM method. Many variants have been tested through this study, and while not reported in the present paper, none fundamentally change the results.

**4.1.** ***System of primitive variables.*** The conservative equation (7) is rewritten in terms of primitive variables, such that

$$\frac{\partial \boldsymbol{Q}}{\partial t} + \boldsymbol{A}\frac{\partial \boldsymbol{Q}}{\partial x} = \boldsymbol{S_Q}. \tag{24}$$

Here $\boldsymbol{Q}$ is the primitive state vector, $\boldsymbol{A} = \partial \boldsymbol{F}/\partial \boldsymbol{Q}$, and $\boldsymbol{S_Q}$ is the viscous source terms reformulated in terms of the primitive variables.

In one dimension, this becomes

$$\begin{pmatrix} \rho \\ u \\ p \\ \rho e \end{pmatrix}_t + \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 1/\rho & 0 \\ 0 & \rho c^2 & u & 0 \\ 0 & \rho e + p & 0 & u \end{pmatrix} \begin{pmatrix} \rho \\ u \\ p \\ \rho e \end{pmatrix}_x = \boldsymbol{S_Q}. \tag{25}$$

Note that here, the system of primitive variables has been extended to include an additional equation for the internal energy, denoted $e$. This avoids several calls to the equation of state, especially in the Riemann solver step.

The eigenvalues of the matrix $\boldsymbol{A}_x$ are given by

$$\boldsymbol{\Lambda}(\boldsymbol{A}_x) = \{u - c, u, u, u + c\}. \tag{26}$$

The right column eigenvectors are

$$\boldsymbol{r}_x = \begin{pmatrix} 1 & 1 & 0 & 1 \\ -c/\rho & 0 & 0 & c/\rho \\ c^2 & 0 & 0 & c^2 \\ h & 0 & 1 & h \end{pmatrix}. \tag{27}$$

The left row eigenvectors, normalized so that $\boldsymbol{l}_x \cdot \boldsymbol{r}_x = \boldsymbol{I}$, are

$$\boldsymbol{l}_x = \begin{pmatrix} 0 & -\rho/(2c) & 1/(2c^2) & 0 \\ 1 & 0 & -1/c^2 & 0 \\ 0 & 0 & -h/c^2 & 0 \\ 0 & \rho/(2c) & 1/(2c^2) & 0 \end{pmatrix}. \tag{28}$$

Note that here, $c$ and $h$ are the sound speed and the enthalpy, respectively.

**4.2.** ***Edge state prediction.*** As discussed at the beginning of Section 4, the fluxes are reconstructed from time-centered edge state values. Thus, the primitive variables are first interpolated in space with the PPM method; then a characteristic tracing operation is performed to extrapolate in time their values at $n + \frac{1}{2}$.

**4.2.1.** *Interpolation and slope limiting.* Basically the goal of the algorithm is to compute a left and a right state of the primitive variables at each edge in order to provide inputs for the Riemann problem to solve.

First, the average cross-cell difference is computed for each primitive variable with a quadratic interpolation as

$$\delta q_i = \tfrac{1}{2}(q_{i+1} - q_{i-1}). \tag{29}$$

In order to enforce monotonicity, $\delta q_i$ is limited with the van Leer [28] method

$$\delta q_i^* = \min(|\delta q_i|, 2|q_{i+1} - q_i|, 2|q_i - q_{i-1}|)\,\mathrm{sgn}(\delta q_i), \tag{30}$$

and the interpolation of the primitive values to the cell face $q_{i+1/2}$ is estimated with

$$q_{i+1/2} = q_i + \tfrac{1}{2}(q_{i+1} - q_i) - \tfrac{1}{6}(\delta q_{i+1}^* - \delta q_i^*). \tag{31}$$

In order to enforce that $q_{i+1/2}$ lies between the adjacent cell averages, the following constraint is imposed:

$$\min(q_i, q_{i+1}) \leqslant q_{i+1/2} \leqslant \max(q_i, q_{i+1}). \tag{32}$$

The next step is to set the values of $q_{R,i-1/2}$ and $q_{L,i+1/2}$, which are the right and left states at the edges bounding a computational cell. Here, a quartic limiter is employed in order to enforce that the interpolated parabolic profile is monotone. The procedure proposed by [20] is adopted, which differs slightly from the original one proposed in [8]. In [20], this specific procedure is followed by the imposition of another limiter based on a flattening parameter to prevent artificial extrema in the reconstructed values. In the present paper, the order of imposition of the different limiting procedures is reversed.

First, the edge state values are defined as

$$q_{L,i+1/2} = q_{i+1/2}, \tag{33}$$

$$q_{R,i-1/2} = q_{i-1/2}. \tag{34}$$

Then the flattening limiter is imposed as

$$q_{L,i+1/2} \leftarrow \chi_i q_{L,i+1/2} + (1+\chi_i) q_i, \tag{35}$$

$$q_{R,i-1/2} \leftarrow \chi_i q_{R,i-1/2} + (1+\chi_i) q_i, \tag{36}$$

where $\chi_i$ is a flattening coefficient computed from the local pressure, and its evaluation is presented in Appendix A.

Finally, the monotonization is performed with the procedure

$$q_{L,i+1/2} = q_{R,i-1/2} = q_i \quad \text{if } (q_{L,i+1/2} - q_i)(q_i - q_{R,i-1/2}) > 0, \tag{37}$$

$$q_{L,i+1/2} = 3q_i - 2q_{R,i-1/2} \quad \text{if } |q_{L,i+1/2} - q_i| \geqslant 2|q_{R,i-1/2} - q_i|, \tag{38}$$

$$q_{R,i-1/2} = 3q_i - 2q_{L,i+1/2} \quad \text{if } |q_{R,i-1/2} - q_i| \geqslant 2|q_{L,i+1/2} - q_i|. \tag{39}$$

**4.2.2.** *Piecewise parabolic reconstruction.* Once the limited values $q_{R,i-1/2}$ and $q_{L,i+1/2}$ are known, the limited piecewise parabolic reconstruction in each cell is done by computing the average value swept out by a parabolic profile across a face, assuming that it moves at the speed of a characteristic wave $\lambda_k$. The average is defined by the integrals

$$\mathcal{I}_{+}^{(k)}(q_i) = \frac{1}{\sigma_k \Delta x} \int_{((i+1/2)-\sigma_k)\Delta x}^{(i+1/2)\Delta x} q_i^I(x)\,\mathrm{d}x, \tag{40}$$

$$\mathcal{I}_{-}^{(k)}(q_i) = \frac{1}{\sigma_k \Delta x} \int_{(i-1/2)\Delta x}^{((i-1/2)+\sigma_k)\Delta x} q_i^I(x)\,\mathrm{d}x, \tag{41}$$

with $\sigma_k = |\lambda_k|\Delta t/\Delta x$, where $\lambda_k = \{u-c, u, u, u+c\}$, while $\Delta t$ and $\Delta x$ are the discretization steps in time and space, respectively, with the assumption that $\Delta x$ is constant in the computational domain.

The parabolic profile is defined by

$$q_i^I(x) = q_{R,i-1/2} + \xi(x)[q_{L,i+1/2} - q_{R,i-1/2} + q_{i,6}(1-\xi(x))] \tag{42}$$

with

$$q_{i,6} = 6q_i - 3(q_{R,i-1/2} + q_{L,i+1/2}) \tag{43}$$

and

$$\xi(x) = \frac{x - x_{i-1/2}}{\Delta x}, \quad x_{i-1/2} \leqslant x \leqslant x_{i+1/2}. \tag{44}$$

Substituting (43) into (40) and (41) leads to the explicit formulations

$$\mathscr{I}_{+}^{(k)}(q_i) = q_{L,i+1/2} - \tfrac{1}{2}\sigma_k\big[q_{L,i+1/2} - q_{L,i+1/2} - \big(1 - \tfrac{2}{3}\sigma_k\big)q_{i,6}\big], \tag{45}$$

$$\mathscr{I}_{-}^{(k)}(q_i) = q_{R,i-1/2} + \tfrac{1}{2}\sigma_k\big[q_{L,i+1/2} - q_{L,i+1/2} + \big(1 - \tfrac{2}{3}\sigma_k\big)q_{i,6}\big]. \tag{46}$$

**4.2.3.** *Characteristic tracing and flux reconstruction.* The next step is to extrapolate in time the integrals $\mathscr{I}_{\pm}^{(k)}$ to get the left and right edge states at time $n + \frac{1}{2}$. This procedure is complex, especially in multidimensions where transverse terms are taken into account; the complete detailed procedure can be found in [20]. In 1D, the left and right edge states are computed as

$$q_{L,i+1/2}^{n+1/2} = \mathscr{I}_{+}^{(k=u+c)} - \sum_{k:\lambda_k \geqslant 0} \beta_k \boldsymbol{l}_k \cdot [\mathscr{I}_{+}^{(k=u+c)} - \mathscr{I}_{+}^{(k)}]\boldsymbol{r}_k + \tfrac{1}{2}\Delta t\, S_i^n, \tag{47}$$

$$q_{R,i-1/2}^{n+1/2} = \mathscr{I}_{-}^{(k=u-c)} - \sum_{k:\lambda_k \leqslant 0} \beta_k \boldsymbol{l}_k \cdot [\mathscr{I}_{-}^{(k=u-c)} - \mathscr{I}_{-}^{(k)}]\boldsymbol{r}_k + \tfrac{1}{2}\Delta t\, S_i^n \tag{48}$$

where

$$\beta_k = \begin{cases} \frac{1}{2} & \text{if } \lambda_k = 0, \\ 1 & \text{otherwise,} \end{cases} \tag{49}$$

and $\boldsymbol{l}_k$ and $\boldsymbol{r}_k$ are the left row and right column of the matrices defined in (27) and (28) for each eigenvalue $k$. Note that here, $S_i^n$ represents any source terms at time $n$ to include in the characteristic tracing operation.

Finally, the time-centered fluxes are computed using an approximate Riemann problem solver. Here the HLLC algorithm [27] is employed. At the end of this procedure the primitive variables are centered in time at $n + \frac{1}{2}$, and in space at the edges of a cell. This is the so-called *Godunov state* and the convective fluxes can be computed to advance (22).

**4.3.** ***The hybrid PPM/WENO method.*** As will be shown in the results in Section 5, the PPM method presented above gives good results for a small computational time compared to the fourth-order finite-volume WENO strategy, which is costly. However, for fine meshes, the PPM method exhibits a significant pile-up of energy in the high-frequency range of the spectra, which is undesirable and limits mesh refinement. It has been found that the pile-up of energy at the high frequencies was sensitive to the slope-limiting procedure presented in Section 4.2.1. As many variants can be found in the literature, an attempt to tweak this procedure was made, for example by playing with the numerical parameters (see Appendix A) or by removing the slope-limiting operation completely. Also, the procedure given in [7] was tested. For all cases, the results were very similar and the impact on the pile-up of energy was modest and not satisfying.

After an intensive trial and error process, it became apparent that the interpolation and slope-limiting procedure described in Section 4.2.1 was not robust, leading to poor results in the high-frequency range. Here we consider replacing this whole procedure by a WENO interpolation.

Basically, the purpose of the hybrid PPM/WENO method is only to replace the procedure in Section 4.2.1, and $q^L_{i+1/2}$ and $q^R_{i-1/2}$ are instead given by (17). Then the PPM algorithm continues exactly the same as in Section 4.2.2.

As shown in Appendix C, as WENO-Z [5] appears to be the most robust and gives satisfying results for a small computational cost compared to other WENO methods, only the WENO-Z method is employed below, but it is emphasized that any other WENO reconstruction methods can be employed. For ease of exposition, the hybrid method will be called PPM/WENO in the remainder of the paper, but one has to keep in mind that the WENO-Z method has been used for the reconstruction at faces.

## 5. Results

The numerical methods presented in the previous section are tested and compared on three very different test cases. The first one is the convection of a smooth compressible vortex. This test case is chosen because it highlights the theoretical order of accuracy of the numerical methods. The second test case is the Shu–Osher problem, which represents the extreme opposite of the smooth vortex test case. The Shu–Osher problem is very difficult to solve numerically, because a shock wave is propagating in an oscillating entropy field, and the challenge is to capture the shock while resolving the phase and amplitude of the fluctuating entropy. As will be shown, all the methods perform correctly, but for all of them the rate of convergence collapses to first-order. The last test case is the decay of compressible homogeneous isotropic turbulence in the presence of eddy shocklets. This test case can be viewed as a combination of the two previous test cases, because it contains both shocks and discontinuities, as well as smooth turbulence structures that lie in a large-bandwidth turbulent spectrum. More specifically, this test case is representative of flows that are encountered in practical CFD applications (see [22] for an example). Note that in the remainder of this section, the initial solution comes from either an analytical solution or a synthetic manufactured solution. It is important to note that there is an averaging process over the volume to provide a consistent initial solution. For the fourth-order method, the procedure is slightly different to preserve a high order of accuracy: the solution is first expressed at the Gauss points, and the integration over the volume is performed via the quadrature rule.

**5.1.** ***2D convection of a smooth compressible vortex.*** The following test case consists of the convection of a 2D compressible vortex. This test case has been used frequently in the literature to assess the performance of outflow characteristic boundary conditions [21; 23]. The interest for this test case is that the solution is smooth and presents weak compressibility effects. Here, the vortex is convected in a periodic domain so as to accumulate numerical errors from the discretization schemes. For each numerical method, the same test case is simulated with increasing mesh resolution. The time step is computed based on the mesh resolution via a constraint on the CFL number, set to 0.7. At the end of a simulation, convergence is measured using the $\mathscr{L}^1$-norm of the difference of the $x$-velocity between the final computed solution and the analytical solution:

$$\varepsilon_u = \mathscr{L}^1_u(S_{\text{sol}} - S_{\text{ref}}) = \frac{\sum_1^N |u_{\text{sol}} - u_{\text{ref}}|}{N}, \tag{50}$$

where subscripts sol and ref identify the numerical solution and the initial solution, and $N$ is the number of computational cells.

The configuration is a single vortex superimposed on a uniform flow field along the $x$-direction. The stream function $\Psi$ of the initial vortex is given by

$$\Psi = \Gamma \exp\left(-\frac{r^2}{2R_v^2}\right), \tag{51}$$

where $r = \sqrt{(x - x_v)^2 + (y - y_v)^2}$ is the radial distance from the center of the vortex located at $[x_v, y_v]$, while $\Gamma$ and $R_v$ are the vortex strength and radius, respectively. The velocity field is then defined as

$$u = \frac{\partial \Psi}{\partial y} + u_0, \qquad v = -\frac{\partial \Psi}{\partial x}. \tag{52}$$

The initial pressure field is expressed as

$$p(r) = p_{\text{ref}} \exp\left(-\frac{\gamma}{2}\left(\frac{\Gamma}{cR_v}\right)^2 \exp\left(-\frac{r^2}{R_v^2}\right)\right), \tag{53}$$

and the corresponding density field is given by

$$\rho(r) = \frac{p(r)}{\mathscr{R}T_{\text{ref}}}, \tag{54}$$

where $T_{\text{ref}}$ is assumed constant. Note that here, $\gamma$ is the ratio of specific heats and is set to $\gamma = 1.4$.

The computational domain is a square of dimension $L = 0.01$ m. The reference temperature $T_{\text{ref}}$ and pressure $p_{\text{ref}}$ are set to 300 K and 101320 Pa, respectively. The vortex is located at $[x_v, y_v]$ = [0, 0], and its parameters are set to $\Gamma = 0.11\,\text{m}^2/\text{s}$ and $R_v = 0.1L$. The initial flow velocity is $u_0 = 100\,\text{m/s}$. In the present test case, only the Euler equations are solved. Thus, the transport coefficients $\eta$, $\varsigma$, and $\lambda$ in (1), (2), and (3) are set to zero.

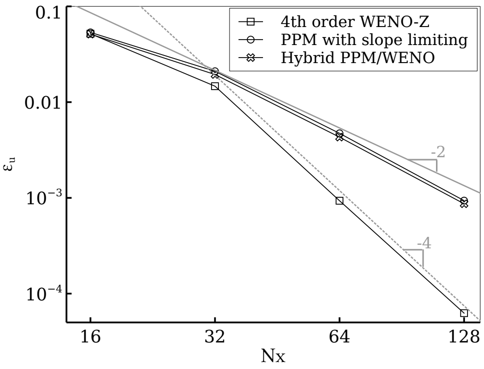


**Figure 1.** Convection of a vortex: evolution of the $\mathscr{L}_1$-norm of the error of the $x$-velocity for different mesh size $N_x$.

The simulations are performed over a physical time of 5 ms, corresponding to five flow through times (FTT), in order to accumulate enough numerical errors from the spatial discretization schemes.

Results are shown in Figure 1. The solid and dotted gray lines represent second- and fourth-order slopes, respectively. As expected, because the solution is smooth, all the numerical methods exhibit a convergence rate that follows their theoretical order of accuracy. The PeleC code (see Section 4) using a second-order Godunov method with either the PPM or the hybrid PPM/WENO method for interpolation presents an almost constant second-order convergence rate. The finite-volume WENO method of the RNS code exhibits fourth-order convergence. From the results depicted in Figure 1, it is clear that a high-order method is superior to a second-order numerical method, because for the same mesh resolution the numerical error of the solutions is significantly lower. However, this superiority is possible because the solution is smooth, and as will be shown below, this observation no longer holds when the solution features shocks and high gradients in the flow.

**5.2.** ***Shock-driven test case: the Shu–Osher problem.*** The so-called Shu–Osher test case simulates the one-dimensional propagation of a normal shock wave interacting with a fluctuating entropy wave, generating a flow field containing both small-scale structures as well as discontinuities. The initial conditions are given by

$$(\rho, u, p) = \begin{cases} (3.857143,\ 2.629369,\ 10.3333) & \text{if } x \leqslant 1, \\ (1+0.2\sin(5x),\ 0,\ 1) & \text{otherwise.} \end{cases} \tag{55}$$

The length of the computational domain is $x \in [0, 10]$, and the solution is advanced in time to $t = 1.2$. For all numerical methods investigated, the mesh is progressively refined from $N_x = 256$ to $N_x = 2048$. The convergence is measured using the $\mathscr{L}^1$-norm (see (50)) of the difference in density between the final computed solution and a reference solution defined to be the solution computed with the

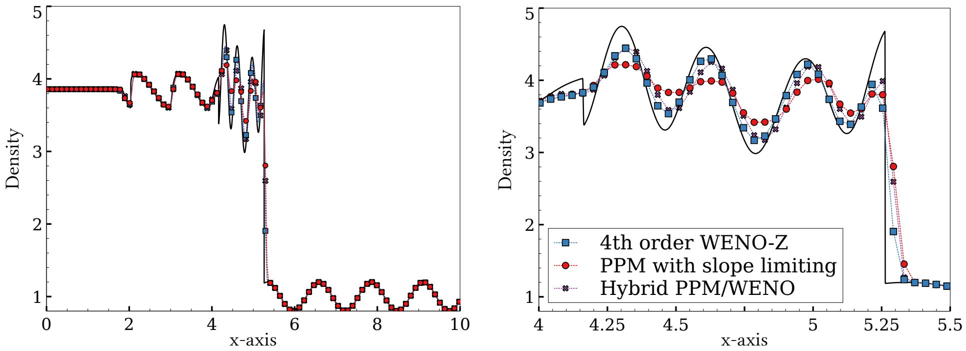


**Figure 2.** Shu–Osher test case: profile of density for $N_x = 256$. Left: full domain. Right: zoom.

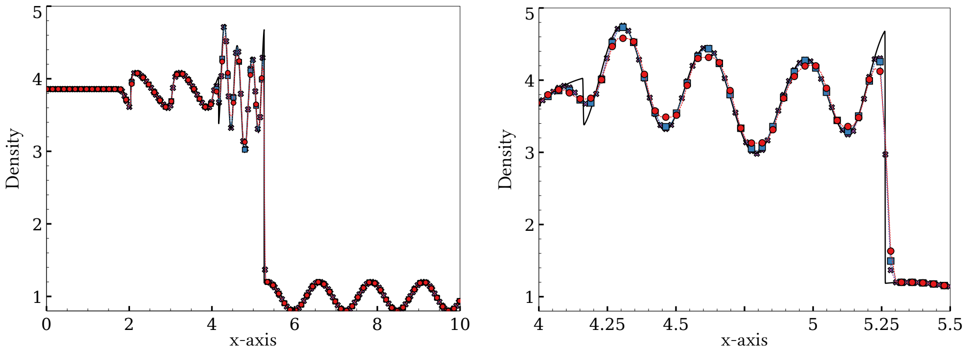


**Figure 3.** Shu–Osher test case: profile of density for $N_x = 512$. Left: full domain. Right: zoom.

second-order Godunov method with PPM interpolation and with a very fine mesh $N_x = 32768$. In all simulations the CFL number is set to 0.5.

The density fields at $t = 1.2$ computed with $N_x = 256$, 512, 1024, and 2048 are shown in Figures 2, 3, 4, and 5, respectively. In these figures, the blue square, red circle, and purple cross represent the fourth-order finite-volume WENO method with the WENO-Z variant, the original PPM method with slope limiting, and the hybrid PPM/WENO method developed in the present paper, respectively (see legend in Figure 2, right). Note that the left and right panels in Figures 2, 3, and 4 present the full domain and a zoom in the domain, respectively, while Figure 5 is only a zoom in the domain. Note also that there is no relation between the symbols and the number of grid points. Several symbols have been removed from the figures for clarity.

For a coarse mesh ($N_x = 256$), a close look at Figure 2, right, reveals that the fourth-order finite-volume WENO method is able to capture the correct phase of the waves, despite a damping of the amplitude. The second-order Godunov method with the original PPM interpolation and the slope-limiting procedure does not

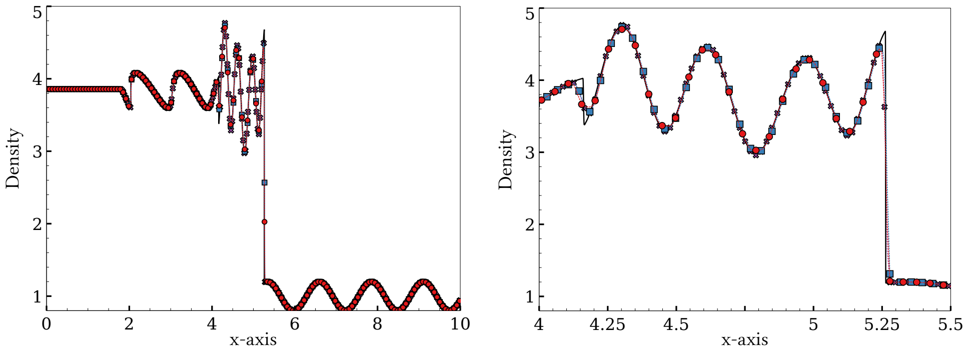


**Figure 4.** Shu–Osher test case: profile of density for $N_x = 1024$. Left: full domain. Right: zoom.

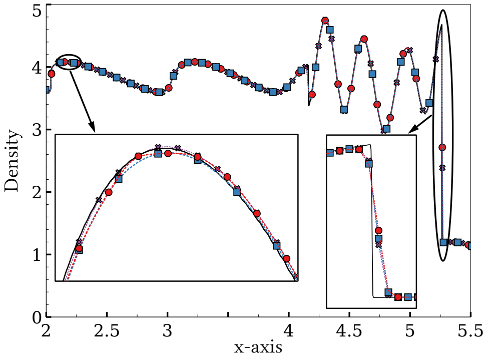


**Figure 5.** Shu–Osher test case: profile of density for $N_x = 2048$.

accurately capture the correct profile of density. However, the hybrid PPM/WENO method presents a profile very similar to the one captured by the fourth-order finite-volume method. It turns out that changing the slope-limiting procedure in the PPM method to the WENO interpolation makes the second-order Godunov method recover the correct profile of density. This can be explained by the fact that the shock is better resolved by the WENO interpolation and that the slope-limiting procedure introduces spurious wiggles in the density waves.

As seen in Figure 3, right, a mesh refinement by a factor of 2 makes all the methods accurately capture the phase of the density waves. However, the original PPM method with slope limiting (red circle symbols) shows a damping of the amplitude, while the hybrid PPM/WENO solution correctly captures both the phase and the amplitude, and is very close to the solution computed with the fourth-order finite-volume WENO method.

As the mesh is further refined, all the methods tend to collapse to the same solution. However, as can be seen in Figure 5 for a fine mesh ($N_x = 2048$), the fourth-order

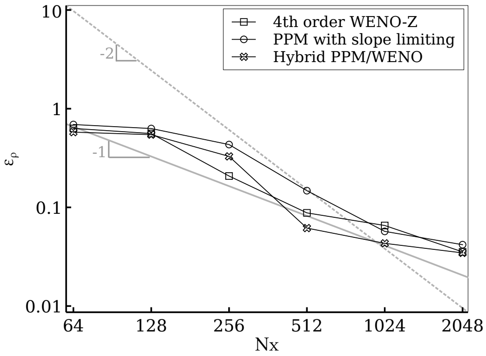


**Figure 6.** Shu–Osher test case: $\mathscr{L}^1$-norm of the error of the density.

| method | $\mathcal{O}(\varepsilon_\rho)$ |
|---|---|
| PPM with slope-limiting | 0.92 |
| fourth-order WENO-Z | 0.89 |
| hybrid PPM/WENO | 0.96 |

**Table 1.** Shu–Osher test case: convergence rate of the $\mathscr{L}^1$-norm of the error on the density.

finite-volume WENO method shows a slight damping of the amplitude of the density wave, whereas the second-order Godunov method with PPM interpolation and slope limiting exhibits some smooth high-frequency oscillations. The best solutions are the ones computed with the second-order Godunov method and the hybrid PPM/WENO method. The shape and amplitude of the density are closer to the reference solution.

Overall, it turns out that for this specific test case, the use of high-order methods is questionable. This is highlighted by the study of the convergence rate of the $\mathscr{L}^1$-norm of the error of the density profile. The error $\varepsilon_\rho$ is reported in Figure 6, and the convergence rate computed with a best-fitting curve method is reported in Table 1. It is obvious that all the numerical methods, either theoretically second- or fourth-order accurate, collapse to less than first-order accuracy because of the presence of the discontinuity. Overall, the present study suggests that reaching a correct approximation of a flow solution can be achieved by a second-order method and sufficient mesh resolution. In the following section, a more realistic three-dimensional compressible turbulent flow is simulated to investigate the capabilities of the second- and fourth-order numerical methods, as well as their effective cost in terms of mesh resolution, when both shocks and small turbulence structures interact in the same domain.

**5.3.** ***Three-dimensional isotropic compressible turbulence decay.*** The present test case consists of the simulation of the decay of a compressible isotropic turbulent field with the presence of eddy shocklets. Originally a physical study of turbulence in the work of Lee et al. [15], these simulations have become a framework to study the properties of numerical schemes to capture turbulence spectra and the decay of physical quantities. Here, the numerical setup described in [14] is reproduced.

The initial condition is built by generating a solenoidal velocity field $\boldsymbol{u}_0$ that satisfies

$$E(k) \sim k^4 \exp(-2(k/k_0)^2), \qquad \frac{3u_{\mathrm{rms},0}^2}{2} = \frac{\langle \boldsymbol{u}_0 \cdot \boldsymbol{u}_0 \rangle}{2} = \int_0^\infty E(k)\, \mathrm{d}k. \tag{56}$$

Here, $k_0$ is the most energetic wavenumber and is set to $k_0 = 4$. The simulation is controlled by two nondimensional parameters: the turbulent Mach number

$$M_{t,0} = \frac{\sqrt{\langle \boldsymbol{u}_0 \cdot \boldsymbol{u}_0 \rangle}}{c_0} \tag{57}$$

where $c_0$ is the sound speed in the initial solution, and the Taylor-scale Reynolds number defined as

$$\mathrm{Re}_{\psi,0} = \frac{\rho_0 \psi_0 u_{\mathrm{rms},0}}{\eta_0} \tag{58}$$

where

$$u_{\mathrm{rms},0} = \sqrt{\frac{\langle \boldsymbol{u}_0 \cdot \boldsymbol{u}_0 \rangle}{3}}, \qquad \psi_0 = \frac{2}{k_0}. \tag{59}$$

In the present simulation, $M_{t,0} = 0.6$ and $\mathrm{Re}_{\psi,0} = 100$. These values are set such that weak shock waves can develop spontaneously from the turbulent motions [14], and allow numerical convergence for relatively coarse mesh grids to keep the computational cost reasonable. Once $M_{t,0}$ and $\mathrm{Re}_{\psi,0}$ are set, $u_{\mathrm{rms},0}$ can be deduced from (57) with the known sound speed, and the viscosity $\eta_0$ can be deduced from (58). Unlike the simulations presented in [14], in the present study the viscosity is held constant throughout the simulation. Moreover, a constant thermal conductivity is set according to

$$\lambda_0 = \frac{\eta_0 C_p}{\mathrm{Pr}} \tag{60}$$

where $C_p$ is the specific heat capacity, set to $C_p = 1.173\,\mathrm{kJ/kg{\cdot}K}$ and the Prandtl number Pr is set to $\mathrm{Pr} = 0.71$. Moreover, the initial temperature and pressure in the flow are set to $T_0 = 1200\,\mathrm{K}$ and $p_0 = 1\,\mathrm{atm}$.

All the simulations are performed over a nondimensional time set to $t/\tau = 4$ where $\tau = \psi_0/u_{\mathrm{rms},0}$. Several mesh resolutions are investigated: $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$, and the CFL number is kept constant at 0.5. Note that the practical procedure to generate the velocity fields $\boldsymbol{u}_0$ is detailed in [14]. It is also important to note that the initial turbulent velocity fields are first generated on a grid of $N_x = 512$ and then integrated over each cell in the mesh. Moreover,

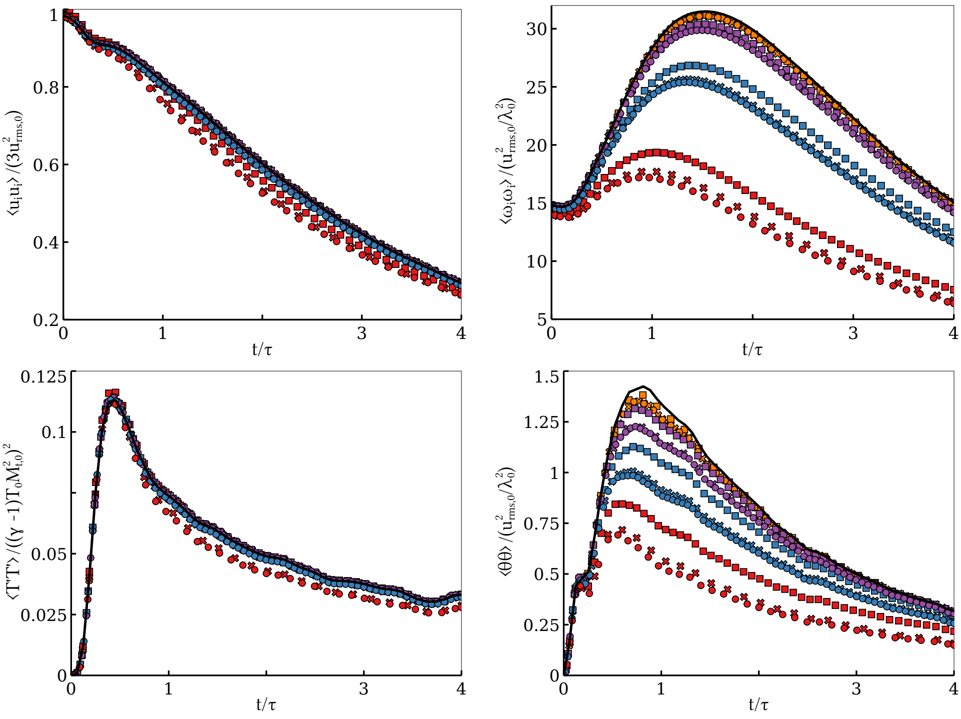


**Figure 7.** Time series of selected physical quantities for simulations performed with different mesh resolution and numerical methods. Legend is recalled in the text and in Figure 9. Top left: kinetic energy. Top right: enstrophy. Bottom left: temperature. Bottom right: dilatation, $\theta = \partial_j u_j$.

the initial solution is exactly the same for all simulations, regardless of the codes, numerical methods, or mesh grids employed.

In order to assess the performance of the second-order Godunov methods and the fourth-order finite-volume WENO method, a reference solution is generated with the very high-order code SMC [10] that employs eighth-order accurate centered finite-difference schemes for the spatial discretization, and a fourth-order Runge–Kutta algorithm for the time advancement. A convergence study for the reference solution is presented in Appendix B. This reference solution will be depicted with a black solid line in the remainder of the paper.

Figure 7, top left, top right, bottom left, and bottom right, presents the temporal evolution of the kinetic energy, the enstrophy, the variance of temperature, and the dilatation from $t = 0$ to $t/\tau = 4$. It can be seen that strong compressibility effects are generated quickly after the beginning of the simulation, suggesting the generation of eddy shocklets in the domain until $t/\tau \approx 0.5$. After $t/\tau \approx 1$, compressible shocks are no longer generated and they start to decay in a monotone way. Figure 8, top left, top right, bottom left, and bottom right, presents the spectra

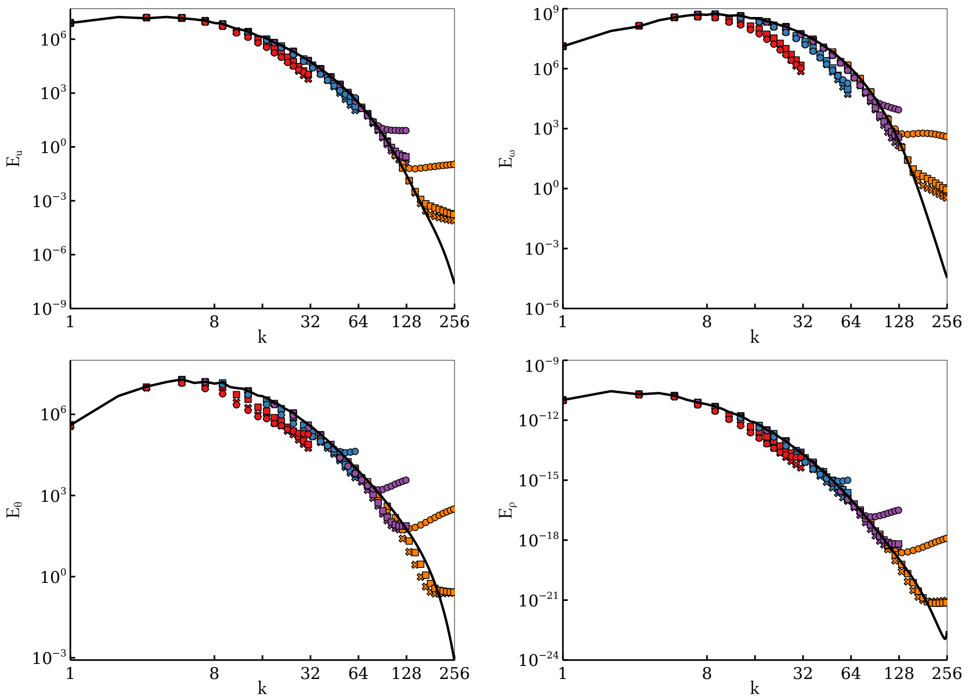


**Figure 8.** Spectra of selected physical quantities for simulations performed with different mesh resolution and numerical methods. Legend is recalled in the text and in Figure 9. Top left: kinetic energy. Top right: vorticity. Bottom left: dilatation. Bottom right: density.

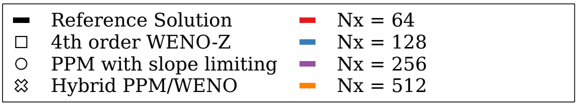


**Figure 9.** Symbols and color legend for Figures 7, 8, and 10.

taken at $t/\tau = 4$ for the kinetic energy, the vorticity, the dilatation, and the density. In these figures, the circle, cross, and square symbols represent second-order Godunov with PPM interpolation and slope limiting, the second-order Godunov method with the hybrid PPM/WENO procedure, and the fourth-order finite-volume WENO strategy, respectively. The red, blue, purple, and orange colors represent simulations performed with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$, respectively. It is emphasized that these figures contain a significant number of curves. For clarity, the legend is recalled in Figure 9 and a zoom on the high-end of the spectra for the kinetic energy is shown in Figure 10 for each mesh resolution. Note that the behavior of the numerical methods highlighted in Figure 10 is virtually the same for the spectra of other physical quantities.

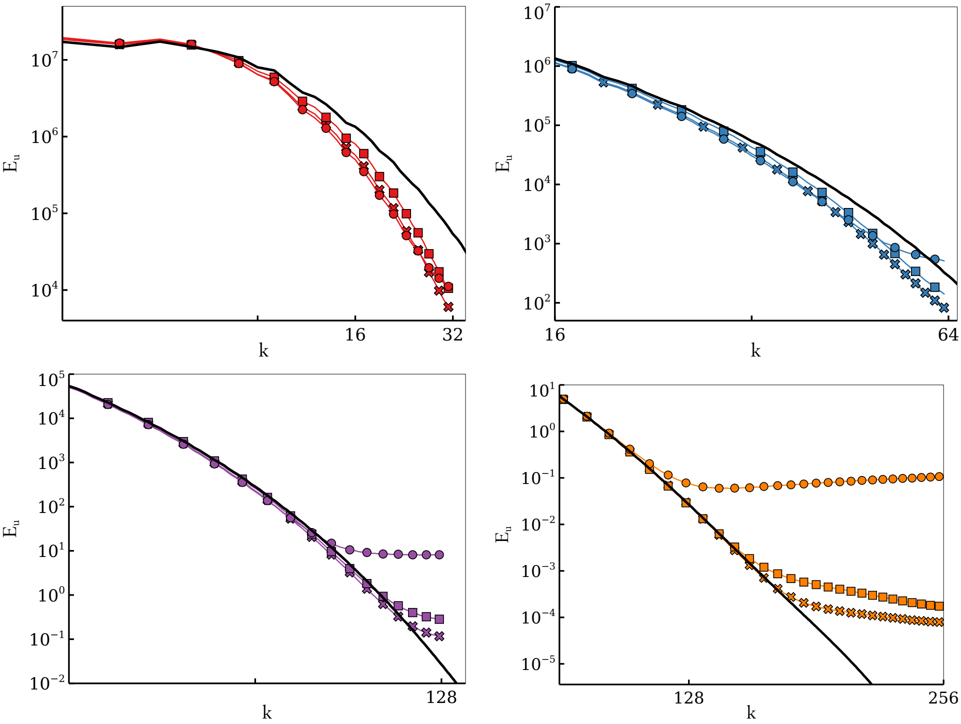


**Figure 10.** Zoom of the spectra of kinetic energy in Figure 8 for results computed with different mesh resolutions. Top left: $N_x = 64^3$. Top right: $N_x = 128^3$. Bottom left: $N_x = 256^3$. Bottom right: $N_x = 512^3$.

From the temporal evolution of physical quantities presented in Figure 7, it is clear that the second-order Godunov method, with either the PPM interpolation method with slope limiting or the hybrid PPM/WENO method, gives virtually the same results, with the exception of the very coarse mesh where some slight differences exist. In any case, for the same mesh resolution, the fourth-order finite-volume WENO method provides a better solution.

However, the spectra depicted in Figure 8 do not follow the same behavior as for the temporal series. Indeed, given a mesh resolution all the numerical methods give virtually the same spectra, but as the refinement of the mesh allows small turbulent structures to be resolved, it turns out that the different numerical methods do not perform equally in the high frequencies of the spectrum. As can be seen in Figure 10, whereas all methods present a pile-up of energy in the high-frequency range, the fourth-order finite-volume WENO method resolves the spectra with a monotonically decreasing energy, which is not the case for the second-order Godunov method with PPM interpolation and slope limiting. Most interesting, the second-order Godunov method with the hybrid PPM/WENO reconstruction method

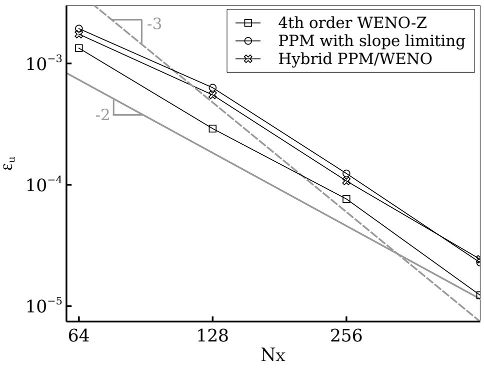


**Figure 11.** HIT test case: $\mathscr{L}^1$-norm of the error of the $x$-velocity.

| method | $\mathcal{O}(\varepsilon_u)$ |
|---|---|
| PPM with slope-limiting | 2.15 |
| fourth-order WENO-Z | 2.22 |
| hybrid PPM/WENO | 2.08 |

**Table 2.** HIT test case: convergence rate of the $\mathscr{L}^1$-norm of the error of the $x$-velocity.

is able to reproduce virtually the same spectra as the fourth-order finite-volume WENO method, meaning that replacing the slope-limiting procedure by the WENO reconstruction method recovers a monotone spectrum close to the reference solution.

Among these general trends, what emerges from all the figures is that for a given mesh resolution, the solutions are very close to each other regardless of the numerical method employed, with the exception of the high-end frequencies at fine mesh resolution. Such observations make sense, because as the turbulent Mach number is 0.6, the present 3D HIT test case can be seen as a mix between the Shu–Osher test case (see Section 5.2) where all the methods collapse to first-order, and the smooth solution test case presented in Section 5.1 where each numerical method follows its own theoretical order of convergence. This is highlighted by the study of the convergence rate with the $\mathscr{L}^1$-norm of the error on the $x$-velocity profile. The error $\varepsilon_u$ is reported in Figure 11 and the convergence rate computed with a best-fitting curve method is reported in Table 2. Overall, all the numerical methods present a second-order convergence rate. It is emphasized that this finding is the opposite of the conclusion in [1], where the fourth-order method always gives better results than the second-order one. This again makes sense, because the decay of turbulence investigated in [1] is simulated in an incompressible regime, leading to a solution that is always smooth. In that case, the findings of the study in [1] are consistent with the behavior shown in our study in Section 5.1, where

| method | nondimensional CPU time [s] |
|---|---|
| PPM with slope-limiting | $5.06 \times 10^{-3}$ |
| fourth-order WENO-Z | 1.1149 |
| hybrid PPM/WENO | $5.03 \times 10^{-3}$ |

**Table 3.** HIT computational time.

a smooth vortex is simulated and where all the numerical methods follow their theoretical order of convergence. Our study highlights that in the presence of strong compressibility effects, the theoretical expectations of a numerical method no longer hold because of the interaction with shocks.

All the results presented so far are investigations of the accuracy of the solutions, but another important parameter to take into account is the computational cost of each numerical method. As the PeleC and RNS codes are based on the AMReX framework, the profiling functionality of the library has been used to extract the actual computational cost to evaluate the hyperbolic terms in the set of governing equations. In practice, a timer has been put around the main routine called to compute the terms. Table 3 presents the average of the computational time for the evaluation of the routines involved in the computation of the hyperbolic convection term, divided by the number of calls during the whole simulation. This nondimensionalization is adopted here because the second-order Godunov procedure requires only one evaluation of the convection term, whereas the finite-volume WENO method is implemented with a Runge–Kutta time-integration procedure that requires many calls per time iteration. Also, the simulations are performed with the same mesh resolution of $N_x = 256$ and with the same parallelization over 512 MPI processes. It turns out that the fourth-order finite-volume WENO method is about 200 times more computationally expensive than the second-order Godunov method. For the Godunov method, the new hybrid PPM/WENO method proposed in the present paper has roughly the same computational cost as the original PPM method with slope limiting.

This significant difference can be explained by the number of interpolation procedures required for each cell and per time step. If we consider only one component in the system of equations, the PPM method requires only six interpolations in total (one per face), whereas for the high-order finite-volume method, the required number of interpolations is estimated with

$$2D(2^D - 1) \times 2 \tag{61}$$

where $D$ is the number of dimensions in the computational domain and the factor of 2 on the right-hand side stems from the number of Runge–Kutta stages. In three dimensions, achieving fourth-order accuracy requires fourteen times more interpolation procedures per cell than with the PPM algorithm, because data have to be evaluated through Gauss integration points. It is emphasized that this computational burden does not only depend on the interpolation procedures via the WENO schemes; to this count must be added the number of calls to the Riemann solver and all the conversions between conservative and primitive variables.

Overall, from the results presented in this section, it becomes apparent that an accurate representation of a compressible turbulent flow can be achieved faster with a second-order accurate Godunov method, together with the new hybrid PPM/WENO strategy for the reconstruction of physical values at faces that can achieve the same spectral resolution as a more complex and costly high-order method. Because the computational cost of the second-order Godunov method with PPM interpolation is significantly lower than that of the high-order finite-volume WENO method, it turns out that refining a simulation with the second-order method is still less costly than running a coarse high-order simulation. In this test case, it appears that the use of a fourth-order finite-volume WENO method is unnecessary in practice. The major finding of this study is that for finite-volume methods, the accuracy of the reconstruction of fluxes at cells interface has significantly more impact than the formal order of the method.

## 6. Conclusions

A comparison between low-order and high-order finite-volume methods has been performed on a series of test cases: the convection of a smooth 2D vortex, the Shu–Osher problem, and the decay of 3D homogeneous isotropic turbulence. The choice to assess the performance of finite-volume methods is justified by the fact that they are more robust and flexible to use in the context of simulations of industrial applications. The study focuses on the second-order Godunov method, as well as the fourth-order finite-volume WENO method. Results show that while on a smooth problem the high-order method performs better than the second-order one, when the solution contains a shock all the methods collapse to first-order accuracy. The study of the decay of compressible homogeneous isotropic turbulence with shocklets shows that the actual overall order of accuracy of the methods reduces to near second-order, despite the use of fifth-order reconstruction schemes. Most important, results in terms of turbulent spectra are similar regardless of the numerical methods employed, except for the higher end of the frequencies. Because our results show that the original PPM method with slope limiting fails to provide an accurate representation in the high-frequency range of the spectra, a novel hybrid PPM/WENO method is proposed. It is demonstrated that such a hybrid PPM/WENO method has the ability to capture the turbulent spectra with the accuracy of a formally high-order method, but at the cost of the second-order

Godunov method. Moreover, this study highlights that for finite-volume methods, the accuracy of the reconstruction of fluxes at cell interfaces has significantly more impact than the formal order of the method. Overall, the present study demonstrates the importance of evaluating the accuracy of a numerical method in terms of its actual spectral dissipation and dispersion properties on mixed smooth/shock cases, rather than by the theoretical formal order of the convergence rate.

## Appendix A: Slope-flattening procedure

In Section 4.2.1 a flattening limiter is imposed in (35) and (36) through a flattening coefficient $\chi_i$. The coefficient $\chi_i \in [0, 1]$, where $\chi_i = 1$ indicates that no additional limiting take place, whereas $\chi_i = 0$ means that the Godunov method is dropped to first-order accuracy. The computation of $\chi_i$ is performed as follows:

(1) First, a dimensionless measure of the shock resolution is computed with

$$\varsigma_i = \frac{p_{i+1} - p_{i-1}}{\max(p_{\text{small}}, |p_{i+2} - p_{i-2}|)} \tag{62}$$

where $p$ is the pressure and $p_{\text{small}}$ is a very small value to avoid a division by zero.

(2) Then the parameter $\tilde{\chi}_i$ is defined as

$$\tilde{\chi}_i = \min\{1, \max[0, a(\varsigma_i - b)]\} \tag{63}$$

where $a = 10$ and $b = 0.75$ are parameters set by the user. In order to confine $\tilde{\chi}_i$ in the range $[0, 1]$, $\tilde{\chi}_i = 0$ if either $u_{i+1} - u_{i-1} < 0$ or

$$\frac{p_{i+1} - p_{i-1}}{\min(p_{i+1}, p_{i-1})} \leqslant c \tag{64}$$

with $c$ a parameter set by the user, which takes the value of $c = \frac{1}{3}$ here.

(3) Finally $\chi_i$ is computed as

$$\chi_i = \begin{cases} 1 - \max(\tilde{\chi}_i, \tilde{\chi}_{i-1}) & \text{if } p_{i+1} - p_{i-1} > 0, \\ 1 - \max(\tilde{\chi}_i, \tilde{\chi}_{i+1}) & \text{otherwise.} \end{cases} \tag{65}$$

## Appendix B: Reference solution with the very high-order SMC code for the decay of homogeneous isotropic turbulence

In order to generate a reference solution, simulations are performed with the very high-order code SMC [10], which employs eighth-order accurate centered finite-difference schemes for the spatial discretization, and a fourth-order Runge–Kutta algorithm for the time advancement. Figure 12, top left, top right, bottom left, and bottom right, presents the temporal evolution of the kinetic energy, the enstrophy,

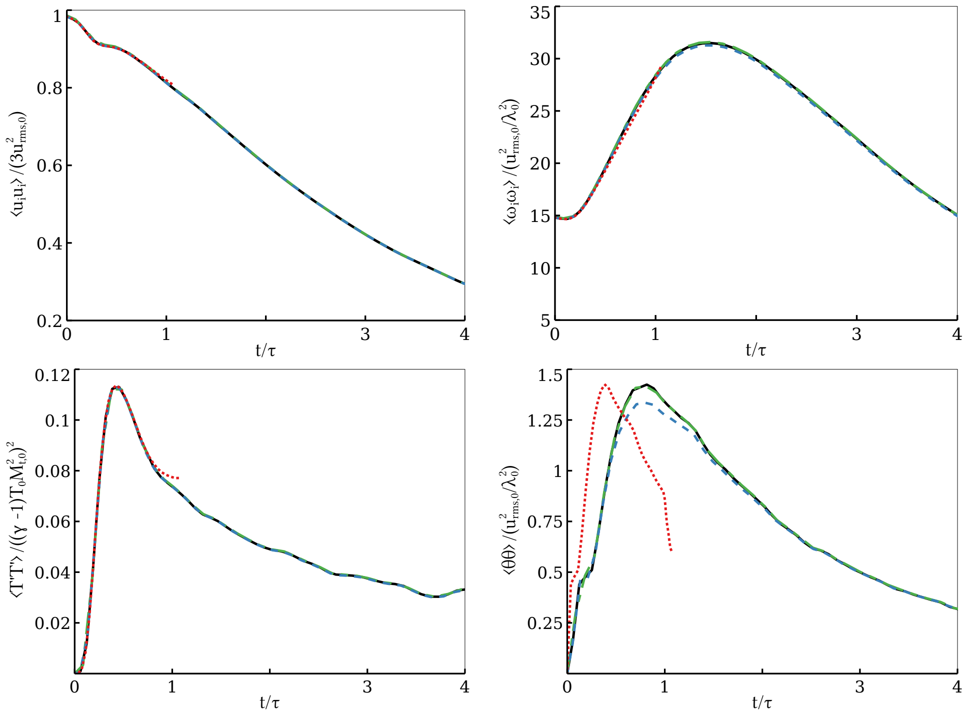


**Figure 12.** Temporal evolution of selected physical quantities for SMC simulations with different mesh resolution. The red dotted line, the blue dashed line, the green dashed-dotted line, and the black line represent the solutions computed on a mesh grid discretized with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$. Top left: kinetic energy. Top right: enstrophy. Bottom left: temperature. Bottom right: dilatation, $\theta = \partial_j u_j$.

the variance of temperature, and the dilatation from $t = 0$ to $t/\tau = 4$. Figure 13, top left, top right, bottom left, and bottom right, presents the spectra taken at $t/\tau = 4$ for the kinetic energy, the vorticity, the dilatation, and the density. In these figures, the red dotted line, the blue dashed line, the green dashed-dotted line, and the solid black line represent the solutions computed on a mesh grid discretized with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$, respectively. As can be seen in Figure 12, the simulation computed with $N_x = 64$ is unable to complete and crashes at approximately $t/\tau = 1$, because the mesh is too coarse to resolve the diffusion up to the Kolmogorov scale. The solution computed with $N_x = 512$ (solid black line) differs slightly from the one computed with $N_x = 256$, and is considered converged and will be used at the reference solution.

## Appendix C: WENO comparisons

As recalled in Section 3.2, a significant amount of schemes for the reconstruction of data at interfaces are based on the WENO paradigm. Indeed, the classical WENO-JS

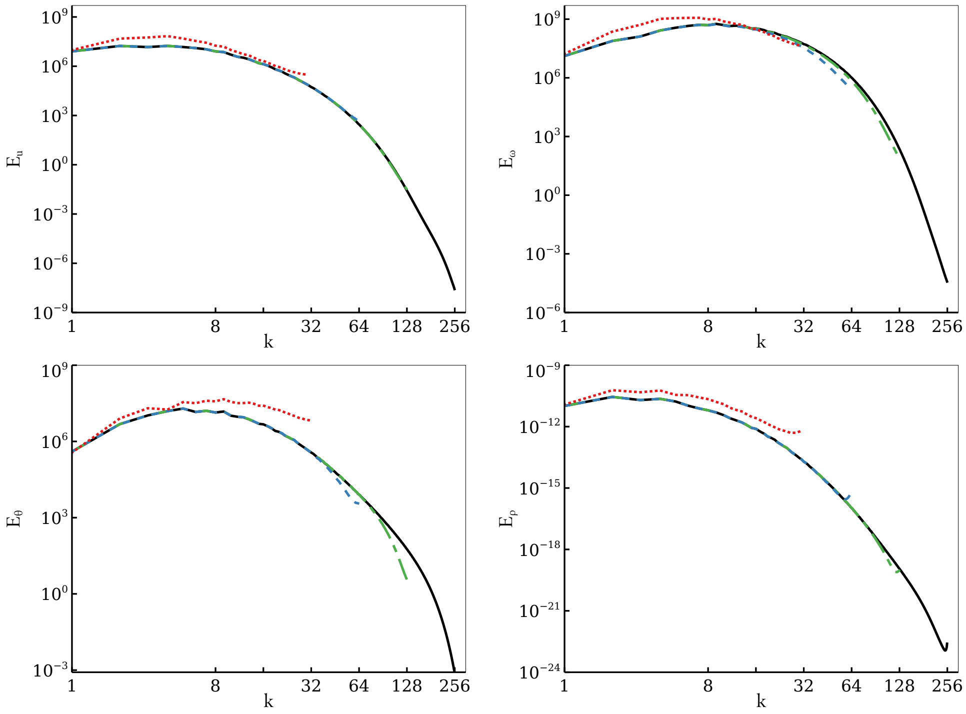


**Figure 13.** Spectra of selected physical quantities for SMC simulations with different mesh resolution. The red dotted line, the blue dashed line, the green dashed-dotted line, and the black line represent the solutions computed on a mesh grid discretized with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$. Top left: kinetic energy. Top right: vorticity. Bottom left: dilatation. Bottom right: density.

scheme is not optimal and is often considered too dissipative in smooth regions. Many variants have been developed to overcome such an issue. Among all of these variants, we have chosen to focus on the most popular ones to assess their robustness and performance on the test cases investigated in the present paper. A complete description of the variations introduced by these schemes is beyond the scope of the paper. So far, the WENO variants tested in this study are WENO-JS [13], WENO-M [12], WENO-Z [5], WENO-MDCD [18], and TENO [11].

**C.1.** ***Shu–Osher test case.*** The density at $t = 1.2$ computed with $N_x = 256$, 512, 1024, and 2048 is shown in Figures 14, 15, 16, and 17, respectively. In these figures, the blue diamond, green cross, purple square, orange plus, and maroon star symbols represent the WENO-JS, WENO-M, WENO-Z, WENO-MDCD, and TENO methods, respectively (see the legend in Figure 14, right). Note also that the left and right panels in Figures 14, 15, and 16 present the full domain and a zoom in the domain, respectively, while Figure 17 is only a zoom in the domain.

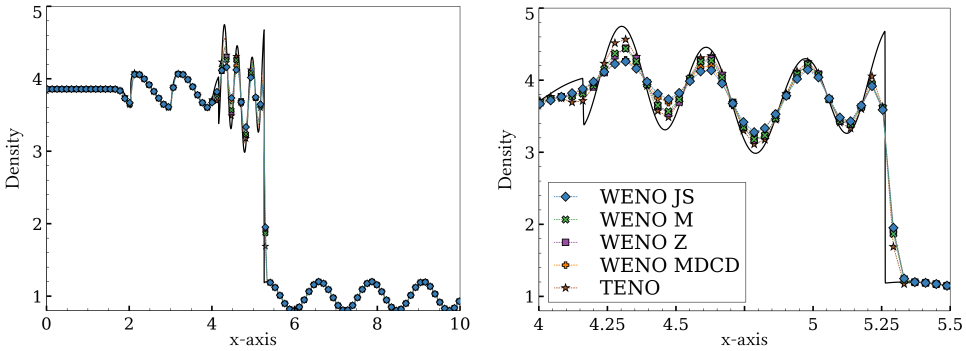


**Figure 14.** Shu–Osher test case: profile of density with PPM and WENO methods for $N_x = 256$. Left: full domain. Right: zoom.

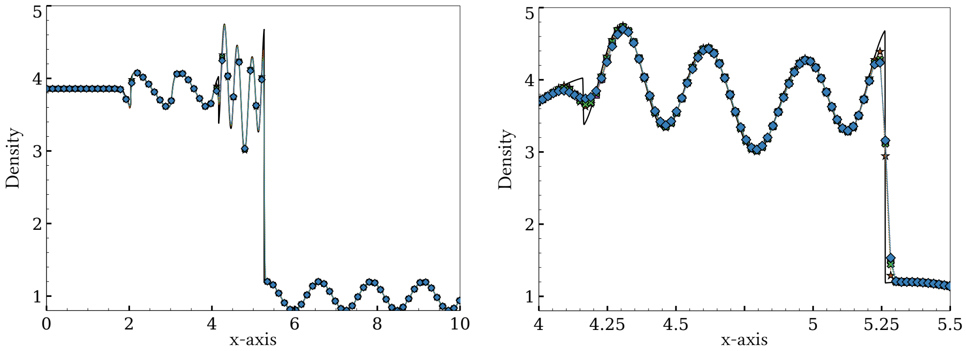


**Figure 15.** Shu–Osher test case: profile of density with PPM and WENO methods for $N_x = 512$. Left: full domain. Right: zoom.

For the coarse mesh, a close look at Figure 14 reveals that all the WENO variants reproduce the correct phase of the oscillation. The WENO-M, WENO-Z, and TENO methods give virtually similar results in terms of estimation of the amplitudes of the waves, while the WENO-JS and WENO-MDCD methods are equivalently the least accurate of the WENO variants. As shown in Figure 15, an increase of the mesh resolution by a factor of 2 leads all the WENO variants to virtually collapse to the same curve.

As shown in Figure 16, with another increase of the mesh resolution by a factor of 2, all the numerical methods investigated in the present study are virtually equivalent and very close to the reference solution computed on a very fine mesh. However, as can be seen at $x \approx 5.25$ after another increase of the mesh resolution by a factor of 2 in Figure 16, right, the TENO method provides an incorrect representation of the discontinuity. As shown in Figure 17, this trend becomes worse when the mesh is refined again by a factor of 2. As can be seen in the detailed

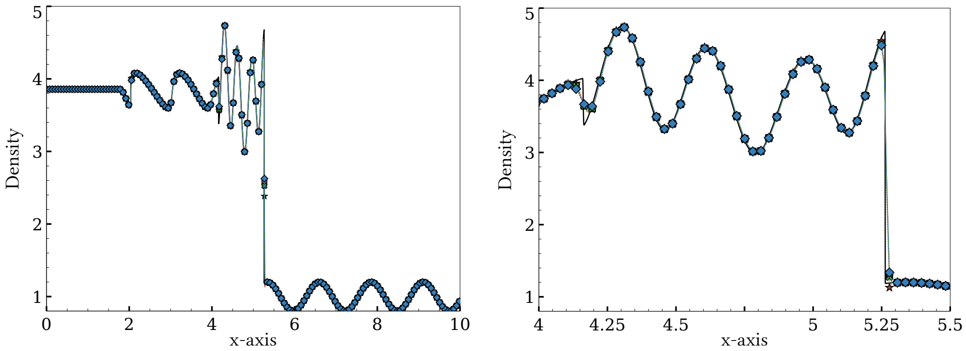


**Figure 16.** Shu–Osher test case: profile of density with PPM and WENO methods for $N_x = 1024$. Left: full domain. Right: zoom.

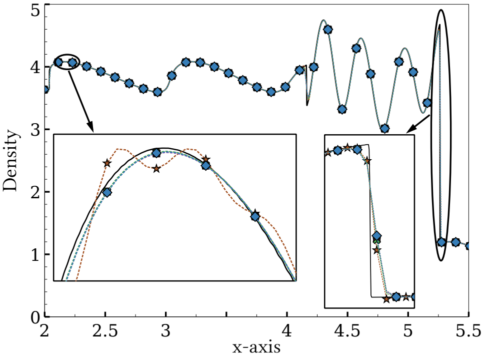


**Figure 17.** Shu–Osher test case: profile of density with PPM and WENO methods for $N_x = 2048$.

zoom, the solution computed with the TENO scheme shows large oscillations in the smooth regions. All other WENO variants are, however, robust.

This present study shows that when the mesh is small enough, it allows high-frequency waves to be resolved but small oscillations around discontinuities can appear and propagate, because the mesh is no longer coarse enough to filter them out. The most surprising result is the fact that the TENO variant, which appears to be a good choice on a coarse mesh, becomes the worst on a fine mesh. This can be attributed to the fact that the method fails to properly avoid the application of the central linear scheme in the region of large gradients. Furthermore, consistent with the convergence rate analysis performed at Section 5.2, it should be noted that all the WENO variants provide the same rate of convergence of the error, which is approximately $\mathbb{O}(0.9)$ here for this test case.

**C.2.** ***Decay of compressible isotropic turbulence.*** The decay of compressible isotropic turbulence is now simulated. Figure 18, top left, top right, bottom left, and

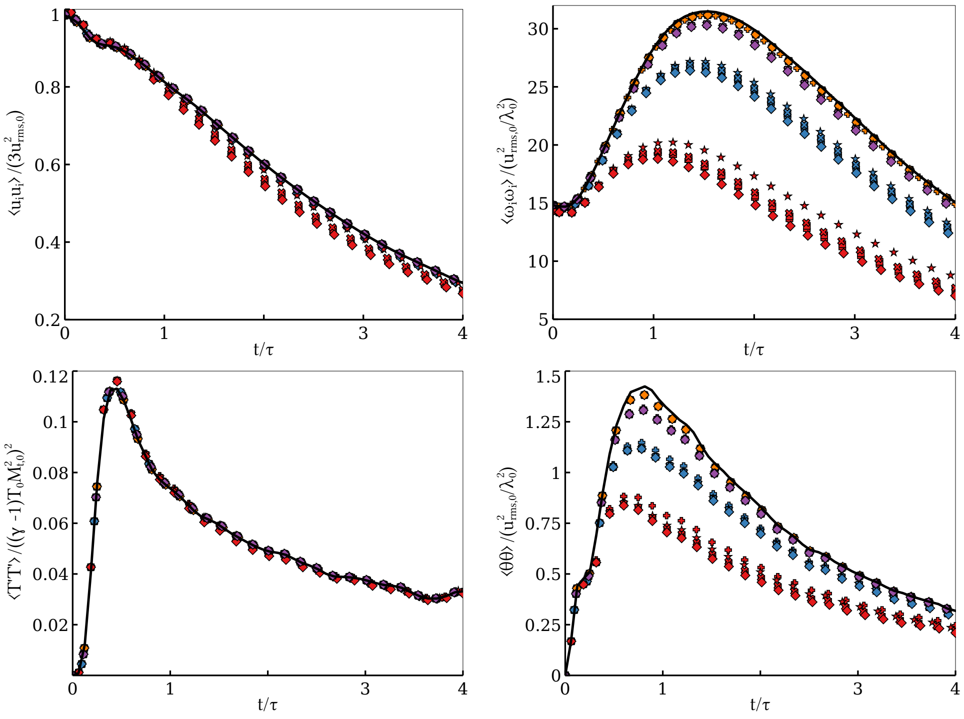


**Figure 18.** Time series of selected physical quantities for simulations performed with different WENO reconstruction schemes and with different mesh resolution. The diamond, cross, square, plus, and star symbols represent the WENO-JS, WENO-M, WENO-Z, WENO-MDCD, and TENO methods, respectively. The red, blue, purple and orange colors represent simulations performed with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$, respectively. Top left: kinetic energy. Top right: enstrophy. Bottom left: temperature. Bottom right: dilatation, $\theta = \partial_j u_j$.

bottom right, presents the temporal evolution of the kinetic energy, the enstrophy, the variance of temperature, and the dilatation from $t = 0$ to $t/\tau = 4$. Figure 19, top left, top right, bottom left, and bottom right, presents the spectra taken at $t/\tau = 4$ for the kinetic energy, the vorticity, the dilatation, and the density. In these figures, the diamond, cross, square, plus, and star symbols represent the WENO-JS, WENO-M, WENO-Z, WENO-MDCD, and TENO methods, respectively. The red, blue, purple, and orange colors represent simulations performed with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$, respectively. It is emphasized that these figures contain a significant number of curves. For clarity, a zoom on the high-end of the spectra of kinetic energy is shown in Figure 20 for each mesh resolution.

Overall, two general trends can be seen in Figures 18 and 19. For the temporal evolution of physical quantities, the different WENO variants investigated present significant differences when the mesh is coarse. However, when the mesh is refined,

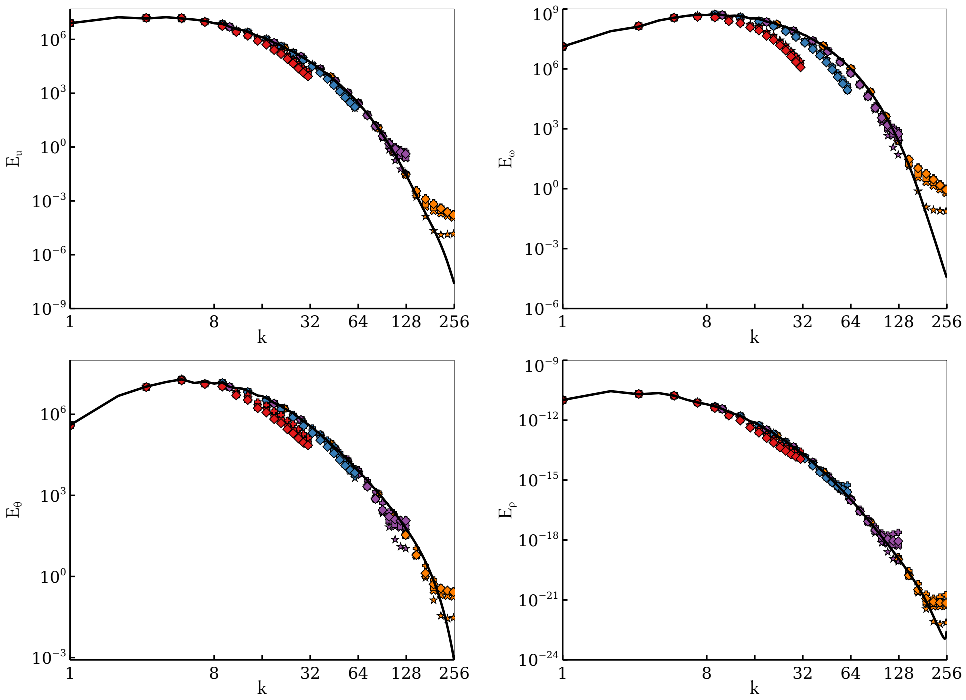


**Figure 19.** Spectra of selected physical quantities for simulations performed with different WENO reconstruction schemes and with different mesh resolutions. The diamond, cross, square, plus, and star symbols represent the WENO-JS, WENO-M, WENO-Z, WENO-MDCD, and TENO methods, respectively. The red, blue, purple, and orange colors represent simulations performed with $N_x = 64$, $N_x = 128$, $N_x = 256$, and $N_x = 512$, respectively. Top left: kinetic energy. Top right: vorticity. Bottom left: dilatation. Bottom right: density.

they quickly collapse to give similar results. However, as shown in Figure 19, all the different WENO variants give virtually the same spectra, at the exception of the very high frequencies of the spectrum when the mesh in refined enough to allow small turbulent structures to be resolved (see Figure 20). Similarly to the previous section, a convergence rate analysis has been performed and all the different WENO variants exhibit a second-order convergence rate. Furthermore, Figure 21 presents the computational time of each WENO variant. Recall here that the normalized CPU time is defined as the averaged wall clock time spent in the routines required for the computation of the hyperbolic terms, divided by the number of iterations performed during the simulation and the number of CPUs employed. It can be seen that the computational cost of the WENO-M variant is higher, followed by TENO, whereas WENO-JS, WENO-Z, and WENO-MDCD are virtually the same.

Based on these results, the present comparison study reveals that TENO is not robust enough to avoid instabilities near strong shocks (see Appendix C.1). Other

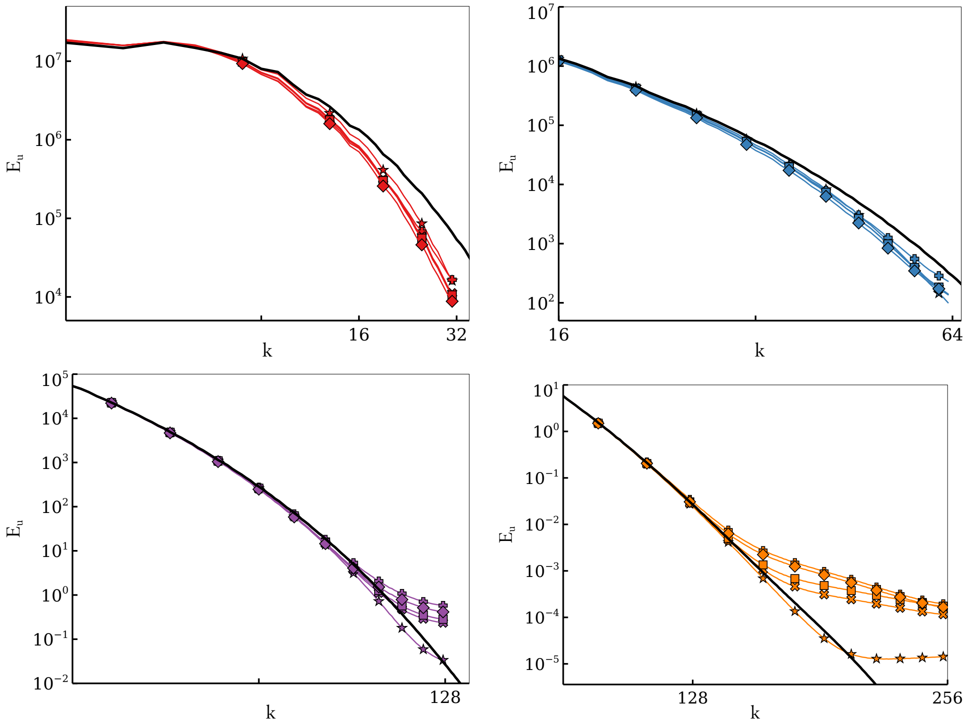


**Figure 20.** Zoom of the spectra of kinetic energy in Figure 19 for results computed with different mesh resolutions. Top left: $N_x = 64^3$. Top right: $N_x = 128^3$. Bottom left: $N_x = 256^3$. Bottom right: $N_x = 512^3$.

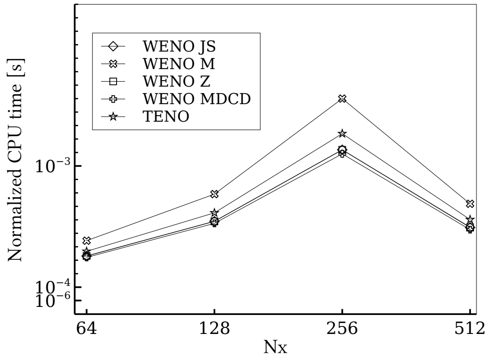


**Figure 21.** CPU time for only the convection term with different methods.

WENO variants are found to be robust, but WENO-M is more costly. Among the remaining variants, the WENO-Z scheme presents slightly better results, and is thus adopted as the best WENO reconstruction scheme for the whole study presented in this paper.

## Acknowledgments

The work here was supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under contract number DE-AC02005CH11231. John Wakefield has been funded by the National Science Foundation (NSF) Division of Mathematical Sciences (DMS) Mathematical Sciences Graduate Internship Program, administered by the Oak Ridge Institute for Science and Education (ORISE) through an interagency agreement between the DOE and NSF. ORISE is managed by ORAU under DOE contract number DE-SC0014664. All opinions expressed in this paper are the authors' and do not necessarily reflect the policies and views of the NSF, DOE, or ORAU/ORISE.

## References

[1] A. S. Almgren, A. J. Aspden, J. B. Bell, and M. L. Minion, *On the use of higher-order projection methods for incompressible turbulent flow*, SIAM J. Sci. Comput. **35** (2013), no. 1, B25–B42. MR Zbl

[2] A. S. Almgren, V. E. Beckner, J. B. Bell, M. S. Day, L. H. Howell, C. C. Joggerst, M. J. Lijewski, A. Nonaka, M. Singer, and M. Zingale, *CASTRO: a new compressible astrophysical solver, I: Hydrodynamics and self-gravity*, Astrophys. J. **715** (2010), no. 2, 1221–1238.

[3] G. M. Arshed and K. A. Hoffmann, *Minimizing errors from linear and nonlinear weights of WENO scheme for broadband applications with shock waves*, J. Comput. Phys. **246** (2013), 58–77. MR Zbl

[4] A. Aspden, N. Nikiforakis, S. Dalziel, and J. B. Bell, *Analysis of implicit LES methods*, Commun. Appl. Math. Comput. Sci. **3** (2008), 103–126. MR Zbl

[5] R. Borges, M. Carmona, B. Costa, and W. S. Don, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, J. Comput. Phys. **227** (2008), no. 6, 3191–3211. MR Zbl

[6] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin, *High-order, finite-volume methods in mapped coordinates*, J. Comput. Phys. **230** (2011), no. 8, 2952–2976. MR Zbl

[7] P. Colella and M. D. Sekora, *A limiter for PPM that preserves accuracy at smooth extrema*, J. Comput. Phys. **227** (2008), no. 15, 7069–7076. MR Zbl

[8] P. Colella and P. R. Woodward, *The Piecewise Parabolic Method (PPM) for gas-dynamical simulations*, J. Comput. Phys. **54** (1984), no. 1, 174–201. Zbl

[9] M. Emmett, E. Motheau, W. Zhang, M. Minion, and J. B. Bell, *A fourth-order adaptive mesh refinement algorithm for the multicomponent, reacting compressible Navier–Stokes equations*, Combust. Theory Model. **23** (2019), no. 4, 592–625. MR

[10] M. Emmett, W. Zhang, and J. B. Bell, *High-order algorithms for compressible reacting flow with complex chemistry*, Combust. Theory Model. **18** (2014), no. 3, 361–387. MR

[11] L. Fu, X. Y. Hu, and N. A. Adams, *A family of high-order targeted ENO schemes for compressible-fluid simulations*, J. Comput. Phys. **305** (2016), 333–359. MR Zbl

[12] A. K. Henrick, T. D. Aslamb, and J. M. Powers, *Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points*, J. Comput. Phys. **207** (2005), no. 2, 542–567. Zbl

[13] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. **126** (1996), no. 1, 202–228. MR Zbl

[14] E. Johnsen, J. Larsson, A. V. Bhagatwala, W. H. Cabot, P. Moin, B. J. Olson, P. S. Rawat, S. K. Shankar, B. Sjögreen, H. C. Yee, X. Zhong, and S. K. Lele, *Assessment of high-resolution methods for numerical simulations of compressible turbulence with shock waves*, J. Comput. Phys. **229** (2010), no. 4, 1213–1237. MR Zbl

[15] S. Lee, S. K. Lele, and P. Moin, *Eddy shocklets in decaying compressible turbulence*, Phys. Fluids **3** (1991), no. 4, 657–664.

[16] S. K. Lele, *Compact finite difference schemes with spectral-like resolution*, J. Comput. Phys. **103** (1992), no. 1, 16–42. MR Zbl

[17] R. J. LeVeque, *Finite volume methods for hyperbolic problems*, Cambridge University, 2002. MR Zbl

[18] M. P. Martín, E. M. Taylor, M. Wu, and V. G. Weirs, *A bandwidth-optimized WENO scheme for the effective direct numerical simulation of compressible turbulence*, J. Comput. Phys. **220** (2006), no. 1, 270–289. Zbl

[19] P. McCorquodale and P. Colella, *A high-order finite-volume method for conservation laws on locally refined grids*, Commun. Appl. Math. Comput. Sci. **6** (2011), no. 1, 1–25. MR Zbl

[20] G. H. Miller and P. Colella, *A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing*, J. Comput. Phys. **183** (2002), no. 1, 26–82. MR Zbl

[21] E. Motheau, A. Almgren, and J. B. Bell, *Navier–Stokes characteristic boundary conditions using ghost cells*, AIAA J. **55** (2017), no. 10, 3399–3408.

[22] E. Motheau, F. Nicoud, and T. Poinsot, *Mixed acoustic–entropy combustion instabilities in gas turbines*, J. Fluid Mech. **749** (2014), 542–576.

[23] T. J. Poinsot and S. K. Lele, *Boundary conditions for direct simulations of compressible viscous flows*, J. Comput. Phys. **101** (1992), no. 1, 104–129. MR Zbl

[24] C.-W. Shu, *High order WENO and DG methods for time-dependent convection-dominated PDEs: a brief survey of several recent developments*, J. Comput. Phys. **316** (2016), 598–613. MR Zbl

[25] V. A. Titarev and E. F. Toro, *Finite-volume WENO schemes for three-dimensional conservation laws*, J. Comput. Phys. **201** (2004), no. 1, 238–260. MR Zbl

[26] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*, 3rd ed., Springer, 2009. MR Zbl

[27] E. F. Toro, M. Spruce, and W. Speares, *Restoration of the contact surface in the HLL-Riemann solver*, Shock Waves **4** (1994), no. 1, 25–34. Zbl

[28] B. van Leer, *Towards the ultimate conservative difference scheme, V: A second-order sequel to Godunov's method*, J. Comput. Phys. **32** (1979), no. 1, 101–136. Zbl

[29] Z. J. Wang, K. Fidkowski, R. Abgrall, and et al., *High-order CFD methods: current status and perspective*, Internat. J. Numer. Methods Fluids **72** (2013), no. 8, 811–845. MR

[30] R. Zhang, M. Zhang, and C.-W. Shu, *On the order of accuracy and numerical performance of two classes of finite volume WENO schemes*, Commun. Comput. Phys. **9** (2011), no. 3, 807–827. MR Zbl

Received February 13, 2019. Revised December 9, 2019.

EMMANUEL MOTHEAU: emotheau@lbl.gov
*Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory, Berkeley, CA, United States*

JOHN WAKEFIELD: jwake@umich.edu
*Department of Mathematics, University of Michigan, Ann Arbor, MI, United States*

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 1, 2020
dx.doi.org/10.2140/camcos.2020.15.37

# A STOCHASTIC VERSION OF STEIN VARIATIONAL GRADIENT DESCENT FOR EFFICIENT SAMPLING

LEI LI, YINGZHOU LI, JIAN-GUO LIU, ZIBU LIU AND JIANFENG LU

We propose in this work RBM-SVGD, a stochastic version of the Stein variational gradient descent (SVGD) method for efficiently sampling from a given probability measure, which is thus useful for Bayesian inference. The method is to apply the random batch method (RBM) for interacting particle systems proposed by Jin et al. to the interacting particle systems in SVGD. While keeping the behaviors of SVGD, it reduces the computational cost, especially when the interacting kernel has long range. We prove that the one marginal distribution of the particles generated by this method converges to the one marginal of the interacting particle systems under Wasserstein-2 distance on fixed time interval $[0, T]$. Numerical examples verify the efficiency of this new version of SVGD.

## 1. Introduction

The empirical measure with samples from some probability measure (which might be known up to a multiplicative factor) has many applications in Bayesian inference [5; 3] and data assimilation [17]. A class of widely used sampling methods is the Markov chain Monte Carlo (MCMC) methods, where the trajectory of a particle is given by some constructed Markov chain with the desired distribution invariant. The trajectory of the particle is clearly stochastic, and the Monte Carlo methods take effect slowly for small number of samples. Unlike MCMC, the Stein variational gradient method (proposed by Liu and Wang in [20]) belongs to particle-based variational inference sampling methods (see also [22; 9]). These methods update particles by solving optimization problems, and each iteration is expected to make progress. As a nonparametric variational inference method, SVGD gives a deterministic way to generate points that approximate the desired probability distribution by solving an ODE system. Suppose that we are interested in some target probability distribution with density $\pi(x) \propto \exp(-V(x))$ ($x \in \mathbb{R}^d$). In SVGD, one sets $V = -\log \pi$, chooses some symmetric positive definite kernel $\mathcal{K}(x, y)$,

*MSC2010:* 62D05, 65C35.
*Keywords:* random batch method, RBM-SVGD, nonparametric variational inference, KL divergence, reproducing kernel Hilbert space, MCMC.

and solves the following ODE system for given initial points $\{X_i(0)\}_{i=1}^N$ [20; 19]:

$$\dot{X}_i = \frac{1}{N}\sum_{j=1}^{N} \nabla_y \mathcal{K}(X_i, X_j) - \frac{1}{N}\sum_{j=1}^{N} \mathcal{K}(X_i, X_j)\nabla V(X_j), \quad i = 1, \ldots, N, \quad (1\text{-}1)$$

where $N$ is the number of particles for the sampling purpose. The subindex "$y$" in $\nabla_y$ means that the gradient is taken with respect to the second variable in $\mathcal{K}(\cdot\,,\cdot)$; i.e., $\nabla_y \mathcal{K}(X_i, X_j) := \nabla_y \mathcal{K}(x, y)|_{(x,y)=(X_i,X_j)}$. When $t$ is large enough, the empirical measure constructed using $\{X_i(t)\}_{i=1}^N$ is expected to be close to $\pi$, i.e.,

$$\frac{1}{N}\sum_{i=1}^{N} \delta(x - X_i(t)) \approx \pi(x)\,dx, \quad t \gg 1.$$

Below, in Section 2, we will explain why this is expected to be true. Theoretic understanding of (1-1) is limited. For example, the convergence of the particle system (1-1) is still open. Recently, there have been a few attempts at understanding the limiting mean field PDE [19; 21]. In particular, Lu et al. [21] showed the convergence of the mean field PDE to the desired measure $\pi$.

In practice, SVGD seems to perform quite well, better compared with some typical Monte Carlo methods in some examples [19; 10]. It provides consistent estimation for generic distributions as Monte Carlo methods do, but with fewer samples. SVGD seems to be more efficient than some Monte Carlo methods in the particle level for approximating the desired measure, when the number of particles is small. Interestingly, it reduces to the maximum a posterior (MAP) method when $N = 1$ [20].

Though (1-1) behaves well when the particle number $N$ is not very big, one sometimes still needs an efficient algorithm to simulate (1-1). For example, when the dimension of the problem is not very high, in a typical MCMC method, the number of particles is several millions, or $N \approx 10^6$, while in SVGD, one may have $N \approx 10^3$. Simulating (1-1) needs $O(N^2)$ work to compute the interactions for each iteration, especially for interaction kernels that are not superlocalized or particles that are not sparse. In fact, for such situations, to compute the interaction force for one particle, one must consider all the other $N - 1$ particles to have enough accuracy. There are $N$ particles, so one must consider $O(N^2)$ interactions, which yields the $O(N^2)$ complexity for one iteration. Though $N \approx 10^2$–$10^3$ is not large, the $O(N^2)$ complexity makes the cost of SVGD for these cases comparable with MCMC with larger number of particles. Hence, it is highly motivated to develop a cheap version of SVGD.

In this work, we propose RBM-SVGD, a stochastic version of SVGD for sampling from a given probability measure. The idea is very natural: we apply the random batch method in [16] to the interacting particle system (1-1). Note that in the random batch method, the "batch" refers to the set for computing the interaction forces, not to be confused with the "batch" of samples for computing gradient as in stochastic gradient descent (SGD). Of course, if $V$ is the loss function corresponding to many samples, or the probability density in Bayesian inference corresponding to many observed data, the data-mini-batch idea can be used to compute $\nabla V$ in SVGD as well [20]. With the random batch idea for computing interaction, the complexity for each iteration now is only $O(N)$. Moreover, it inherits the advantages of SVGD (i.e., efficient for sampling when the number of particles is not large) since the random batch method is designed to approximate the particle system directly. In fact, we will prove that the one marginal of the random batch method converges to the one marginal of the interacting particle systems under Wasserstein-2 distance on a fixed time interval $[0, T]$. Note that the behavior of randomness in RBM-SVGD is different from that in MCMC. In MCMC, the randomness is required to ensure that the desired probability is invariant under the transition. The randomness in RBM-SVGD is simply due to the batch for computing the interaction forces, which is mainly for speeding up the computation. Though this randomness is not essential for sampling from the invariant measure, it may have other benefits. For example, it may lead to better ergodic properties for the particle system.

## 2. Mathematical background of SVGD

We now give a brief introduction to the SVGD proposed in [20] and provide some discussions. The derivation here is a continuous counterpart of that in [20].

Assume that random variable $X \in \mathbb{R}^d$ has density $p_0(x)$. Consider some mapping $\mathcal{T} : \mathbb{R}^d \to \mathbb{R}^d$, and we denote the distribution of $\mathcal{T}(X)$ by $p := \mathcal{T}_{\#} p_0$, which is called the push-forward of $p_0$ under $\mathcal{T}$. The goal is to make $\mathcal{T}_{\#} p_0$ closer to $\pi(x)$ in some sense. The way to measure the closeness of measures in [20] is taken to be the Kullback–Leibler (KL) divergence, which is also known as the relative entropy, defined by

$$\mathrm{KL}(\mu \parallel \nu) = \mathbb{E}_{Y\sim\mu} \log\left(\frac{d\mu}{d\nu}(Y)\right), \tag{2-1}$$

where $\frac{d\mu}{d\nu}$ is the well known Radon–Nikodym derivative. In [20, Theorem 3.1], it is shown that the Gateaux differential of $\mathcal{T} \mapsto G(\mathcal{T}) := \mathrm{KL}(p \parallel \pi)$ is given by

$$\left\langle \frac{\delta G}{\delta \mathcal{T}}, \phi \right\rangle = -\mathbb{E}_{Y\sim p} S_\pi \phi(Y) \quad \text{for all } \phi \in C_c^\infty(\mathbb{R}^d; \mathbb{R}^d) \tag{2-2}$$

where $S_q$ associated with a probability density $q$ is called the Stein operator given by

$$S_q \phi(x) = \nabla(\log q(x)) \cdot \phi(x) + \nabla \cdot \phi(x). \tag{2-3}$$

In fact, using the formula

$$\frac{d}{d\epsilon}(\mathcal{T}+\epsilon\phi\circ\mathcal{T})_{\#}p_0|_{\epsilon=0}=\frac{d}{d\epsilon}(I+\epsilon\phi)_{\#}p|_{\epsilon=0}=-\nabla\cdot(p\phi)=-pS_p\phi, \qquad (2\text{-}4)$$

and $\frac{\delta\,\mathrm{KL}(p\|\pi)}{\delta p}=\log p+1-\log\pi$, one finds

$$\left\langle\frac{\delta G}{\delta\mathcal{T}},\phi\right\rangle=\left\langle\frac{\delta\,\mathrm{KL}(p\parallel\pi)}{\delta p},-\nabla\cdot(p\phi)\right\rangle=-\int_{\mathbb{R}^d}pS_\pi\phi\,dx. \qquad (2\text{-}5)$$

The quantity $\langle\frac{\delta G}{\delta\mathcal{T}},\phi\rangle$ can be understood as the directional derivative of $G(\,\cdot\,)$ in the direction given by $\phi$. The paring in the second term above is in $L^2(\mathbb{R}^d)$ sense.

Based on this calculation, we now consider a continuously varying family of mappings $\mathcal{T}_\tau$ with $\tau\ge 0$ and

$$\frac{d}{d\tau}\mathcal{T}_\tau=\phi_\tau\circ\mathcal{T}_\tau.$$

Here, "$\circ$" means composition, i.e., for any given $x$, $\frac{d}{d\tau}\mathcal{T}_\tau(x)=\phi_\tau(\mathcal{T}_\tau(x))$. In this sense $x\mapsto X(\tau;x):=\mathcal{T}_\tau(x)$ is the trajectory of $x$ under this mapping; $x$ can be viewed as the so-called Lagrangian coordinate as in fluid mechanics while $\phi_\tau$ is the flow field. We denote

$$p_\tau:=(\mathcal{T}_\tau)_{\#}p_0. \qquad (2\text{-}6)$$

The idea is then to choose $\phi_\tau$ such that the functional $\tau\mapsto G(\mathcal{T}_\tau)$ decays as fast as possible. Note that to optimize the direction, we must require the field to have bounded magnitude $\|\phi_\tau\|_H\le 1$, where $H$ is some subspace of the functions defined on $\mathbb{R}^d$. The optimized curve $\tau\mapsto\mathcal{T}_\tau$ is a constant-speed curve (in some manifold). Hence, the problem is reduced to the optimization problem

$$\sup\{\mathbb{E}_{Y\sim p}S_\pi\phi(Y)\mid\|\phi\|_H\le 1\}. \qquad (2\text{-}7)$$

It is observed in [20] that this optimization problem can be solved by a convenient closed formula if $H$ is the so-called (vector) reproducing kernel Hilbert space (RKHS) [1; 2]. A (scalar) RKHS is a Hilbert space, denoted by $\mathcal{H}$, consisting of functions defined on some space $\Omega$ (in our case $\Omega=\mathbb{R}^d$) such that the evaluation function $f\mapsto E_x(f):=f(x)$ is continuous for all $x\in\Omega$. There thus exists $k_x\in\mathcal{H}$ such that $E_x(f)=\langle f,k_x\rangle_{\mathcal{H}}$. Then the kernel $\mathcal{K}(x,y):=\langle k_x,k_y\rangle_{\mathcal{H}}$ is symmetric and positive definite, meaning that $\sum_{i=1}^n\sum_{j=1}^n\mathcal{K}(x_i,x_j)c_ic_j\ge 0$ for any $x_i\in\Omega$ and $c_i\in\mathbb{R}$. Reversely, given any positive definite kernel, one can construct a RKHS consisting of functions $f(x)$ of the form $f(x)=\int\mathcal{K}(x,y)\psi(y)\,d\mu(y)$ where $\mu$ is some suitably given measure on $\Omega$. For example, if $\mu$ is the counting measure, choosing $\psi(y)=\sum_{j=1}^\infty a_j1_{x_j}(y)$ ($a_j\in\mathbb{R}$) can recover the form of RKHS in [20]. All such constructions yield isomorphic RKHS as guaranteed by the Moore–Aronszajn theorem [1]. Now, consider a given $\mu$ and $H = \mathcal{H}^d$ to be the vector RKHS:

$$H = \left\{ f = \int_{\mathbb{R}^d} \mathcal{K}(\,\cdot\,, y)\psi(y)\, d\mu(y) \,\middle|\, \psi : \mathbb{R}^d \to \mathbb{R}^d, \right.$$
$$\left. \iint_{\mathbb{R}^d \times \mathbb{R}^d} \mathcal{K}(x, y)\psi(x) \cdot \psi(y)\, d\mu(x)\, d\mu(y) < \infty \right\}.$$

The inner product is defined as

$$\begin{aligned}\langle f^{(1)}, f^{(2)} \rangle_H &= \iint_{\mathbb{R}^d \times \mathbb{R}^d} \mathcal{K}(x, y)\psi^{(1)}(x) \cdot \psi^{(2)}(y)\, d\mu(x)\, d\mu(y) \\ &= \sum_{j=1}^{d} \iint_{\mathbb{R}^d \times \mathbb{R}^d} \mathcal{K}(x, y)\psi_j^{(1)}(x)\psi_j^{(2)}(y)\, d\mu(x)\, d\mu(y). \qquad (2\text{-}8)\end{aligned}$$

This inner product therefore induces a norm $\|f\|_H = \sqrt{\langle f, f \rangle_H}$. Clearly, $H$ consists of functions with $\|\cdot\|_H$ to be finite. The optimization problem (2-7) can be solved by the Lagrange multiplier method

$$\mathcal{L} = \int_{\mathbb{R}^d} (S_\pi \phi) p_\tau(y)\, dy - \lambda \iint_{\mathbb{R}^d \times \mathbb{R}^d} \mathcal{K}(x, y)\psi(x) \cdot \psi(y)\, d\mu(x)\, d\mu(y),$$

where $dy$ means Lebesgue measure and $\phi(x) = \int_{\mathbb{R}^d} \mathcal{K}(x, y)\psi(y)\, d\mu(y)$. Using $\frac{\delta \mathcal{L}}{\delta \phi} = 0$, we find

$$2\lambda\phi = \int_{\mathbb{R}^d} \mathcal{K}(x, y)(S_\pi^* p_t)(y)\, dy =: \mathcal{V}(p_t), \qquad (2\text{-}9)$$

where $S_\pi^*$ is given by

$$S_\pi^*(f) = f(y)\nabla(\log \pi) - \nabla f(y) = -f(y)\nabla V(y) - \nabla f(y). \qquad (2\text{-}10)$$

The ODE flow

$$\frac{d}{d\tau}\mathcal{T}_\tau = \frac{1}{2\lambda(\tau)}\mathcal{V}(p_\tau) \circ \mathcal{T}_\tau$$

gives the constant-speed optimal curve, so that the velocity is the unit vector in $H$ along the gradient of $G$. Reparametrizing the curve $t = t(\tau)$ so that $\frac{d\tau}{dt} = 2\lambda$ and denoting $\rho_t := p_{\tau(t)}$, then

$$\frac{d}{dt}\mathcal{T}_t = \mathcal{V}(\rho_t) \circ \mathcal{T}_t. \qquad (2\text{-}11)$$

Clearly, the curve of $\mathcal{T}_t$ is not changed by this reparametrization. Using (2-4), one finds that $\rho$ satisfies the equation

$$\partial_t \rho = -\nabla \cdot (\mathcal{V}(\rho)\rho) = \nabla \cdot (\rho \mathcal{K} * (\rho \nabla V + \nabla \rho)). \qquad (2\text{-}12)$$

Here, $\mathcal{K} * f(x) := \int \mathcal{K}(x, y) f(y)\, dy$. It is easy to see that $\exp(-V)$ is invariant under this PDE. According to the explanation here, the right-hand side gives the optimal decreasing direction of KL divergence if the transport flow is measured by RKHS. Hence, one expects it to be the negation of gradient of KL divergence in the manifold of probability densities with metric defined through RKHS. Indeed, Liu made the first attempt to justify this in [19, §3.4].

The above theory has a little trouble for empirical measures because the KL divergence is simply infinity. For empirical measure, $\nabla\rho$ must be in the distributional sense. The good thing for RKHS is that we can move the gradient from $\nabla\rho$ onto the kernel $\mathcal{K}(x, y)$ so that the flow (2-11) becomes (1-1), which makes perfect sense. In fact, if (1-1) holds, the empirical measure is a measure solution to (2-12) (by testing on smooth function $\varphi$) [21, Proposition 2.5]. Hence, the ODE system is justified in this level, and one expects that (1-1) will give an approximation for the desired density. The numerical tests in [20] indeed justify this expectation. In this sense, the ODE system is formally a gradient flow of KL divergence, though the KL divergence functional is infinity for empirical measures.

Typical examples of $\mathcal{K}(x, y)$ include $\mathcal{K}(x, y) = (\alpha x \cdot y + 1)^m$, Gaussian kernel $\mathcal{K}(x, y) = e^{-|x-y|^2/(2\sigma^2)}$ for $\mathbb{R}^d$, and $\mathcal{K}(x, y) = (\sin a(x - y))/(\pi(x - y))$ for 1D space $\mathbb{R}$. By Bochner's theorem [25], if a function $K$ has a positive Fourier transform, then

$$\mathcal{K}(x, y) = K(x - y) \tag{2-13}$$

is a positive definite kernel. With this kernel, (1-1) becomes

$$\dot{X}_i = -\frac{1}{N}\sum_{j=1}^{N} \nabla K(X_i - X_j) - \frac{1}{N}\sum_{j=1}^{N} K(X_i - X_j)\nabla V(X_j), \tag{2-14}$$

as used in [21]. Both Gaussians and $1/|x|^\alpha$ with $\alpha \in (0, d)$ have positive Fourier transforms. The difference is that the Gaussian has a short range of interaction while the latter has a long range of interaction. One can smoothen $1/|x|^\alpha$ out by mollifying with Gaussian kernels, resulting in positive definite smooth kernels but with long-range interaction. Choosing localized kernels like Gaussians may have some issues in very high-dimensional spaces [12; 10]. Due to its simplicity, when the dimension is not very high, we choose Gaussian kernels in Section 4.

As a further comment, one may consider other metrics to gauge the closeness of probability measures, such as Wasserstein distances. Also, one can consider other norms for $\phi$ and get gradient flows in different spaces. These variants have been explored by some authors already [18; 8]. In general, computing the Frechét derivatives in closed form for these variants seems not that easy.

**Remark.** If we optimize (2-7) for $\phi$ in $L^2(\mathbb{R}^d;\mathbb{R}^d)$ spaces, the flow is then given by

$$\frac{d}{dt}\mathcal{T} = (S_\pi^* \rho)\circ\mathcal{T}. \tag{2-15}$$

The corresponding PDE is $\partial_t \rho = \nabla\cdot(\rho(\rho\nabla V + \nabla\rho)) = \nabla\cdot(\rho^2\nabla\log(\rho/\pi))$. This is in fact the case when we choose $\mathcal{K}(x, y) = \delta(x-y)$. This PDE, however, will not make sense for empirical measures since $\rho\nabla\rho$ is hard to justify (clearly, the equivalent ODE system has the same trouble). By using RKHS, the derivative on $\nabla\rho$ can be moved onto the kernel and then the ODE system makes sense.

## 3. The new sampling algorithm: RBM-SVGD

In this section, we introduce the "random batch" or "mini-batch" idea, which has already appeared in many places, and recall the random batch method for simulating interacting particle systems in [16]. By applying the random batch method to (1-1), we obtain a new algorithm, called RBM-SVGD. The proof that RBM-SVGD is close to SVGD on finite time interval is given in Section 3.2.

**3.1.** ***The algorithms.*** Before we present the random batch method and RBM-SVGD, let us briefly explain what the "random mini-batch" idea is. Let us consider a typical optimization problem in machine learning:

$$\min L(\omega) := \min \frac{1}{M}\sum_{j=1}^{M} \ell(g(z_j;\omega), y_j), \tag{3-1}$$

where $(z_j, y_j)_{j=1}^M$ are some given data set, $g(\,\cdot\,;\omega)$ is a model that takes $z_j$ as an input and gives some prediction to $y_j$, and $\ell(\,\cdot\,,\cdot\,)$ is some function to gauge the discrepancy between $g(z_j;\omega)$ and $y_j$. Hence, the problem is to find $\omega$ such that the discrepancy is small enough. Often $\ell(\,\cdot\,,\cdot\,)$ is a neural network so that computing the gradient is not easy. Hence, if one aims to find the minimizer using gradient descent, the computation cost is high. The idea of "mini-batch" or "random batch" is to choose a small random subset $\xi$ of $\{1, 2, \dots, N\}$, and consider the unbiased random estimate

$$L_\xi(\omega) := \frac{1}{B}\sum_{j\in\xi} \ell(g(z_j;\omega), y_j),$$

with $B = |\xi|$, the size of $\xi$. Using this unbiased estimation $L_\xi$ to replace the original true gradient $\nabla L_\xi \approx \nabla L$, one can form the so-called stochastic gradient descent (SGD) [4; 6]. Using a similar idea for Langevin dynamics, Welling and Teh obtained a Markov chain Monte Carlo method, called the stochastic gradient Langevin dynamics (SGLD), useful for Bayesian inference [29].

```
for $m$ in $1 : N_T$ do
  Divide $\{1, 2, \dots, pn\}$ into $n$ batches randomly.
  for each batch $\mathscr{C}_q$ do
    Update $X_i$ $(i \in \mathscr{C}_q)$ by solving the equation for $t \in [t_{m-1}, t_m)$:
      $$\dot{X}_i = \frac{1}{N} F(X_i, X_i) + \left(1 - \frac{1}{N}\right) \frac{1}{p-1} \sum_{j \in \mathscr{C}_q,\, j \neq i} F(X_i, X_j). \qquad (3\text{-}4)$$
  end for
end for
```

**Algorithm 1.** Random batch method without replacement.

Consider in general the interacting particle system of the form

$$\dot{X}_i = \frac{1}{N} \sum_{j=1}^{N} F(X_i, X_j) = \frac{1}{N} F(X_i, X_i) + \frac{1}{N} \sum_{j : j \neq i} F(X_i, X_j). \tag{3-2}$$

Here, $F(x, y)$ does not have to be symmetric, and also $F(x, x)$ is not necessarily zero. It is desirable to develop some cheap random approximation to the interacting forces so that the one-step $O(N^2)$ complexity can be reduced. One idea is to use the “random batch” idea, but how to develop the concrete “random batch” algorithm depends on the concrete applications. Regarding the interacting particle systems, Jin et al. proposed some random grouping approach to achieve this goal in [16].

Here, we adopt the random batch method in [16] to (3-2) and then obtain a stochastic version method for the SVGD ODE system (1-1). For this reason, we explain the random batch method a little bit. Choose a time step $\eta$. We define time grid points

$$t_m = m\eta. \tag{3-3}$$

At $t_m$, one divides the particles into groups randomly, and each group is called a “batch”, and then turns on interactions inside batches only. As indicated in [16], the random division of the particles into $n$ batches takes $O(N)$ operations (one can for example use random permutation). Depending on whether one does batches without or with replacement, one can have different versions (see Algorithms 1 and 2). For the ODEs in the algorithms, one can apply any suitable ODE solver. For example, one can use the forward Euler discretization if $F$ is smooth like Gaussian kernels. If $K$ is singular, one may take $p = 2$ and apply the splitting strategy in [16].

For the SVGD ODE system (1-1), the kernel $F$ takes the form

$$F(x, y) = \nabla_y \mathscr{K}(x, y) - \mathscr{K}(x, y) \nabla V(y). \tag{3-6}$$

Applying the random batch method to this special kernel and using any suitable ODE solvers, we get a class of sampling algorithms, which we will call RBM-SVGD. In

```
for $m$ in $1 : N_T * (N/p)$ do
    Pick a set $\mathscr{C}$ of size $p$ randomly.
    Update $X^i$ ($i \in \mathscr{C}$) by solving the following with pseudotime $s \in [s_{m-1}, s_m)$:
        $$\dot{X}_i = \frac{1}{N} F(X_i, X_i) + \left(1 - \frac{1}{N}\right) \frac{1}{p-1} \sum_{j \in \mathscr{C},\, j \neq i} F(X_i, X_j). \qquad (3\text{-}5)$$
end for
```

**Algorithm 2.** Random batch method with replacement.

```
for $k$ in $0 : N_T - 1$ do
    Divide $\{1, 2, \dots, pn\}$ into $n$ batches randomly.
    for each batch $\mathscr{C}_q$ do
        For all $i \in \mathscr{C}_q$,
            $$X_i^{(k+1)} \leftarrow X_i^{(k)} + \frac{1}{N}(\nabla_y \mathcal{K}(X_i^{(k)}, X_i^{(k)}) - \mathcal{K}(X_i^{(k)}, X_i^{(k)}) \nabla V(X_i^{(k)}))\eta_k + \Phi_{k,i}\eta_k,$$
        where
            $$\Phi_{k,i} = \frac{N-1}{N(p-1)} \sum_{j \in \mathscr{C}_q,\, j \neq i} (\nabla_y \mathcal{K}(X_i^{(k)}, X_j^{(k)}) - \mathcal{K}(X_i^{(k)}, X_j^{(k)}) \nabla V(X_j^{(k)})). \qquad (3\text{-}7)$$
    end for
end for
```

**Algorithm 3.** RBM-SVGD.

this work, we will focus on the ones without replacement. The one with forward Euler discretization (with possible variant step size) is shown in Algorithm 3. Clearly, the complexity is $O(pN)$ for each iteration.

Here, $N_T$ is the number of iterations and $\{\eta_k\}$ is the sequence of time steps, which play the same role as learning rate in SGD [4; 6]. For some applications, one may simply set $\eta_k = \eta \ll 1$ to be a constant and get relatively good results. However, in many high-dimensional problems, choosing $\eta_k$ to be constant may yield divergent sequences [23]. One may decrease $\eta_k$ to obtain convergent data sequences. For example, one may simply choose $\eta_k = 1/k$ as in SGD. Another frequently used strategy is the AdaGrad approach [11; 28].

**3.2.** ***Theoretic results.*** We now give convergence analysis regarding the time-continuous version of RBM-SVGD on torus $\mathbb{T}^d$ (i.e., choosing the particular force (3-6) for Algorithm 1 and $X_i \in \mathbb{T}^d$). The analysis in this section justifies the expectation that RBM-SVGD should give similar performance as the original SVGD, as confirmed by the numerical experiments in Section 4.

By “torus”, we mean the domain is equipped with periodic boundary conditions. The derivation of SVGD clearly stays unchanged for the torus. The reason we consider a torus is that (1-1) is challenging to analyze in $\mathbb{R}^d$ because of the nonlocal effect of the external force. On the torus, all functions are smooth and bounded. Moreover, using bounded domains with periodic boundary condition can always approximate the problem in $\mathbb{R}^d$ in practice.

Consider the random force for $z = (x_1, \dots, x_N) \in \mathbb{T}^{Nd}$ defined by

$$f_i(z) := \left(1 - \frac{1}{N}\right)\frac{1}{p-1}\sum_{j:j\in\mathscr{C}} F(x_i, x_j), \tag{3-8}$$

where $\mathscr{C}$ is the random batch that contains $i$ in the random batch method. Correspondingly, the exact force is given by

$$F_i(z) = \frac{1}{N}\sum_{j:j\neq i} F(x_i, x_j).$$

Define the “noise” by

$$\chi_i(z) := \frac{1}{N}\sum_{j:j\neq i} F(x_i, x_j) - f_i(z). \tag{3-9}$$

We have the following consistency result regarding the random batch.

**Lemma 1.** *For given $z = (x_1, \dots, x_N) \in \mathbb{T}^{Nd}$ (or $\mathbb{R}^{Nd}$), it holds that*

$$\mathbb{E}\chi_i(z) = 0. \tag{3-10}$$

*Moreover, the second moment is given by*

$$\mathbb{E}|\chi_i(z)|^2 = \left(1 - \frac{1}{N}\right)^2\left(\frac{1}{p-1} - \frac{1}{N-1}\right)\Lambda_i(z), \tag{3-11}$$

*where*

$$\Lambda_i(z) = \frac{1}{N-2}\sum_{j:j\neq i}\left|F(x_i, x_j) - \frac{1}{N-1}\sum_{k:k\neq i} F(x_i, x_k)\right|^2. \tag{3-12}$$

The proof is similar to that in [16], but we also attach it in Appendix A for convenience.

We recall that the Wasserstein-2 distance is given by [26]

$$W_2(\mu, \nu) = \left(\inf_{\gamma\in\Pi(\mu,\nu)}\int_{\mathbb{T}^d\times\mathbb{T}^d}|x-y|^2\,d\gamma\right)^{1/2}, \tag{3-13}$$

where $\Pi(\mu, \nu)$ is called the transport plan, consisting of all the joint distributions whose marginal distributions are $\mu$ and $\nu$, respectively: i.e., for any Borel set $E \subset \mathbb{T}^d$, $\mu(E) = \iint_{x\in E, y\in\mathbb{T}^d}\gamma\,(dx, dy)$ and $\nu(E) = \int_{x\in\mathbb{T}^d, y\in E}\gamma\,(dx, dy)$.

We now state the convergence result for the time-continuous version of RBM-SVGD, where we recall that $F(x, y)$ given by (3-6). We use $\widetilde{X}$ to denote the process generated by the random algorithm while $X$ is the process by (1-1). The particles are exchangeable if the initial values are sampled i.i.d. from the same distribution. Hence, the distributions of $X_i$ are the same, and we call this one-particle distribution the one marginal distribution, which is a probability measure in $\mathbb{T}^d$ or $\mathbb{R}^d$. We denote it by $\mu_N^{(1)}$ for convenience. Similarly, we introduce the one marginal distribution for the particles generated by the random algorithm, denoted by $\tilde{\mu}_N^{(1)}$.

**Theorem 2.** *Assume $V$ and $K$ are smooth on torus $\mathbb{T}^d$. The initial data $X_i^0$ are drawn independently from the same initial distribution. Given $T > 0$, there exists $C(T) > 0$, such that*

$$\sup_{t\le T} \mathbb{E}|X_i(t) - \widetilde{X}_i(t)|^2 \le C(T)\frac{\eta}{p-1}.$$

*Consequently, the one marginals $\mu_N^{(1)}$ and $\tilde{\mu}_N^{(1)}$ are close under Wasserstein-2 distance:*

$$\sup_{t\le T} W_2(\mu_N^{(1)}(t), \tilde{\mu}_N^{(1)}(t)) \le C(T)\sqrt{\frac{\eta}{p-1}}.$$

*Proof.* In the proof below, the constant $C$ will represent a general constant independent of $N$ and $p$, but its concrete meaning can change for every occurrence.

Consider the corresponding two processes and $t \in [t_{m-1}, t_m]$:

$$\begin{aligned}\frac{d}{dt}\widetilde{X}_i = \frac{1}{N}(\nabla_y \mathcal{K}(\widetilde{X}_i, \widetilde{X}_i) - \mathcal{K}(\widetilde{X}_i, \widetilde{X}_i)\nabla V(\widetilde{X}_i))& \\ + \frac{1-1/N}{p-1}\sum_{j:j\in\mathcal{C}}(\nabla_y \mathcal{K}(\widetilde{X}_i, \widetilde{X}_j) - \mathcal{K}(\widetilde{X}_i, \widetilde{X}_j)\nabla V(\widetilde{X}_j))& \quad (3\text{-}14)\end{aligned}$$

and

$$\begin{aligned}\frac{d}{dt}X_i = \frac{1}{N}(\nabla_y \mathcal{K}(X_i, X_i) - \mathcal{K}(X_i, X_i)\nabla V(X_i))& \\ + \frac{1}{N}\sum_{j:j\neq i}(\nabla_y \mathcal{K}(X_i, X_j) - \mathcal{K}(X_i, X_j)\nabla V(X_j)).& \quad (3\text{-}15)\end{aligned}$$

Taking the difference and dotting with $\widetilde{X}_i - X_i$, one has

$$(\widetilde{X}_i - X_i)\cdot\frac{d}{dt}(\widetilde{X}_i(t) - X_i(t)) \le \frac{C}{N}|\widetilde{X}_i(t) - X_i(t)|^2 + (\widetilde{X}_i(t) - X_i(t))\cdot(I_1 + I_2)$$

where

$$I_1 = \frac{1-1/N}{p-1}\biggl(\sum_{j:j\in\mathscr{C}}(\nabla_y\mathscr{K}(\widetilde{X}_i,\widetilde{X}_j)-\mathscr{K}(\widetilde{X}_i,\widetilde{X}_j)\nabla V(\widetilde{X}_j)) \\ -\sum_{j:j\in\mathscr{C}}(\nabla_y\mathscr{K}(X_i,X_j)-\mathscr{K}(X_i,X_j)\nabla V(X_j))\biggr),$$

$$I_2 = \frac{1-1/N}{p-1}\sum_{j:j\in\mathscr{C}}(\nabla_y\mathscr{K}(X_i,X_j)-\mathscr{K}(X_i,X_j)\nabla V(X_j)) \\ -\frac{1}{N}\sum_{j:j\neq i}(\nabla_y\mathscr{K}(X_i,X_j)-\mathscr{K}(X_i,X_j)\nabla V(X_j)).$$

Hence, introducing

$$u(t) = \mathbb{E}|X_i(t)-\widetilde{X}_i(t)|^2 = \mathbb{E}|X_1(t)-\widetilde{X}_1(t)|^2,$$

we have

$$\frac{d}{dt}u \le \frac{C}{N}u(t)+\mathbb{E}(X_i-\widetilde{X}_i)\cdot I_1+\mathbb{E}(X_i-\widetilde{X}_i)\cdot I_2.$$

Due to the smoothness of $K$ and $V$ on the torus, we easily find

$$|I_1| \le C\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}(|X_i-\widetilde{X}_i|+|X_j-\widetilde{X}_j|) \\ = C|X_i-\widetilde{X}_i|+C\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}|X_j-\widetilde{X}_j|,$$

where $C$ is independent of $N$. Note that $\mathscr{C}$ is not independent of $X_j(t)$ for $t > t_{m-1}$, so to continue we must consider conditional expectation. Let $\mathscr{F}_{m-1}$ be the $\sigma$-algebra generated by $X_i(\tau)$, $\widetilde{X}_i(\tau)$ for $\tau \le t_{m-1}$ (including the initial data drawn independently) and the random division of the batches at $t_{m-1}$. Then (3-14) directly implies almost surely that

$$\mathbb{E}(|X_j(t)-X_j(t_{m-1})|\,|\,\mathscr{F}_{m-1})\le C\eta, \quad \mathbb{E}(|\widetilde{X}_j(t)-\widetilde{X}_j(t_{m-1})|\,|\,\mathscr{F}_{m-1})\le C\eta. \tag{3-16}$$

Thus, defining the error process

$$Y_i(t) = \widetilde{X}_i(t)-X_i(t), \tag{3-17}$$

we have $\mathbb{E}(|Y_i(t)-Y_i(t_{m-1})|)\le C\eta$, yielding

$$|\sqrt{u}(t)-\sqrt{u}(t_{m-1})| \le C\eta. \tag{3-18}$$

Note that

$$\mathbb{E}\biggl(|X_i-\widetilde{X}_i|\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}|X_j-\widetilde{X}_j|\biggr) \le \sqrt{u}\biggl(\frac{1}{p-1}\mathbb{E}\sum_{j\in\mathscr{C},\, j\neq i}|X_j-\widetilde{X}_j|^2\biggr)^{1/2}.$$

The inside of the parentheses can be estimated as

$$\frac{1}{p-1}\mathbb{E}\sum_{j\in\mathscr{C},\, j\neq i}|X_j-\widetilde{X}_j|^2=\frac{1}{p-1}\mathbb{E}\sum_{j\in\mathscr{C},\, j\neq i}|X_j(t_{m-1})-\widetilde{X}_j(t_{m-1})|^2 \\ +\frac{1}{p-1}\mathbb{E}(\mathbb{E}((|X_j-\widetilde{X}_j|^2-|X_j(t_{m-1})-\widetilde{X}_j(t_{m-1})|^2)\mid \mathscr{F}_{m-1})).$$

The first term on the right-hand side then becomes $u(t_{m-1})$ by Lemma 1. By (3-16), it is clear that

$$\mathbb{E}((|X_j-\widetilde{X}_j|^2-|X_j(t_{m-1})-\widetilde{X}_j(t_{m-1})|^2)\mid \mathscr{F}_{m-1}) \\ \le 2|X_j(t_{m-1})-\widetilde{X}_j(t_{m-1})|C\eta+C\eta^2.$$

Hence,

$$\mathbb{E}(X_i-\widetilde{X}_i)\cdot I_1\le Cu(t)+Cu(t_{m-1})+C\sqrt{u(t_{m-1})}\eta+C\eta^2,$$

where $C$ is independent of $N$. Since $u(t_{m-1})\le Cu(t)+C\eta^2$ by (3-18), then

$$\mathbb{E}(X_i-\widetilde{X}_i)\cdot I_1\le Cu(t)+C\eta^2.$$

Letting $Z=(X_1,\ldots,X_N)$, one sees easily that $I_2=\chi_i(Z(t))$. Then, we find

$$Y_i(t)\cdot I_2(t)=(Y_i(t)-Y_i(t_{m-1}))\cdot\chi_i(Z(t))+Y_i(t_{m-1})\cdot\chi_i(Z(t))=: J_1+J_2.$$

In $J_2$, $Y_i(t_{m-1})$ is independent of the random batch division at $t_{m-1}$. Then, Lemma 1 tells us that

$$\mathbb{E}J_2=0.$$

Using (3-14), we have

$$Y_i(t)-Y_i(t_{m-1})=-\int_{t_{m-1}}^{t}\chi_i(Z(s))\,ds+\int_{t_{m-1}}^{t}f_i(\widetilde{Z}(s))-f_i(Z(s))\,ds. \tag{3-19}$$

Since $\chi_i$ is bounded,

$$\left|\mathbb{E}\int_{t_{m-1}}^{t}\chi_i(Z(s))\cdot\chi_i(Z(t))\,ds\right|\le\|\Lambda_i\|_\infty\eta\le 2\|F\|_\infty\frac{\eta}{p-1}, \tag{3-20}$$

where $C$ is related to the infinity norm of the variance of $\chi_i(t)$. This is the main term in the local truncation error. Just as we did for $I_1$,

$$|f_i(\widetilde{Z}(s))-f_i(Z(s))|\le C\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}(|X_i-\widetilde{X}_i|+|X_j-\widetilde{X}_j|) \\ =C|X_i-\widetilde{X}_i|+\frac{C}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}|X_j-\widetilde{X}_j|.$$

Since

$$\mathbb{E}\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}|X_j-\widetilde{X}_j|\le \mathbb{E}\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}|X_j(t_{m-1})-\widetilde{X}_j(t_{m-1})| \\ +\mathbb{E}\biggl(\frac{1}{p-1}\sum_{j\in\mathscr{C},\, j\neq i}\mathbb{E}(|X_j(s)-\widetilde{X}_j(s)-(X_j(t_{m-1})-\widetilde{X}_j(t_{m-1}))|\mid\mathscr{F}_{m-1})\biggr),$$

this is controlled by $C\sqrt{u(t_{m-1})}+C\eta$. Hence,

$$\mathbb{E}J_1\le 2\|F\|_\infty\frac{\eta}{p-1}+C\sqrt{u(t_{m-1})}\eta+C\eta^2.$$

Using the fact that $u(t_{m-1})\le u(t)+C\eta$, one eventually has that

$$\frac{d}{dt}u\le Cu+2\|F\|_\infty\frac{\eta}{p-1}+C\eta^2.$$

Applying Grönwall's inequality, we find

$$\sup_{t\le T}u(t)\le C(T)\frac{\eta}{p-1}.$$

The last claim for $W_2$ distance follows from the definition of $W_2$. □

Note that the one marginal $\mu_N^{(1)}(t)$ is the distribution of $X_i(t)$ for any $i$, which is deterministic. This should be distinguished from the empirical measure $\mu_N=(1/N)\sum_i\delta(x-X_i(t))$ which is random. As can be seen from the proof, the main contribution in the local truncation error comes from the variance of the noise $\chi_i$.

As can be seen, the error bound is given by the square root of variance of the random force times $\sqrt{\eta}=\sqrt{T/N_T}$ with $N_T$ being the number of steps. Hence, the result is a type of law of large number convergence result (see [16] for more details). The bigness of the variance on one hand depends on the batch size as $1/(p-1)-1/(N-1)$, while on the other hand depends on the bigness of the interaction. As long as the variance is bounded, the convergence of random batch method is ensured.

One crucial part is that the bigness of the variance depends on the bigness of the interaction, instead of the range of the interaction. This means that the random batch version of the algorithm is particularly useful when the interaction has long range or when the particles are not sparse. In fact, if the interaction has short range and the particles are sparse, one can use some data structure like cell-list [13, Appendix F] to reduce the computation of the interactions from $O(N^2)$ to $O(N)$. However, when the interaction has long range or is not sparse (like the case in the example in Section 4.2), those data structures cannot be used any more, and RBM-SVGD becomes useful: it can still reduce the cost from $O(N^2)$ to $O(N)$.

As another observation, according to (3-11) and (3-12), the bigness of the variance depends on the bigness of the interaction kernel. As long as the variance stays controlled, the convergence of RBM-SVGD to SVGD is guaranteed. In this sense, the range of the interaction kernel is not sensitive to RBM-SVGD, so it can intrinsically be used for kernels that have long range. The choice of kernels clearly affects the performance of SVGD, but it seems not so significant for RBM-SVGD to approximate SVGD. In other words, we expect RBM-SVGD to work well when the kernel is chosen such that SVGD behaves well. In fact, our experience in Section 4 confirmed this.

**Remark.** We believe the error bound in Theorem 2 can be made independent of $T$ due to the intrinsic structure of SVGD discussed above in Section 2. Then RBM-SVGD can be used as the efficient sampling algorithm from the desired distribution $\pi$. Such long time estimates are often established by some contracting properties of the ODE flows, so one may want to find the intrinsic converging structure of (1-1). However, rigorously establishing such results seems nontrivial due to the nonlocal effects of the external forces ($\nabla V$ terms).

## 4. Numerical experiments

We consider some test examples in [19] to validate RBM-SVGD algorithm and compare with the original SVGD algorithm. In particular, in a toy example for 1D Gaussian mixture, RBM-SVGD is proved to be effective in the sense that the particle system converges to the expected distribution with less running time than the original SVGD method. A more practical example, namely Bayesian logistic regression, is also considered to verify the effectiveness of RBM-SVGD on large data sets in high dimension. Competitive prediction accuracy is presented by RBM-SVGD, and less time is needed. Hence, RBM-SVGD seems to be a more efficient method.

All numerical results in this section are implemented with Matlab R2018a and performed on a machine with Intel Xeon CPU E5-1650v2 at 3.50 GHz with 64 GB memory.

**4.1.** ***1D Gaussian mixture.*** As a first example, we use the Gaussian mixture probability in [20] for RBM-SVGD. The initial distribution is $\mathcal{N}(-10, 1)$, Gaussian with mean $-10$ and variance 1. The target density is given by the Gaussian mixture

$$\pi(x) = \frac{1}{3}\cdot\frac{1}{\sqrt{2\pi}}e^{-(x+2)^2/2} + \frac{2}{3}\cdot\frac{1}{\sqrt{2\pi}}e^{-(x-2)^2/2}. \tag{4-1}$$

The kernel for the RKHS is the Gaussian kernel

$$K(x) = \frac{1}{\sqrt{2\pi h}}e^{-x^2/2h}, \tag{4-2}$$

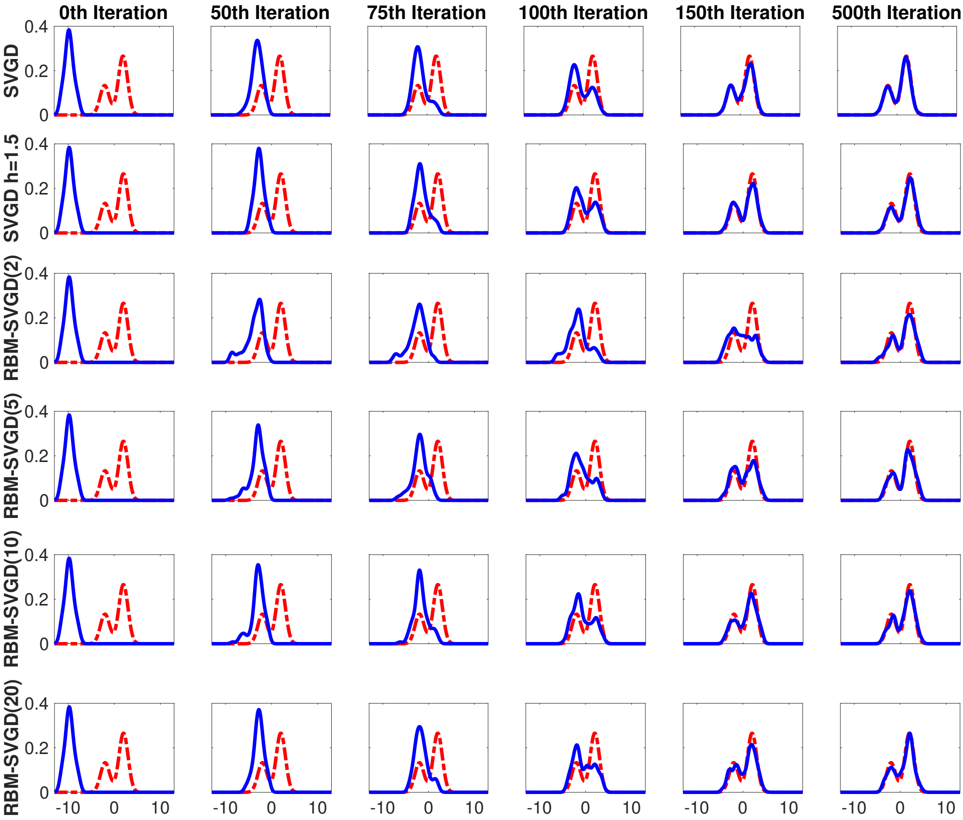


**Figure 1.** Comparison between SVGD and RBM-SVGD with different batch sizes using $N = 100$ particles. The first row reproduces results in [20]; the second row uses a fixed bandwidth $h = 2$ with other settings being the same as in the first row; the third to fifth rows apply RBM-SVGD with batch sizes 2, 5, and 20, respectively, and other settings are the same as in the second row. In all figures, red dashed curves indicate target density functions whereas blue curves are empirical density estimators (estimated using the kernel density estimator).

where $h$ is the bandwidth parameter. For a fair comparison with the numerical results in [20], we first reproduce their results using $N = 100$ particles and dynamic bandwidth parameter $h = \mathrm{med}^2/(2\log N)$, where med is the median of the pairwise distance between the current points. Since dynamic bandwidth is infeasible for RBM-SVGD, we produce the results with fixed bandwidth $h = 2$ for the comparison between SVGD and RBM-SVGD. The RBM-SVGD uses Algorithm 3 with initial step size 0.2 and the following step sizes generated from AdaGrad. Different batch sizes are tested to demonstrate the efficiency of RBM-SVGD. Numerical results are illustrated in Figure 1 with the same initial random positions of particles following an $\mathcal{N}(-10, 1)$ distribution.

As stated in [20], the difficulty lies in the strong disagreement between the initial density function and the target density $\pi(x)$. According to the first and second rows in Figure 1, SVGD with and without the fixed bandwidth parameter capture the target density efficiently and the corresponding convergence behaviors are similar to each other. Reading from the last column of Figure 1, we observe that RBM-SVGD inherits the advantage of SVGD in the sense that it can conquer the challenge and also show compelling result with SVGD. When the batch size is small, e.g., $p = 2$ or $p = 5$, the estimated densities differ from that of SVGD, and according to our experience, the estimated densities are not very stable across several executions while, in theory, RBM-SVGD runs $N/p$ times faster than SVGD. Hence, RBM-SVGD with $p = 5$ at the 500-th iteration costs the same as 50 iterations of SVGD. According to Figure 1, RBM-SVGD(2) at the 500-th iteration significantly outperforms the 50-th iteration of SVGD. As we increase the batch size, as in the last two rows of Figure 1, more stable and similar behavior to SVGD is observed.

Provided the good performance of RBM-SVGD, we also check the sampling power and its computational cost. We conduct the following simulations with $N = 256$ particles for 500 iterations with the Gaussian kernel (4-2). For RBM-SVGD, we use fixed bandwidth $h = 2$ whereas SVGD uses the aforementioned dynamic bandwidth strategy. When we apply SVGD or RBM-SVGD with different batch sizes, the same initial random positions of particles is used. For a given test function $h(x)$, we compute the estimated expectation $\bar{h} = (1/N)\sum_{i=1}^{N} h(X_i(T))$ and the sampling accuracy is measured via the minimum square error (MSE) over 100 random initializations following the same distribution as before:

$$\mathrm{MSE} = \frac{1}{100}\sum_{j=1}^{100}(\bar{h}_j - \mathbb{E}_{X\sim\pi} h(X))^2,$$

where $\mathbb{E}_{X\sim\pi} h(X)$ denotes the underlying truth. Three test functions are explored, $h_1(x) = x$, $h_2(x) = x^2$, and $h_3(x) = \cos 2x$, with their corresponding true expectations being $\frac{2}{3}$, 5, and $(\cos 4)/e^2$. The reported run time is also averaged over 100 random initializations.

Figure 2 shows the MSE against different batch sizes for $h_1(x)$, $h_2(x)$, and $h_3(x)$, respectively. The results of RBM-SVGD with different batch sizes are connected by lines, whereas the results of SVGD are the isolated points with batch size $p = 256$. In general, the estimations of $h_1(x)$ and $h_2(x)$ are better than that of $h_3(x)$, which agrees with the difficulty of the problems. Table 1 shows the averaged run time of RBM-SVGD and SVGD for different batch sizes under two different implementations in Matlab. RBM-SVGD is faster than SVGD for all choices of batch size. With respect to the two implementations in Matlab, for the first block row, within each batch, a matrix operation is adopted in computing the kernel matrix

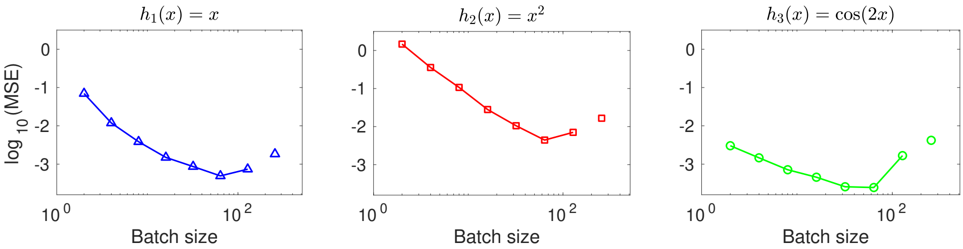


**Figure 2.** MSEs of (left) $h_1(x) = x$, (center) $h_2(x) = x^2$, and (right) $h_3(x) = \cos 2x$, against different batch sizes.

| Matlab | batch size | RBM-SVGD<br>2 | 4 | 8 | 16 | 32 | 64 | 128 | SVGD<br>256 |
|---|---|---|---|---|---|---|---|---|---|
| matrix op. | run time (s) | 0.055 | 0.095 | 0.178 | 0.341 | 0.270 | 0.238 | 0.314 | 0.733 |
| | speedup (×) | 13.3 | 7.7 | 4.1 | 2.1 | 2.7 | 3.1 | 2.3 | |
| row op. | run time (s) | 0.055 | 0.095 | 0.175 | 0.332 | 0.646 | 1.274 | 2.527 | 4.968 |
| | speedup (×) | 91.0 | 52.5 | 28.4 | 15.0 | 7.7 | 3.9 | 2.0 | |

**Table 1.** Averaged run time for different batch sizes.

whereas for the second block row, the kernel matrix is computed row by row. Matlab naturally is more favorable in the first implementation, which hence achieves fastest run time for all different batch sizes. For other programming languages, e.g., C++, Fortran, etc., the speedup of the second block row is excepted, which is close to ideal case as we predicted earlier.

**4.2.** ***Double banana.*** In this section, we will compare RBM-SVGD with MCMC, specifically Metropolis–Hastings (MH) [15]. The algorithmic detail of Metropolis–Hastings is available in Appendix B. The performance of RBM-SVGD and MH on a Bayesian inference task is compared to illustrate the advantage of RBM-SVGD. When the number of particles is not very large and desired accuracy is not high, RBM-SVGD can be more efficient.

We run MH and RBM-SVGD on a Bayesian inference task which is exactly the experiment in [10]. In this inference problem, our unknown parameter $x$ is in $\mathbb{R}^2$. The observational data $y$ is a real number which is determined by the forward map $\mathcal{F}(x)$ and the observational noise, i.e., $y = \mathcal{F}(x) + \xi$, where the forward map is a scalar logarithmic Rosenbrock function [24] $\mathcal{F}(x) = \log((1 - x_1)^2 + 100(x_2 - x_1^2)^2)$ for $x = (x_1, x_2)$ and the Gaussian noise $\xi$ satisfies $\xi \sim \mathcal{N}(0, \sigma^2)$ for $\sigma = 0.3$. The relationship between parameter $x$ and observation $y$ implies that the likelihood function is $p(y \mid x) = \mathcal{N}(F(x), \sigma^2)$. Finally, we set the prior distribution for $x$ to be Gaussian, i.e., $\pi_0(x) = \mathcal{N}(0, \tau^2 I_2)$, where $I_2$ is the identity matrix and $\tau$ will be specified later. Thus, the unnormalized posterior density is given by

$$\pi(x) = \pi_0(x) p(y \mid x) = \exp\left(-\frac{\|x\|^2}{2\tau^2} - \frac{(y - F(x))^2}{2\sigma^2}\right). \tag{4-3}$$

$N = 512$ particles are sampled in RBM-SVGD, and the maximum iteration number is 800. Different batch sizes are tested for performance, and the bandwidth parameter is fixed to be $h = 0.1$. To make MH comparable with RBM-SVGD regarding the number of sampling points, we viewed MH as a method with batch size 1, so the total number of iterations we performed for MH was $N \cdot 200$. We apply burn-in technique by only considering the second-half iterations. To reduce correlation, only 1 sample is drawn from every 100 iterations. Therefore, a total number of $N$ samples are selected from MH, which agrees with the number of particles we employ in RBM-SVGD. According to the performance test in Appendix B, we compare RBM-SVGD with MH by choosing $\tau = 5 \cdot 10^{-3}$, which is tested to be convergent and presents the best visual performance among different choices of $\tau$. For both RBM-SVGD and MH, the initial points are sampled from a Gaussian distribution $\mathcal{N}(0, 0.4^2)$. The target distribution is double banana with centers near $(0, 0.5)$ and $(0, -0.5)$. Hence, we adopt two test functions as $h_1(x_1, x_2) = \exp(-(x_1^2 + (x_2 - 0.5)^2)/(2 \cdot 0.5^2))$ and $h_2(x_1, x_2) = \exp(-(x_1^2 + (x_2 + 0.5)^2)/(2 \cdot 0.5^2))$.

In Figure 3, we plotted the position of each particle after RBM-SVGD iteration or MH together with the contour map of the target distribution. From the picture we can tell that both MH and RBM-SVGD can recover the shape of the target density and produce persuasive samplings. Although RBM-SVGD slightly harmed the aggregation of particles around the true distribution (which also paid off with a much shorter running time) compared to the original SVGD (RBM-SVGD with batch size $= 512$), it can still provide a convincing sampling by almost recovering the shape of the target density. In Table 2, we give further quantitative comparison. All numbers in the table are averaged over 100 different initializations. The run time for any RBM-SVGD with different batch sizes is faster than that of MH, and RBM-SVGD with batch size 2 is more than 20× faster while, regarding the MSE for both $h_1$ and $h_2$, RBM-SVGD is much better than MH for $h_1$ and better than MH for $h_2$. Hence, we conclude, for this example, SVGD outperforms MH both in run time and accuracy. RBM-SVGD further significantly reduces the run time of regular SVGD without loss of accuracy.

**4.3.** ***Bayesian logistic regression.*** In this experiment, we apply RBM-SVGD to conduct Bayesian logistic regression for binary classification for the Covertype data set with 581012 data points and 54 features [14]. Under the same setting as Gershman [14; 20], the regression weights $w$ of dimension 54 are assigned with a Gaussian prior $p_0(\omega \mid \alpha) = \mathcal{N}(w, \alpha^{-1})$, and the variance satisfies $p_0(\alpha) =$

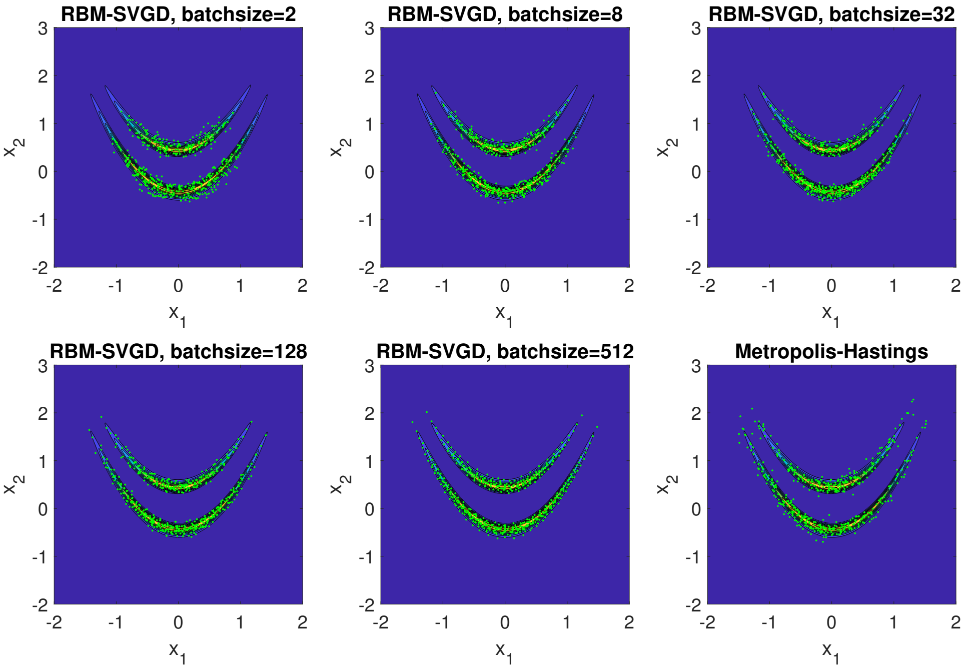


**Figure 3.** Comparison between RBM-SVGD and Metropolis–Hastings.

| | RBM-SVGD | | | | | MH |
|---|---|---|---|---|---|---|
| batch size | 2 | 8 | 32 | 128 | 512 | |
| run time (s) | 0.1321 | 0.3960 | 0.8268 | 0.9732 | 1.6086 | 2.9459 |
| $h_1$ MSE $\times 10^3$ | 0.0942 | 0.1862 | 0.2270 | 0.2910 | 0.3850 | 6.0689 |
| $h_2$ MSE $\times 10^3$ | 0.9559 | 2.2466 | 2.0151 | 1.0617 | 0.5634 | 3.5240 |

**Table 2.** Run time and MSE of $h_1$ and $h_2$ for RBM-SVGD and Metropolis–Hastings.

$\Gamma(\alpha, 1, 0.01)$, where $\Gamma$ represents the density of Gamma distribution. The inference is applied on posterior $p(x \mid D)$ with $x = [w, \log \alpha]$ of dimension 55. The kernel $K(\cdot)$ is taken again to be the same Gaussian kernel as (4-2).

Since the problem is in high dimension, we adopt $N = 512$ particles in this experiment, which also create more space for the selection of batch sizes. The training is done on 80% of the data set, and the other 20% is used as the test data set. For particle system (1-1), the computation of $-\nabla V = \nabla \log p(x)$ is expensive. Hence, we use the same strategy as mentioned in [20, §3.2], i.e., using data-mini-batch[1] of the data to form a stochastic approximation of $p(x)$ with the data-mini-batch size being 100. Since $\nabla \log p$ depends only on $x$ as in Algorithm 3,

[1] To avoid confusion with our batch of particles, we call it data-mini-batch instead.

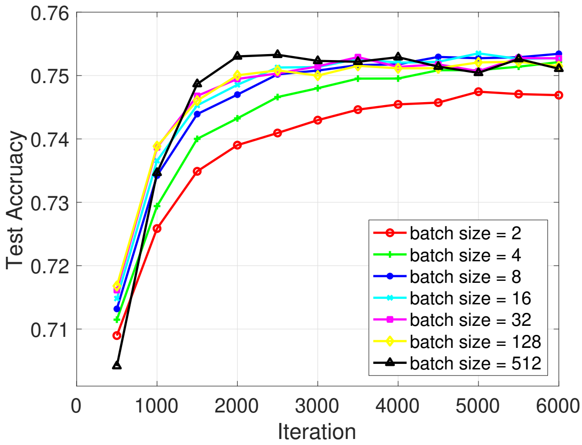


**Figure 4.** Test accuracy under different batch sizes of RBM-SVGD.

| | RBM-SVGD | | | | | | SVGD |
|---|---|---|---|---|---|---|---|
| batch size | 2 | 4 | 8 | 16 | 32 | 128 | 512 |
| run time (s) | 8.59 | 11.24 | 16.28 | 26.15 | 21.66 | 19.42 | 47.01 |
| speedup ($\times$) | 5.5 | 4.2 | 2.9 | 1.8 | 2.2 | 2.4 | |

**Table 3.** Average run time of 6000 iterations.

at each time step, we call this function only once and compute $\nabla \log p$ for all particles, which means the same data-mini-batches are used for $\nabla \log p$ of all particles. In this experiment, we use fixed bandwidth $h = 256$ for RBM-SVGD and dynamic bandwidth strategy for SVGD. The RBM-SVGD uses Algorithm 3 with initial step size being 0.05, and the following step sizes are generated from AdaGrad. Large $h$ is used here for the reason of high dimensionality. Different batch sizes are tested to demonstrate the efficiency of RBM-SVGD. Each configuration is executed on 50 random initializations. The averaged test accuracies for different batch sizes are illustrated in Figure 4.

As shown in Figure 4, RBM-SVGD is almost as efficient as SVGD even for small batch sizes. When $p = 2$, the test accuracy converges to a value slightly off that of SVGD. RBM-SVGD with $p = 4$ converges to the same accuracy as SVGD but at a slower convergent rate. For RBM-SVGD with batch size greater than 4, we observe similar convergence behavior as that of SVGD. The run time of RBM-SVGD, as shown in Table 3, is lower than that of SVGD, where the run time of 6000 iterations is reported. Comparing to the similar run time table for the 1D Gaussian mixture example (Table 1), the acceleration of RBM-SVGD is not as significant as before. This is due to the linear but expensive evaluation of $\nabla \log p$,

| | iteration | 1000 | 2000 | 3000 | 4000 | 5000 | 6000 |
|---|---|---|---|---|---|---|---|
| RBM-SVGD | mean | 0.7090 | 0.7349 | 0.7409 | 0.7446 | 0.7457 | 0.7471 |
| $p=2$ | std | 0.0045 | 0.0040 | 0.0040 | 0.0034 | 0.0034 | 0.0038 |
| RBM-SVGD | mean | 0.7342 | 0.7470 | 0.7508 | 0.7518 | 0.7527 | 0.7534 |
| $p=8$ | std | 0.0073 | 0.0056 | 0.0041 | 0.0045 | 0.0039 | 0.0033 |
| SVGD | mean | 0.7347 | 0.7530 | 0.7523 | 0.7529 | 0.7504 | 0.7511 |
| | std | 0.0068 | 0.0048 | 0.0071 | 0.0048 | 0.0061 | 0.0062 |

**Table 4.** Statistics of RBM-SVGD and SVGD.

where RBM-SVGD and SVGD spend the same amount of time in the evaluation each iteration. Although the evaluation of $\nabla \log p$ is expensive, it is linear in $N$. As $N$ increases, the advantage of RBM-SVGD would be more significant. In Table 4, we list the mean and standard deviation of RBM-SVGD with $p = 2$ and $p = 8$ and SVGD of different iterations. Based on the statistics, we conclude that RBM-SVGD and SVGD are of similar prediction power and RBM-SVGD is efficient also in high-dimensional particle systems as well.

## 5. Conclusion

We have applied the random batch method for interacting particle systems to SVGD, resulting in RBM-SVGD, which turns out to be a cheap sampling algorithm and inherits the efficiency of the original SVGD algorithm. Theory and numerical experiments have validated the algorithm, and hence, it can potentially have many applications, like Bayesian inference. Moreover, as a hybrid strategy, one may increase the batch size as time goes on to increase the accuracy, or apply some variance reduction approach.

## Appendix A: Proof of Lemma 1

*Proof of Lemma 1.* The proof is pretty much like the one in [16]. We use the random variable $I(i, j)$ to indicate whether $i$ and $j$ are in a common batch. In particular, $I(i, j) = 1$ if $i$ and $j$ are in a common batch while $I(i, j) = 0$ otherwise. Then it is not hard to compute [16]

$$\begin{aligned} \mathbb{E}1_{I(i,j)=1} &= \frac{p-1}{N-1}, \\ \mathbb{P}(I(i, j)I(j, k) = 1) &= \frac{(p-1)(p-2)}{(N-1)(N-2)}. \end{aligned} \tag{A-1}$$

We note

$$\chi_i(x) = \frac{1}{N} \sum_{j:j\neq i} \left(1 - \frac{N-1}{p-1} I(i, j)\right) F(x_i, x_j). \tag{A-2}$$

The first equation in (A-1) clearly implies that $\mathbb{E}\chi_i(x) = 0$. Using (A-1), we can compute directly that

$$\mathbb{E}|\chi_i(x)|^2 = \frac{1}{N^2}\Bigg(\sum_{j:j\neq i}\left(\frac{N-1}{p-1}-1\right)|F(x_i,x_j)|^2 + \sum_{j,k:j\neq i,\,k\neq i,\,j\neq k}\left(\frac{(N-1)(p-2)}{(N-2)(p-1)}-1\right)F(x_i,x_k)\cdot F(x_i,x_j)\Bigg).$$

Rearranging this, we get the claimed expression. □

## Appendix B: Metropolis–Hastings method and performance

Metropolis–Hastings (MH) is a method of MCMC which produces a reversible Markov chain where the unnormalized target distribution $\pi$ is invariant. This reversibility is realized by its "accept or reject" machinery. Roughly speaking, MH first generates a candidate according to a proposal distribution (which is always chosen as a normal distribution) and then determines whether to accept or reject the candidate according to the unnormalized target distribution $\pi$ [15]. In detail, the algorithm has four steps:

(1) *initialization*. Draw $X_0$ according to a given prior distribution $\pi_0$.

(2) *generate a candidate*. Given $X_n$, draw candidate $X'$ through a normal distribution with mean $X_n$ and covariance $C$, i.e.,

$$X' = X_n + \mathcal{N}(0, C).$$

(3) *calculate the acceptance rate*. Acceptance rate $\alpha$ is set as

$$\alpha = \min\left\{1, \frac{\pi(X')}{\pi(X)}\right\}.$$

(4) *accept or reject*. Then we accept $X'$ with probability $\alpha$ and reject it with probability $1-\alpha$, i.e., $X_{n+1} = X'$ with probability $\alpha$ and $X_{n+1} = X_n$ with probability $1-\alpha$.

The Markov chain constructed in this algorithm has transition kernel $h(x, y)$ which can be written as

$$h(x,y) = \exp\left(-\frac{(x-y)^T C^{-1}(x-y)}{2}\right)\cdot\min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}.$$

Direct calculation indicates that $h(x, y)\pi(x) = h(y, x)\pi(y)$. Thus, this Markov chain satisfies the detailed balance condition and hence has invariant measure $\pi$.

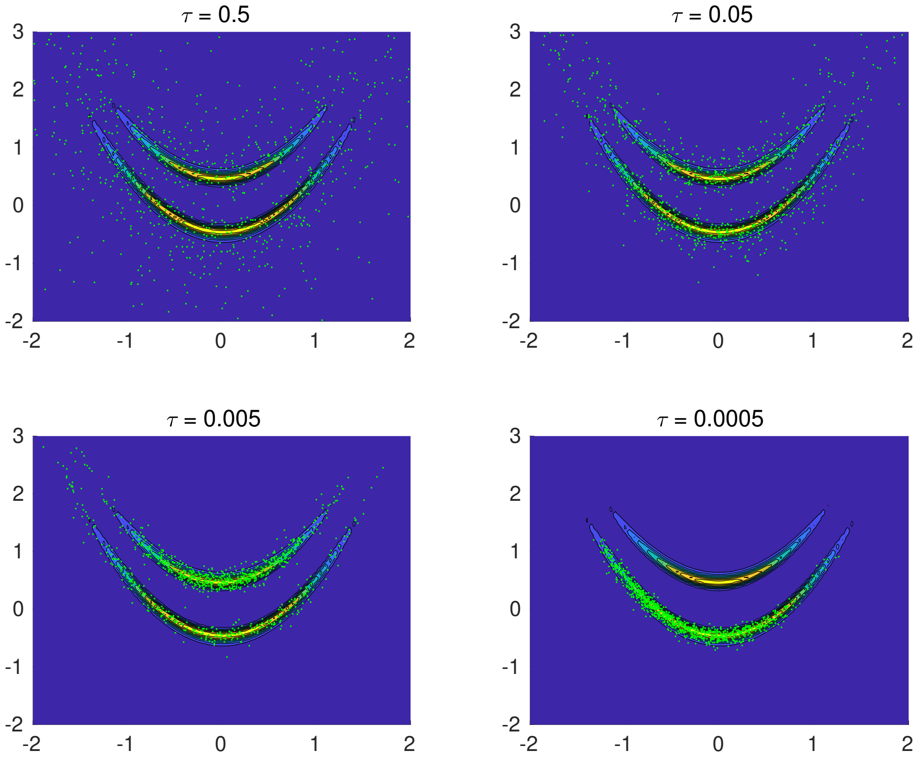


**Figure 5.** Samples from MH with different $\tau$.

Given the prior and target distribution as in (4-3), we first test the performance of MH among different values of parameter $\tau$, which represents the variance of the proposal distribution.

Figure 5 illustrates the samples together with the contour map of target distribution for $\tau = 5 \cdot 10^{-1}, 5 \cdot 10^{-2}, 5 \cdot 10^{-3}, 5 \cdot 10^{-4}$. Clearly, the best performance was attained when $\tau = 5 \cdot 10^{-3}$. For $\tau$ greater than $5 \cdot 10^{-3}$, samples are still wandering around the true distribution without accumulating due to a high variance, whereas for smaller $\tau$, samples are merely aggregating around the "lower" banana rather than the "upper" banana. This phenomenon can be explained by the small variance of proposal distribution, which confines the particles around upper banana.

Moreover, a convergence diagnosis for MCMC was also conducted by computing the auto-correlation [7]. A lower auto-correlation always implies better convergence because a higher auto-correlation indicates that effective sampling size is smaller and more iteration is necessary [27]. Figure 6 plots the auto-correlation of the first coordinate of samples at different time lag $\kappa$. For $\tau = 5 \cdot 10^{-1}, 5 \cdot 10^{-2}$, auto-correlation at $\kappa \leq 5 \cdot 10^3$ (2.5% of the number of samples) is plotted, while for $\tau = 5 \cdot 10^{-3}, 5 \cdot 10^{-4}$, auto-correlation at $\kappa \leq 5 \cdot 10^4$ (25% of the number of

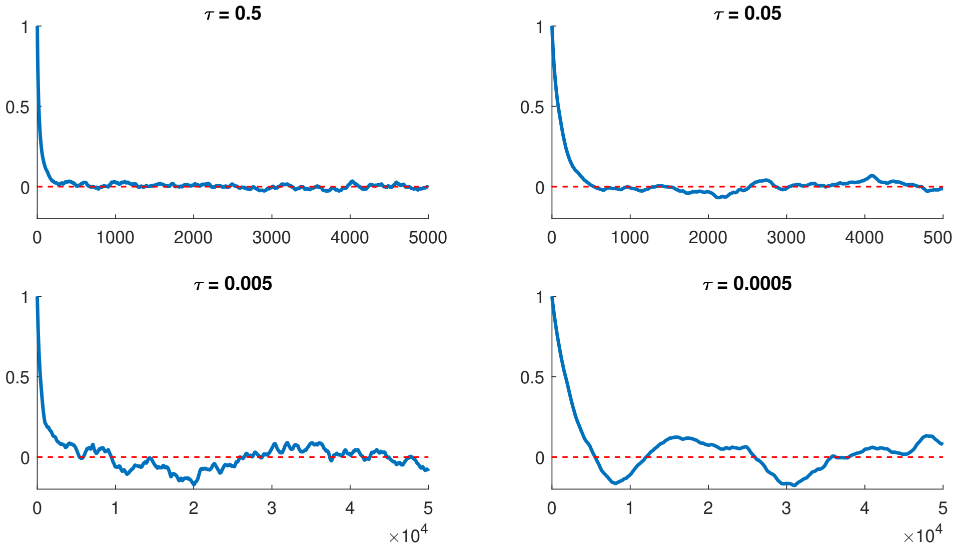


**Figure 6.** Auto-correlation curve of samples from MH with different $\tau$.

samples) is plotted. This figure shows that auto-correlation decays rapidly for $\tau = 5 \cdot 10^{-1}$, $5 \cdot 10^{-2}$ and oscillates around 0 with small magnitude. For $\tau = 5 \cdot 10^{-3}$, although it decays quickly, its oscillation has a greater magnitude. For $\tau = 5 \cdot 10^{-4}$, it does not converge to 0 at all. In conclusion, the convergence of MH with $\tau = 5 \cdot 10^{-1}$, $5 \cdot 10^{-2}$, $5 \cdot 10^{-4}$ is acceptable. Hence, in this paper, we use MH with $\tau = 5 \cdot 10^{-3}$ as a reference.

## Acknowledgements

This work is supported by KI-Net, NSF RNMS11-07444. The work of L. Li was partially sponsored by Shanghai Sailing Program 19YF1421300, the work of Y. Li was partially supported by OAC-1450280, the work of J.-G. Liu was partially supported by NSF DMS-1812573, and the work of J. Lu was supported in part by NSF DMS-1454939.

## References

[1] N. Aronszajn, *Theory of reproducing kernels*, Trans. Amer. Math. Soc. **68** (1950), 337–404. MR Zbl

[2] A. Berlinet and C. Thomas-Agnan, *Reproducing kernel Hilbert spaces in probability and statistics*, Kluwer, Boston, 2004. MR Zbl

[3] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, *Variational inference: a review for statisticians*, J. Amer. Statist. Assoc. **112** (2017), no. 518, 859–877. MR

[4] L. Bottou, *On-line learning and stochastic approximations*, On-line learning in neural networks (D. Saad, ed.), Cambridge University, 1998, pp. 9–42. Zbl

[5] G. E. P. Box and G. C. Tiao, *Bayesian inference in statistical analysis*, Addison-Wesley, Reading, MA, 1973. MR Zbl

[6] S. Bubeck, *Convex optimization: algorithms and complexity*, Found. Trends Machine Learn. **8** (2015), no. 3–4, 231–357. Zbl

[7] C. Chatfield, *The analysis of time series: an introduction*, 6th ed., Chapman & Hall/CRC Texts in Statistical Science Series, Chapman & Hall/CRC, Boca Raton, FL, 2004. MR Zbl

[8] C. Chen, R. Zhang, W. Wang, B. Li, and L. Chen, *A unified particle-optimization framework for scalable Bayesian sampling*, Uncertainty in Artificial Intelligence: proceedings of the thirty-fourth conference (Monterey, CA, 2018) (A. Globerson and R. Silva, eds.), AUAI, Corvallis, OR, 2018, p. 263.

[9] B. Dai, N. He, H. Dai, and L. Song, *Provable Bayesian inference via particle mirror descent*, Artificial intelligence and statistics (Cádiz, Spain, 2016) (A. Gretton and C. C. Robert, eds.), Proceedings of Machine Learning Research, no. 51, 2016, pp. 985–994.

[10] G. Detommaso, T. Cui, Y. Marzouk, A. Spantini, and R. Scheichl, *A Stein variational Newton method*, Advances in Neural Information Processing Systems 31 (Montréal, 2018), NIPS Proceedings, 2018.

[11] J. Duchi, E. Hazan, and Y. Singer, *Adaptive subgradient methods for online learning and stochastic optimization*, J. Mach. Learn. Res. **12** (2011), 2121–2159. MR Zbl

[12] D. Francois, V. Wertz, and M. Verleysen, *About the locality of kernels in high-dimensional spaces*, Applied Stochastic Models and Data Analysis (Brest, France, 2005) (J. Janssen and P. Lenca, eds.), ENST Bretagne, 2005, pp. 238–245.

[13] D. Frenkel and B. Smit, *Understanding molecular simulation: from algorithms to applications*, 2nd ed., Academic, 2002. Zbl

[14] S. Gershman, M. Hoffman, and D. Blei, *Nonparametric variational inference*, Proceedings of the 29th International Conference on Machine Learning (Edinburgh, 2012), Omnipress, Madison, WI, 2012, pp. 235–242.

[15] W. K. Hastings, *Monte Carlo sampling methods using Markov chains and their applications*, Biometrika **57** (1970), no. 1, 97–109. MR Zbl

[16] S. Jin, L. Li, and J.-G. Liu, *Random batch methods (RBM) for interacting particle systems*, J. Comput. Phys. **400** (2020), art. id. 108877. MR

[17] K. Law, A. Stuart, and K. Zygalakis, *Data assimilation: a mathematical introduction*, Texts in Applied Mathematics, no. 62, Springer, 2015. MR Zbl

[18] C. Liu and J. Zhu, *Riemannian Stein variational gradient descent for Bayesian inference*, Thirty-Second AAAI Conference on Artificial Intelligence (New Orleans, 2018), Association for the Advancement of Artificial Intelligence, 2018, pp. 3627–3634.

[19] Q. Liu, *Stein variational gradient descent as gradient flow*, Advances in Neural Information Processing Systems 30 (Long Beach, CA, 2017), NIPS Proceedings, 2017.

[20] Q. Liu and D. Wang, *Stein variational gradient descent: a general purpose Bayesian inference algorithm*, Advances in Neural Information Processing Systems 29 (Barcelona, 2016), NIPS Proceedings, 2016.

[21] J. Lu, Y. Lu, and J. Nolen, *Scaling limit of the Stein variational gradient descent: the mean field regime*, SIAM J. Math. Anal. **51** (2019), no. 2, 648–671. MR Zbl

[22] D. Rezende and S. Mohamed, *Variational inference with normalizing flows*, International Conference on Machine Learning (Lille, France, 2015) (F. Bach and D. Blei, eds.), Proceedings of Machine Learning Research, no. 37, 2015, pp. 1530–1538.

[23] H. Robbins and S. Monro, *A stochastic approximation method*, Ann. Math. Statistics **22** (1951), 400–407. MR Zbl

[24] H. H. Rosenbrock, *An automatic method for finding the greatest or least value of a function*, Comput. J. **3** (1960), 175–184. MR

[25] W. Rudin, *Fourier analysis on groups*, Interscience Tracts in Pure and Applied Mathematics, no. 12, Interscience, New York, 1962. MR Zbl

[26] F. Santambrogio, *Optimal transport for applied mathematicians: calculus of variations, PDEs, and modeling*, Progress in Nonlinear Differential Equations and their Applications, no. 87, Springer, 2015. MR Zbl

[27] A. Sokal, *Monte Carlo methods in statistical mechanics: foundations and new algorithms*, Functional integration (Cargèse, 1996) (C. DeWitt-Morette, P. Cartier, and A. Folacci, eds.), NATO Adv. Sci. Inst. Ser. B Phys., no. 361, Plenum, New York, 1997, pp. 131–192. MR Zbl

[28] R. Ward, X. Wu, and L. Bottou, *AdaGrad stepsizes: sharp convergence over nonconvex landscapes*, International Conference on Machine Learning (Long Beach, CA, 2019) (K. Chaudhuri and R. Salakhutdinov, eds.), Proceedings of Machine Learning Research, no. 97, 2019, pp. 6677–6686.

[29] M. Welling and Y. W. Teh, *Bayesian learning via stochastic gradient Langevin dynamics*, Proceedings of the 28th International Conference on International Conference on Machine Learning (L. Getoor and T. Scheffer, eds.), Omnipress, Madison, WI, 2011, pp. 681–688.

Received April 10, 2019. Revised November 12, 2019.

LEI LI: leili2010@sjtu.edu.cn
*School of Mathematical Sciences, Institute of Natural Sciences, Key Lab of Scientific and Engineering Computing, Ministry of Education, Shanghai Jiao Tong University, Shanghai, China*

YINGZHOU LI: yingzhou.li@duke.edu
*Department of Mathematics, Duke University, Durham, NC, United States*

JIAN-GUO LIU: jliu@phy.duke.edu
*Department of Mathematics, Department of Physics, Duke University, Durham, NC, United States*

ZIBU LIU: zibu.liu@duke.edu
*Department of Mathematics, Duke University, Durham, NC, United States*

JIANFENG LU: jianfeng@math.duke.edu
*Department of Mathematics, Department of Physics, Department of Chemistry, Duke University, Durham, NC, United States*

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 1, 2020
dx.doi.org/10.2140/camcos.2020.15.65

# A THIRD-ORDER MULTIRATE RUNGE–KUTTA SCHEME FOR FINITE VOLUME SOLUTION OF 3D TIME-DEPENDENT MAXWELL'S EQUATIONS

MARINA KOTOVSHCHIKOVA, DMITRY K. FIRSOV AND SHIU HONG LUI

A third-order multirate time-stepping based on an SSP Runge–Kutta method is applied to solve the three-dimensional Maxwell's equations on unstructured tetrahedral meshes. This allows for an evolution of the solution on fine and coarse meshes with time steps satisfying a local stability condition to improve the computational efficiency of numerical simulations. Two multirate strategies with flexible time-step ratios are compared for accuracy and efficiency. Numerical experiments with a third-order finite volume discretization are presented to validate the theory. Our results of electromagnetic simulations demonstrate that 1D analysis is also valid for linear conservation laws in 3D. In one of the methods, significant speedup in 3D simulations is achieved without sacrificing third-order accuracy.

## 1. Introduction

Many real life simulations require complicated geometries and highly nonuniform meshes. When explicit methods are used, the maximum allowed time step is defined by the smallest elements in the mesh. When a fine mesh is required only in a small region of a computational domain, it is not a desirable expense. In addition, when a small time step is used on a coarse grid, it often generates dissipation in the solution. To overcome the need for a restrictive time step, local time-stepping (LTS) or multirate methods are very useful. In this case local stability conditions (CFL) are imposed on subdomains of the computational domain in place of a global more restrictive stability condition.

The earliest works on multirate methods include multirate Runge–Kutta schemes by Rice [32] and Andrus [2; 3], multirate linear multistep by Gear and Wells [16], and local time-stepping with forward Euler by Osher and Sanders [30]. Over the last three decades multirate versions of many traditional temporal schemes, such as explicit Runge–Kutta [10; 11; 20; 27; 38], Adams–Bashforth [33], as well as implicit-explicit (IMEX) methods [34] were designed.

*MSC2010:* 65L06, 65M08, 78M12.

*Keywords:* multirate Runge–Kutta schemes, Maxwell's equations, three-dimensional unstructured meshes, finite volume.

In the computational electromagnetics literature one can find LTS versions of leap-frog schemes [9; 29; 18; 4], multistage (Runge–Kutta, predictor-corrector) [4; 15] and multistep (Adams–Bashforth) explicit methods [17; 19], Cauchy–Kovalevskaja procedures [39], and locally implicit time integration [12]. In [8] a method based on Yee's scheme with special discrete transmission conditions for unknown values at the interface between LTS subdomains was developed, and applied to the 3D Maxwell's equations in [9]. A space-time mesh refinement method is implemented with a discontinuous Galerkin (DG) space discretization for first-order hyperbolic systems in [14]. The advantage of the space-time mesh refinement method is that it guarantees the stability of the scheme by enforcing conservation of discrete energy. But it requires solution of a linear system at the interface between two grids at each time step. This becomes more and more computationally expensive as we increase the number of multirate domains in 3D space. An LTS method based on the symplectic Störmer–Verlet scheme was proposed by Piperno in [31]. The scheme with two levels of refinement was proven to conserve discrete energy. In [29] Montseny et al. followed the same idea to develop a leap-frog-based LTS scheme. In both cases time increments proportional to 2 are used and the latest available solution is used for coupling at the interface between domains with different time steps. In [13] Diaz and Grote derived an arbitrary (even) high-order LTS method for the second-order wave equation. Their method is based on an extension of the second-order leap-frog scheme by a modified equation approach [36]. The method was proven to conserve discrete energy under some CFL condition. Its implementation for the 2D Maxwell's equations can be found in [18]. An LTS method based on Adams–Bashforth multistep schemes was developed by the same authors in [19], and another implementation can be found in [17]. In [39] an LTS technique based on the arbitrary high-order derivatives (ADER) DG method was proposed. Unlike methods based on multistage time integration, there is no consistency challenge between solutions at different time increments in the ADER approach. This allows for a more flexible distribution of local time steps with optimal performance. A causal-path LTS technique utilizing multistage time schemes has been proposed by Angulo et al. in [4]. It was applied to Maxwell's equations using fourth-order RK and second-order leap-frog as base time integration schemes. Their LTS approach requires a computation of the stage value of neighbors in order to advance the solution on a given subdomain. Therefore, the idea is similar to the one proposed by Tang and Warnecke in [38].

In this work we analyze and implement two LTS approaches based on third-order strong stability preserving (SSP) Runge–Kutta to improve efficiency of third-order-accurate 3D electromagnetic simulations. One is a third-order extension of the idea proposed by Tang and Warnecke [38]. It is based on a projection of the solution to provide consistent coupling at LTS interfaces. Another one uses interpolation of stage values for the same purpose [27]. Both schemes allow arbitrary time-step ratios and are relatively inexpensive to implement with any 3D finite volume scheme on tetrahedral meshes. The flexible time-step ratio gives more optimal simulation speedup on nonuniform meshes with large differences in cell size without loss of accuracy. A linear version of order conditions is used to analyze the accuracy of both schemes. Our analysis shows that a third-order extension of the scheme from [38] leads to only a first-order coupling, while the scheme proposed in [27] maintains third-order accuracy. Both schemes are implemented for the 3D time-domain Maxwell's equations with a third-order finite volume spatial approximation. Two strategies to define local time-step distribution are compared. One is a traditional power of 2 base partition, and another one is based on a more flexible time-step ratios and optimization algorithm. Numerical results in 1D and 3D with both schemes confirm our theoretical results. Moreover, both proposed time-step distribution strategies lead to the same accuracy in our simulations confirming flexibility of considered schemes. Significant speedup is observed in both schemes for problems with large linear cell-size ratio.

The paper is organized as follows. Section 2 describes Maxwell's equations in the time domain and their finite volume discretization. Section 3 discusses multirate Runge–Kutta schemes in 1D and their accuracy analysis for linear problems. In Section 4 a 3D implementation of algorithms using arbitrary time-step distribution is presented. Finally, Section 5 shows numerical validation of third-order LTS schemes on 3D electromagnetic problems.

## 2. Finite volume scheme for Maxwell's equations

Consider the propagation of electromagnetic waves in a three-dimensional heterogeneous linear isotropic medium with space-varying electric permittivity $\epsilon = \epsilon(\mathbf{x})$ and magnetic permeability $\mu = \mu(\mathbf{x})$. Given a bounded region $\Omega \subset \mathbb{R}^3$, the electric field $\mathbf{E}(\mathbf{x}, \mathrm{t})$ and the magnetic field $\mathbf{H}(\mathbf{x}, \mathrm{t})$ are governed by the system of Maxwell's equations

$$\begin{cases} \epsilon \frac{\partial \mathbf{E}}{\partial \mathrm{t}} - \nabla \times \mathbf{H} = \mathbf{J}_E & \text{in } [0, T] \times \Omega, \\ \mu \frac{\partial \mathbf{H}}{\partial \mathrm{t}} + \nabla \times \mathbf{E} = \mathbf{J}_H, & \text{in } [0, T] \times \Omega, \\ a\hat{\boldsymbol{n}} \times \mathbf{E} + b\hat{\boldsymbol{n}} \times (\hat{\boldsymbol{n}} \times \mathbf{H}) = 0 & \text{on } [0, T] \times \partial\Omega, \end{cases} \tag{1}$$

where $\mathbf{J}_E$ and $\mathbf{J}_H$ are the sources consisting of imposed currents and terms introduced by scattered field formulation, and $\hat{\boldsymbol{n}}$ is the outward unit normal of the boundary $\partial\Omega$. Parameters $a$ and $b$ define different boundary conditions:

- perfect electric conductor (PEC), $a = 1$ and $b = 0$,
- perfect magnetic conductor (PMC), $a = 0$ and $b = 1$, and
- Silver–Müller absorbing boundary condition, $a = 1$ and $b = \sqrt{\mu/\epsilon}$.

Consider the normalized quantities

$$\boldsymbol{x} = l^{-1}\mathbf{x}, \qquad t = c_0 l^{-1}\mathrm{t}, \tag{2}$$

where $l$ is a reference length and $c_0 = (\mu_0\epsilon_0)^{-1/2}$ is a dimensional speed of light in vacuum with $\epsilon_0 \approx 8.854 \cdot 10^{-12}\,\frac{\mathrm{A \cdot s}}{\mathrm{V \cdot m}}$ and $\mu_0 = 4\pi \cdot 10^{-7}\,\frac{\mathrm{V \cdot s}}{\mathrm{A \cdot m}}$. The fields $\mathbf{E}$ and $\mathbf{H}$ can be normalized to a typical electric field intensity $E$ by

$$\boldsymbol{E} = \frac{\mathbf{E}}{E}, \qquad \boldsymbol{H} = \frac{Z_0}{E}\mathbf{H}, \qquad \boldsymbol{J}_E = \frac{lZ_0}{E}\mathbf{J}_E, \qquad \boldsymbol{J}_H = \frac{l}{E}\mathbf{J}_H, \tag{3}$$

where $Z_0 = \sqrt{\mu_0/\epsilon_0}$ is the dimensional free-space intrinsic impedance. Then the system (1) can be written in nondimensional form as

$$\begin{cases} \epsilon_r \frac{\partial \boldsymbol{E}}{\partial t} - \nabla \times \boldsymbol{H} = \boldsymbol{J}_E & \text{in } [0, c_0 l^{-1} T] \times \Omega, \\ \mu_r \frac{\partial \boldsymbol{H}}{\partial t} + \nabla \times \boldsymbol{E} = \boldsymbol{J}_H & \text{in } [0, c_0 l^{-1} T] \times \Omega, \\ a_r \hat{\boldsymbol{n}} \times \boldsymbol{E} + b_r \hat{\boldsymbol{n}} \times (\hat{\boldsymbol{n}} \times \boldsymbol{H}) = 0 & \text{on } [0, c_0 l^{-1} T] \times \partial\Omega, \end{cases} \tag{4}$$

where $\epsilon_r = \epsilon/\epsilon_0$, $\mu = \mu/\mu_0$, $a_r = a$, and $b_r = b/Z_0$. For a finite volume discretization, the first two equations of (4) are written in conservative form as

$$\boldsymbol{\alpha}\frac{\partial \boldsymbol{U}}{\partial t} + \nabla \cdot \boldsymbol{F}(\boldsymbol{U}) = \boldsymbol{J},$$

where

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{H} \\ \boldsymbol{E} \end{bmatrix}, \qquad \boldsymbol{F}(\boldsymbol{U}) = [\boldsymbol{F}_1(\boldsymbol{U}), \boldsymbol{F}_2(\boldsymbol{U}), \boldsymbol{F}_3(\boldsymbol{U})]^T, \quad \boldsymbol{F}_i = \begin{bmatrix} -e_i \times \boldsymbol{H} \\ e_i \times \boldsymbol{E} \end{bmatrix},$$

and

$$\boldsymbol{\alpha} = \begin{bmatrix} \epsilon_r & 0 \\ 0 & \mu_r \end{bmatrix}, \qquad \boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_E \\ \boldsymbol{J}_H \end{bmatrix}.$$

Consider a partition of the bounded domain $\Omega \subset \mathbb{R}^3$ into a tetrahedral mesh $\overline{\Omega}_T = \bigcup_{i=1}^{N} \overline{T}_i$. It is assumed that material properties are constant in each cell $T_i$. Integrating (4) over each tetrahedron $T_i$ and defining the cell-averaged values of a given function $u$ as $\bar{u}_i = (1/|T_i|)\int_{T_i} u\,dV$, the following semidiscrete finite volume scheme for Maxwell's equations is derived:

$$\boldsymbol{\alpha}_i \frac{\partial \overline{\boldsymbol{U}}_i}{\partial t} + \frac{1}{|T_i|}\int_{\partial T_i} \hat{\boldsymbol{n}} \cdot \boldsymbol{F}\,dS = \boldsymbol{\alpha}_i \frac{\partial \overline{\boldsymbol{U}}_i}{\partial t} + \frac{1}{|T_i|}\sum_{j=1}^{4} |S_{ij}|\hat{\boldsymbol{n}} \cdot \boldsymbol{F}|_{S_{ij}} = \boldsymbol{J}_i, \tag{5}$$

where $\hat{\boldsymbol{n}}$ is the outward unit normal of the tetrahedron boundary $\partial T_i$ consisting of four triangular surfaces $S_{ij}$, $j = 1, \ldots, 4$. Fluxes are computed using physical properties on elements $T_i$ and $T_j$. Physical properties are the same inside a homogeneous medium and different on boundaries between dielectrics. To approximate the flux on each triangular surface $S_{ij}$, an upwind scheme based on the Steger–Warming flux vector splitting [37] is used. Then a third-order linear scheme [40; 26] is used to approximate the field components.

## 3. Multirate Runge–Kutta methods in 1D

Consider the semidiscrete problem defined by the ODE

$$u_t = Lu \tag{6}$$

on some bounded region $\Omega \subset \mathbb{R}$ with a given initial value $u(0) = u^0$. Here the operator $L$ represents the spatial approximation of the linear operator in the conservation law with some given order $p$. The computational domain is partitioned into two nonoverlapping subdomains $\Omega = D_1 \cup D_2 \cup \Gamma_{12}$, where $D_1$ has a fine mesh with size $h/2$ and $D_2$ has a coarse mesh with size $h$, and $\Gamma_{12} = \partial D_1 \cap \partial D_2$ is the boundary between $D_1$ and $D_2$. Assuming that the local time step satisfying the CFL condition on $D_2$ is $\Delta t$, then the local time step on $D_1$ is $\Delta t/2$. Denote by $L_1$ and $L_2$ two projections of the operator $L$ onto domains $D_1$ and $D_2$, respectively; then we can split the right-hand side of (6) as

$$u_t = L_1 u + L_2 u. \tag{7}$$

For the analysis of multirate Runge–Kutta schemes it is convenient to consider their partitioned form (MPRK) [10; 23; 34]. The $s$-stage multirate Runge–Kutta method for (7) with two levels of refinement (local time steps) can be written as

$$u^{(i)} = u^n + \Delta t \sum_{k=1,2} \sum_{j=1}^{i-1} a_{ij}^{(k)} L_k u^{(j)}, \quad i = 1, \dots, s, \tag{8}$$

$$u^{n+1} = u^n + \Delta t \sum_{k=1,2} \sum_{i=1}^{s} b_i^{(k)} L_k u^{(i)}. \tag{9}$$

It should be noted that the time-step factor is taken into account in the coefficients $a_{ij}^{(1)}$ and $a_{ij}^{(2)}$ and $s$ is the number of MPRK stages. The scheme (8)–(9) is internally consistent if [23]

$$c_i^{(1)} = c_i^{(2)}, \qquad c_i^{(k)} = \sum_{j=1}^{s} a_{ij}^{(k)}, \quad i = 1, \dots, s. \tag{10}$$

This condition ensures that the stage values on adjacent subdomains are consistent approximations to $u(t^n + c_i \Delta t)$. Failure to satisfy the internal consistency condition may lead to lower accuracy at interface points.

The accuracy of the MPRK schemes in the sense of a truncation error can be determined using the classic order conditions [21; 24; 1]. A few multirate schemes based on second-order Runge–Kutta methods satisfying second-order conditions exist in literature [10; 38]. But generalizations of these schemes by using third-order base methods do not automatically generate a third-order MPRK method. The number of conditions quickly increases with order, and it becomes challenging to satisfy all of them. For linear problems, however, this number is reduced. For the third-order scheme the order conditions are given by the following lemma.

**Lemma.** *The multirate partitioned Runge–Kutta method* (8)–(9), *where* $L_1$ *and* $L_2$ *are linear constant-coefficient operators*, *is third-order accurate if the following order conditions are satisfied*:

$$(\textit{first order}) \qquad (\boldsymbol{b}^{(k_1)})^T \mathbf{1} = 1, \qquad k_1, k_2 = 1, 2, \tag{11}$$

$$(\textit{second order}) \qquad (\boldsymbol{b}^{(k_1)})^T \boldsymbol{c}^{(k_2)} = \tfrac{1}{2}, \qquad k_1, k_2 = 1, 2, \tag{12}$$

$$(\textit{third order}) \qquad (\boldsymbol{b}^{(k_1)})^T \boldsymbol{A}^{(k_2)} \boldsymbol{c}^{(k_3)} = \tfrac{1}{6}, \quad k_1, k_2, k_3 = 1, 2. \tag{13}$$

*Proof.* The proof is based on the estimate of the local truncation error $\tau^{n+1} = u^{n+1} - v(t^{n+1})$ after the time step $\Delta t$, where $v$ is defined by

$$v_t = Lv, \qquad v(t^n) = u^n.$$

Using Taylor series expansion for $v(t^{n+1})$ and substituting (8) into (9), the following expression for the truncation error is derived:

$$\begin{aligned}\tau^{n+1} = \Delta t\Bigg[\sum_{k_1=1,2} (1 - (\boldsymbol{b}^{(k_1)})^T \mathbf{1}) L_{k_1}\Bigg] v^n \\ + \Delta t^2 \Bigg[\sum_{k_1,k_2=1,2} \big(\tfrac{1}{2} - (\boldsymbol{b}^{(k_1)})^T \boldsymbol{c}^{(k_2)}\big) L_{k_1} L_{k_2}\Bigg] v^n \\ + \Delta t^3 \Bigg[\sum_{k_1,k_2,k_3=1,2} \big(\tfrac{1}{6} - (\boldsymbol{b}^{(k_1)})^T \boldsymbol{A}^{(k_2)} \boldsymbol{c}^{(k_3)}\big) L_{k_1} L_{k_2} L_{k_3}\Bigg] v^n + O(\Delta t^4).\end{aligned}$$

The truncation error is $O(\Delta t^4)$ if the first three terms are zero. □

Now we consider two multirate schemes with a third-order SSP Runge–Kutta method as a base. The first scheme is an extension of the second-order scheme developed in [38] by Tang and Warnecke (MRK-TW). A generalization of their scheme for two time increments $\Delta t$ and $\Delta t/2$ with an arbitrary base method $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is given in Table 1, where

$$\boldsymbol{A}_1 = [\boldsymbol{A}\hat{\boldsymbol{e}}_1, \boldsymbol{Z}_{s,s-1}], \qquad \boldsymbol{A}_2 = \boldsymbol{A} - \boldsymbol{A}_1, \tag{14}$$

$$\boldsymbol{b}_1 = b_1 \hat{\boldsymbol{e}}_1, \quad \boldsymbol{b}_2 = \boldsymbol{b} - \boldsymbol{b}_1, \quad \hat{\boldsymbol{e}}_1 = [\underbrace{1, 0, \ldots, 0}_{s}]^T, \tag{15}$$

and $\boldsymbol{Z}_{s,s-1}$ is the $s \times (s-1)$ zero matrix.

$$D_1:\quad \begin{array}{c|c} \boldsymbol{c}^{(1)} & \boldsymbol{A}^{(1)} \\ \hline & [\boldsymbol{b}^{(1)}]^T \end{array} \quad = \quad \begin{array}{c|cc} \frac{1}{2}\boldsymbol{c} & \frac{1}{2}\boldsymbol{A} & \\ \frac{1}{2}\mathbf{1}+\frac{1}{2}\boldsymbol{c} & \frac{1}{2}\boldsymbol{b}^T\otimes\mathbf{1} & \frac{1}{2}\boldsymbol{A} \\ \hline & \frac{1}{2}\boldsymbol{b}^T & \frac{1}{2}\boldsymbol{b}^T \end{array}$$

$$D_2:\quad \begin{array}{c|c} \boldsymbol{c}^{(2)} & \boldsymbol{A}^{(2)} \\ \hline & [\boldsymbol{b}^{(2)}]^T \end{array} \quad = \quad \begin{array}{c|cc} \frac{1}{2}\boldsymbol{c} & \frac{1}{2}\boldsymbol{A} & \\ \frac{1}{2}\hat{\boldsymbol{e}}_1+\boldsymbol{c} & \frac{1}{2}\boldsymbol{b}^T\otimes\hat{\boldsymbol{e}}_1+\boldsymbol{A}_1 & \boldsymbol{A}_2 \\ \hline & \boldsymbol{b}_1^T & \boldsymbol{b}_2^T \end{array}$$

**Table 1.** MPRK-TW scheme for arbitrary base method $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ and time-step ratio 2.

**Theorem.** *The partitioned Runge–Kutta scheme defined by the Butcher tableau in Table 1 is internally consistent if*

$$\boldsymbol{c} = \mathbf{1} - \hat{\boldsymbol{e}}_1 \tag{16}$$

*and is second-order accurate if the base method $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is at least second-order accurate and satisfies*

$$b_1 = \tfrac{1}{2}. \tag{17}$$

*Moreover, it has at most second-order accurate coupling regardless of the base method.*

*Proof.* The proof of internal consistency is a straightforward application of the condition (10). Assuming that the base method $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ satisfies the second-order conditions, the coupling conditions ((12), $k_1 \neq k_2$) applied to the scheme in Table 1 give us

$$\begin{aligned} b_i^{(1)} c_i^{(2)} &= \tfrac{1}{2}\boldsymbol{b}^T\big(\tfrac{1}{2}\boldsymbol{c}+\tfrac{1}{2}\hat{\boldsymbol{e}}_1+\boldsymbol{c}\big) = \tfrac{3}{4}(\boldsymbol{b})^T\boldsymbol{c}+\tfrac{1}{4}b_1 = \tfrac{1}{2} && \iff && b_1 = \tfrac{1}{2}, \\ b_i^{(2)} c_i^{(1)} &= (b_1\hat{\boldsymbol{e}}_1^T)\tfrac{1}{2}\boldsymbol{c}+(\boldsymbol{b}-b_1\hat{\boldsymbol{e}}_1)^T\big(\tfrac{1}{2}\mathbf{1}+\tfrac{1}{2}\boldsymbol{c}\big) \\ &= \tfrac{1}{2}(\boldsymbol{b})^T\mathbf{1}-\tfrac{1}{2}b_1+\tfrac{1}{2}(\boldsymbol{b})^T\boldsymbol{c} = \tfrac{1}{2} && \iff && b_1 = \tfrac{1}{2}. \end{aligned}$$

Hence, the method is second-order accurate provided that $b_1 = \frac{1}{2}$. Assume that the base method $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ is second-order accurate and also satisfies the third-order conditions for linear problems ((13), $k_1 = k_2 = k_3$). One of the linear coupling conditions in (13) with $k_1 = 1$ and $k_2 = k_3 = 2$ gives

$$\begin{aligned} (\boldsymbol{b}^{(1)})^T\boldsymbol{A}^{(2)}\boldsymbol{c}^{(2)} &= \tfrac{1}{2}\boldsymbol{b}^T\tfrac{1}{2}\boldsymbol{A}\tfrac{1}{2}\boldsymbol{c}+\tfrac{1}{2}\boldsymbol{b}^T\big(\tfrac{1}{2}\boldsymbol{b}^T\otimes\hat{\boldsymbol{e}}_1+\boldsymbol{A}_1\big)\tfrac{1}{2}\boldsymbol{c}+\tfrac{1}{2}\boldsymbol{b}^T\boldsymbol{A}_2\big(\tfrac{1}{2}\hat{\boldsymbol{e}}_1+\boldsymbol{c}\big) \\ &= \tfrac{1}{8}\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{c}+\tfrac{1}{16}b_1+\tfrac{1}{2}\boldsymbol{b}^T\boldsymbol{A}\boldsymbol{c} = \tfrac{5}{48}+\tfrac{1}{16}b_1. \end{aligned}$$

Therefore, the second- and third-order conditions cannot hold together. □

It follows from the theorem that the partitioned Runge–Kutta scheme defined by the Butcher tableau in Table 1 is only first-order accurate with any third-order base method.

To get internal consistency for schemes of order $r > 2$, solutions on both sides of the interface $\Gamma_{12}$ need to be adjusted. One strategy that provides higher-order coupling for linear problems was proposed in [27]. It is third-order accurate for third-order SSP Runge–Kutta base methods for linear problems (MRK-LLH).

For linear problems the Runge–Kutta method can be written as

$$\boldsymbol{u} = \boldsymbol{C}\boldsymbol{T}_{\Delta t}\boldsymbol{L}_s\boldsymbol{u}^n, \tag{18}$$

$$u^{n+1} = u^n + \boldsymbol{b}^T\boldsymbol{L}\boldsymbol{u}, \tag{19}$$

where

$$\boldsymbol{u} = [u^{(1)}, u^{(2)}, \dots, u^{(s)}]^T, \qquad \boldsymbol{u}^n = [\underbrace{u^n, \dots, u^n}_{s}]^T,$$

$$\boldsymbol{L}_s = \mathrm{diag}\{I, L, L^2, \dots, L^{s-1}\}, \qquad \boldsymbol{L} = \mathrm{diag}\{\underbrace{L, L, \dots, L}_{s}\},$$

$$\boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & a_{21} & 0 & \cdots & 0 \\ 1 & \sum a_{3j} & a_{32}a_{21} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & 0 \\ 1 & \sum a_{sj} & \sum a_{sj}a_{jk} & \cdots & a_{s,s-1}\cdots a_{21} \end{bmatrix} \tag{20}$$

and

$$\boldsymbol{T}_{\Delta t} = \mathrm{diag}\{1, \Delta t, \dots, \Delta t^{s-1}\}. \tag{21}$$

Since for the linear case we have

$$L^i u|_{t=t^n} = \left.\frac{d^i u}{dt^i}\right|_{t=t^n}, \tag{22}$$

therefore the RK stage values $\boldsymbol{u}$ can be written in terms of time derivatives of $u^n$.

Now consider the partition $\Omega = D_1 \cup D_2 \cup \Gamma_{12}$ defined by the local time steps $\Delta t_1 = \Delta t/2$, and $\Delta t_2 = \Delta t$. First the solution is advanced on both subdomains from $t = t^n$ with their local time steps $\Delta t_1$ and $\Delta t_2$. The stage values at the time level $t^n$ inside of each subdomain are computed by

$$\boldsymbol{u}_k = \boldsymbol{C}\boldsymbol{T}_{\Delta t_k}\boldsymbol{d}\boldsymbol{u}^n, \quad k = 1, 2, \tag{23}$$

where

$$\boldsymbol{d}\boldsymbol{u}^n = \left[u, \frac{du}{dt}, \frac{d^2u}{dt^2}, \dots, \frac{d^{s-1}u}{dt^{s-1}}\right]^T_{t=t^n}. \tag{24}$$

To calculate the fluxes on the interface $\Gamma_{12}$ the stage values $\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{u}}_2$ are needed for time advancing from $t = t^n$ on $D_2$ and $D_1$, respectively. Using (23) the stages

$$D_1\colon\quad \begin{array}{c|c} \boldsymbol{c}^{(1)} & \boldsymbol{A}^{(1)} \\ \hline & [\boldsymbol{b}^{(1)}]^T \end{array} \;=\; \begin{array}{c|cc} \frac{1}{2}\boldsymbol{c} & \frac{1}{2}\boldsymbol{A} & \boldsymbol{0} \\ \frac{1}{2}\boldsymbol{1}+\frac{1}{2}\boldsymbol{c} & \frac{1}{2}\boldsymbol{b}^T\otimes\boldsymbol{1} & \frac{1}{2}\boldsymbol{A} \\ \hline & \frac{1}{2}\boldsymbol{b}^T & \frac{1}{2}\boldsymbol{b}^T \end{array}$$

$$D_2\colon\quad \begin{array}{c|c} \boldsymbol{c}^{(2)} & \boldsymbol{A}^{(2)} \\ \hline & [\boldsymbol{b}^{(2)}]^T \end{array} \;=\; \begin{array}{c|cc} \frac{1}{2}\boldsymbol{c} & \frac{1}{2}\boldsymbol{A} & \boldsymbol{0} \\ \boldsymbol{q} & \boldsymbol{Q} & \boldsymbol{0} \\ \hline & \boldsymbol{b}^T\boldsymbol{G}^{(1)} & \boldsymbol{0} \end{array}$$

**Table 2.** MPRK-LLH scheme for arbitrary base method $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{c})$ and time-step ratio 2.

$\tilde{\boldsymbol{u}}_1$ and $\tilde{\boldsymbol{u}}_2$ are obtained using the coupling [27]

$$[\tilde{\boldsymbol{u}}_1]_{t^n} = \boldsymbol{C}\boldsymbol{T}_{\Delta t_2}\boldsymbol{d}\boldsymbol{u}^n = \boldsymbol{C}\boldsymbol{T}_{\Delta t_2}\boldsymbol{T}^{-1}_{\Delta t_1}\boldsymbol{C}^{-1}\boldsymbol{u}_1 = \boldsymbol{G}^{(1)}\boldsymbol{u}_1, \tag{25}$$

$$[\tilde{\boldsymbol{u}}_2]_{t^n} = \boldsymbol{C}\boldsymbol{T}_{\Delta t_1}\boldsymbol{d}\boldsymbol{u}^n = \boldsymbol{C}\boldsymbol{T}_{\Delta t_1}\boldsymbol{T}^{-1}_{\Delta t_2}\boldsymbol{C}^{-1}\boldsymbol{u}_2 = \boldsymbol{G}^{(2)}\boldsymbol{u}_2. \tag{26}$$

Matrices $\boldsymbol{G}^{(1)}$ and $\boldsymbol{G}^{(2)}$ are lower triangular and have the properties

$$\boldsymbol{G}^{(1)}\boldsymbol{G}^{(2)} = \boldsymbol{G}^{(2)}\boldsymbol{G}^{(1)} = \boldsymbol{I}_s, \tag{27}$$

$$\sum_{j=1}^{s} G^{(1)}_{ij} = \sum_{j=1}^{s} G^{(2)}_{ij} = 1, \tag{28}$$

$$\boldsymbol{G}^{(1)}\tilde{\boldsymbol{u}}_2 = \boldsymbol{u}_2, \qquad \boldsymbol{G}^{(2)}\tilde{\boldsymbol{u}}_1 = \boldsymbol{u}_1. \tag{29}$$

At the second step the solution is advanced on the fine mesh only using coupling stage values $\tilde{\boldsymbol{u}}_2$ at the time level $t = t^n + \Delta t_1$ computed by [27]

$$[\tilde{\boldsymbol{u}}_2]_{t^n+\Delta t_1} = \boldsymbol{C}\boldsymbol{T}_{\Delta t_1}\boldsymbol{H}_{\Delta t_1}\boldsymbol{T}^{-1}_{\Delta t_2}\boldsymbol{C}^{-1}\boldsymbol{u}_2 =: \boldsymbol{K}\boldsymbol{u}_2, \tag{30}$$

where

$$\boldsymbol{H}_{\Delta t} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 & \cdots & \Delta t^{s-1}/(s-1)! \\ 0 & 1 & \Delta t & \cdots & \Delta t^{s-2}/(s-2)! \\ 0 & 0 & 1 & \cdots & \Delta t^{s-3}/(s-3)! \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix}. \tag{31}$$

The MRK-LLH method described above can be written in the Butcher tableau form shown in Table 2, where

$$\boldsymbol{Q} = \boldsymbol{K}\boldsymbol{A}\boldsymbol{G}^{(1)}, \tag{32}$$

and $\boldsymbol{q} = [q_1, \dots, q_s]^T$, with

$$q_i = \sum_{j=1}^{s} Q_{ij} = \sum_{j=1}^{s-1}\Bigg(\sum_{k=j+1}^{s}\sum_{l=j}^{k-1} K_{ik}a_{kl}G^{(1)}_{lj}\Bigg). \tag{33}$$

Consistency and accuracy analysis for the MPRK-LLH scheme can be summarized by the following theorem.

**Theorem.** *The partitioned Runge–Kutta scheme defined by the Butcher tableau in Table 2 is internally consistent and third-order accurate with three-stage SSP RK*3 *schemes for linear problems.*

*Proof.* The scheme is internally consistent by design. It can be shown by verifying the condition

$$\boldsymbol{q} = \tfrac{1}{2}\boldsymbol{1} + \tfrac{1}{2}\boldsymbol{c}. \tag{34}$$

It also follows from the application of the order conditions (11)–(13) that the scheme MPRK-LLH is third-order accurate for three-stage RK3 base methods for linear problems. □

The extension of the scheme to any arbitrary time-step ratio is straightforward [27]. To update stage values the following general coupling expressions replace (25)–(26) and (30):

$$[\tilde{\boldsymbol{u}}_1]_{t^{n,k_2}} = \boldsymbol{K}^{(1)}_{\Delta t_{k_2,k_1}} [\boldsymbol{u}_2]_{t^{n,k_1}}, \quad \boldsymbol{K}^{(1)}_{\Delta t_{k_2,k_1}} = \boldsymbol{C}\boldsymbol{T}_{\Delta t_2}\boldsymbol{H}_{\Delta t_{k_2,k_1}}\boldsymbol{T}^{-1}_{\Delta t_1}\boldsymbol{C}^{-1}, \tag{35}$$

$$[\tilde{\boldsymbol{u}}_2]_{t^{n,k_2}} = \boldsymbol{K}^{(2)}_{\Delta t_{k_2,k_1}} [\boldsymbol{u}_1]_{t^{n,k_1}}, \quad \boldsymbol{K}^{(2)}_{\Delta t_{k_2,k_1}} = \boldsymbol{C}\boldsymbol{T}_{\Delta t_1}\boldsymbol{H}_{\Delta t_{k_2,k_1}}\boldsymbol{T}^{-1}_{\Delta t_2}\boldsymbol{C}^{-1}. \tag{36}$$

In the next section we present the steps to implement both schemes on 3D meshes with arbitrary time-step ratios.

## 4. MRK scheme in 3D

Consider a semidiscrete system of Maxwell's equations (5) written as

$$\boldsymbol{U}_t = L\boldsymbol{U}, \tag{37}$$

and defined on a computational domain with mesh $\overline{\Omega}_T = \bigcup_{i=1}^{N} \overline{T}_i$. This domain is partitioned into $K$ multirate groups of elements $\overline{\Omega}_T = \bigcup_{k=1}^{K} \overline{D^{(k)}}$ using a local stability criterion. Let $\{\Delta\tau_i\}_{i=1}^{N}$ be a set of characteristic stable time steps obtained by [7]

$$\Delta\tau_i \le \frac{|T_i|}{c_i \sum_{j\in\mathcal{I}_i} |S_{ij}|}, \tag{38}$$

for each cell $T_i$ with volume $|T_i|$, where $\mathcal{I}_i$ is the set of indexes of neighboring elements, and $|S_{ij}|$ is the area of the face shared by element $T_i$ and its $j$-th neighbor. Let $\Delta t_{\min} = \min_i\{\Delta\tau_i\}$ and $\Delta t_{\max} = \max_i\{\Delta\tau_i\}$. Each time step $\Delta t_k$ can be defined as a product of $\Delta t_{\min}$ and some rational number $0 \le p_k \le \Delta t_{\max}/\Delta t_{\min}$. Then $K$ multirate groups can be defined as

$$D^{(k)} = \begin{cases} \{T_i \in \Omega,\ \Delta\tau_i \in [\Delta t_k, \Delta t_{k+1})\}, & k = 1, \dots, K-1, \\ \{T_i \in \Omega,\ \Delta\tau_i \in [\Delta t_k, \Delta t_{\max})\}, & k = K. \end{cases} \tag{39}$$

Each multirate group consists of elements of bulk group $D^{(k)}_{\text{bulk}}$ and inner buffer group $D^{(k)}(0)$. Bulk group $D^{(k)}_{\text{bulk}}$ includes all elements of $D^{(k)}$ that are sufficiently far from the boundary $\Gamma_k = \partial D^{(k)} \cap \big(\bigcup_{l=1,\, l\neq k}^{K} \partial D^{(l)}\big)$; therefore, time integration on these elements does not depend on values from neighboring multirate groups. The size of the inner buffer $D^{(k)}(0)$ depends on the order of finite volume approximation and consists of elements of $D^{(k)}$ nearest to $\Gamma_k$ for which time integration involves values from adjacent multirate groups. In addition to bulk and inner buffer groups we need to define outer buffer groups, where the values of adjacent multirate groups are updated to ensure proper coupling. Let $\Delta t$ be the global time step, at which the solution in all multirate groups is synchronized, and the final time is achieved after $N^t$ global time integrations, i.e., $T = N^t \Delta t$. Assuming that each global time step from $t^n$ to $t^{n+1}$ consists of $m$ local multirate stages, we associate to each multirate group $D^{(k)}$ the local time $t^{n,l}_k$, $l \in \{1, \dots, m\}$, at the $l$-th multirate stage. The global time $t^{n,l}$ is defined by local times $t^{n,l}_k$, and its definition depends on the multirate scheme.

The most common definition of local time steps [15; 29; 31; 35] is given by

$$\{\Delta t_k\}_{k=1}^{K_2} = \{2^{k-1} \Delta t_{\min}\}_{k=1}^{K_2}, \qquad K_2 = \left\lfloor \log_2 \frac{\Delta t_{\max}}{\Delta t_{\min}} \right\rfloor + 1.$$

Another set of factors $\{p_k\}_{k=1}^{K}$ that we found to give better distribution of local time steps is given by

$$\{p_k\}_{k=1}^{K} = \{K/k \Delta t_{\min}\}_{k=0}^{K},$$

where

$$K = \left\lfloor \frac{\Delta t_{\max}}{\Delta t_{\min}} \right\rfloor. \tag{40}$$

This partition is then optimized by varying the parameters $\Delta t_{\min}$ and $\kappa$ and removing unnecessary groups. We will refer to this partition as optimized partition (OP). The outline of the optimization procedure is the following:

(1) Using the values of $\Delta t_{\min}$ and $K$ defined by (40) we introduce two parameters for the new multirate partition

$$\Delta t^*_{\min} = \alpha \Delta t_{\min}, \qquad \alpha \in [0.8, 1], \tag{41}$$

$$K^* = \beta K, \qquad \beta \in [0.8, 1.2]. \tag{42}$$

It should be noted that broader ranges of values for the parameters $\alpha$ and $\beta$ introduced too many local minima (and repeating time-step distributions) for the optimization procedure to be efficient.

(2) For a randomly chosen pair $(\Delta t^*_{\min}, K^*)$ using a certain search procedure,

(a) construct a multirate partition;

  (b) remove unnecessary multirate groups: if a subdomain with $\Delta t_k$ consists of a few isolated elements, add it to the subdomain with $\Delta t_{k-1}$; and
  (c) estimate theoretical speedup from the resulting partition.

(3) Go to step 2 if the convergence criterion is not satisfied. If converged, take the best estimated partition as the final choice.

In the present work an improved controlled random search algorithm by [28] was used as a searching procedure. It was run once before the simulation with a limit of 200 iterations, and usually took around 30 iterations to converge. The final partition is defined by the optimal pair $(\Delta t^*_{\min}, K^*)$ and after merging of unnecessary MRK groups has the total number of multirate groups $\leq K^*$. The convergence criterion is based on the theoretical speedup formula given by

$$S = \frac{\Delta t_{\min}^{-1} s N}{\sum_{k=1}^{K} \Delta t_k^{-1} s N_{D^{(k)}}}, \tag{43}$$

where $N$ is the total number of mesh elements, $s$ is the number of Runge–Kutta stages, and $N_{D^{(k)}}$ is the number of elements in the $D^{(k)}$ multirate group. During the initialization, the local time-step partition is computed and subdomains are determined. Multirate partitions are defined so that all local times $t_k^{n,l}$ are synchronized at some global time step $\Delta t$. As a result the computational process can be divided into $N$ blocks with global time step $\Delta t$. In this work only static meshes were used in simulations. The same idea for multirate partitioning can be applied to dynamic mesh refinement. In this case, multirate groups have to be defined for each mesh refinement at minimum computational cost.

**4.1.** ***Tang–Warnecke scheme.*** The coupling in the MR-TW scheme is done by projecting the solution using the Runge–Kutta step in the adjacent multirate group. Therefore, with three-stage RK3 base scheme the outer buffer consists of three-stage coupling groups $D^{(k)}(q)$, $q = 1, \ldots, 3$.

Consider the partition into $K$ multirate groups with time steps $\Delta t_k$ defined by any partition method. Let $m$ be the number of local time updates from $t^n = t^{n,0}$ to $t^{n+1} = t^n + \Delta t = t^{n,m}$ at which all multirate groups are synchronized. Local times $t_k^{n,l}$, $1 \leq l \leq m$, are updated at the beginning of the time cycle by

$$t_k^{n,l} = \begin{cases} t_k^{n,l-1} + \Delta t_k & \text{if } t_k^{n,l-1} = t^{n,l-1}, \\ t_k^{n,l-1} & \text{if } t_k^{n,l-1} > t^{n,l-1}. \end{cases} \tag{44}$$

Then the global time corresponding to the $l$-th multirate stage is obtained by

$$t^{n,l} = \min_k t_k^{n,l}. \tag{45}$$

At the beginning of each multirate stage $l$ the initial stage values are given by

$$\boldsymbol{W}^{(1)} = \begin{cases} \boldsymbol{U}_k^{n,l-1} & \text{on } D^{(k)}, \\ \boldsymbol{U}_j^{n,l^*} & \text{on } \bigcup_{j=1, j\neq k}^{K} \big(D^{(j)} \cap \big(\bigcup_{r=1}^{s} D^{(k)}(r)\big)\big). \end{cases} \tag{46}$$

Here $l^* \leq l-1$ is the last multirate stage with $t_k^{n,l^*} = t_j^{n,l^*}$. The $q$-th stage value of the Runge–Kutta scheme on multirate groups $D^{(k)}$ is then computed by

$$\boldsymbol{U}_k^{(q)} = \boldsymbol{U}_k^{n,l-1} + \Delta t_k \sum_{r=1}^{q-1} a_{qr} L_k \boldsymbol{W}^{(r)}, \quad q = 2, \ldots, s, \tag{47}$$

and coupling values denoted by $\boldsymbol{V}_j^{(q)}$ are computed in the outer buffer of $D^{(k)}$ by

$$\boldsymbol{V}_j^{(q)} = \boldsymbol{U}_j^{n,l^*} + \Delta t_{k,j} \sum_{r=1}^{q-1} a_{qr} L_j \boldsymbol{W}^{(r)}, \tag{48}$$

where $\Delta t_{k,j} = t_k^{n,l} - t_j^{n,l^*}$ and

$$\boldsymbol{W}^{(q)} = \begin{cases} \boldsymbol{U}_k^{(q)} & \text{on } D^{(k)}, \\ \boldsymbol{V}_j^{(q)} & \text{on } \bigcup_{j=1, j\neq k}^{K} \big(D^{(j)} \cap \big(\bigcup_{r=1}^{s+1-q} D^{(k)}(r)\big)\big), \end{cases} \quad q = 2, \ldots, s.$$

**4.2.** ***Liu–Li–Hu linear scheme.*** The coupling in the MRK-LLH is done by modifying the latest stage values in cells closest to the multirate interface. The outer buffer includes only one coupling group $D^k(1)$.

Consider a partition into $K$ multirate groups with time steps $\Delta t_k$ and $m$ local time updates from $t^n = t^{n,0}$ to $t^{n+1} = t^n + \Delta t = t^{n,m}$. Local times $t_k^{n,l}$, $l \in \{0, \ldots, m-1\}$, associated with each multirate group $D^{(k)}$ are updated at the end of the $l$-th stage by

$$t_k^{n,l+1} = \begin{cases} t_k^{n,l} + \Delta t_k & \text{if } t_k^{n,l} + \Delta t_k = t^{n,l+1}, \\ t_k^{n,l} & \text{if } t_k^{n,l} + \Delta t_k > t^{n,l+1}, \end{cases} \tag{49}$$

where

$$t^{n,l+1} = \min_k (t_k^{n,l} + \Delta t_k). \tag{50}$$

At each multirate stage $l$ for every $D^{(k)}$ with $t_k^{n,l} = t^{n,l}$ the coupling RK stage values $\boldsymbol{V}_j$ are computed in the outer buffer $\bigcup_{j=1, j\neq k}^{K} (D^{(j)} \cap D^{(k)}(1))$ by

$$\boldsymbol{V}_j^{(q)} = \begin{cases} \sum_{r=1}^{q} [\boldsymbol{C}\boldsymbol{T}_{\Delta t_j} \boldsymbol{T}_{\Delta t_k}^{-1} \boldsymbol{C}^{-1}]_{qr} \boldsymbol{U}_j^{(r)} & \text{if } t_j^{n,l} = t_k^{n,l}, \\ \sum_{r=1}^{s} [\boldsymbol{C}\boldsymbol{T}_{\Delta t_j} \boldsymbol{H}_{\Delta t_{k,j}^{n,l}} \boldsymbol{T}_{\Delta t_k}^{-1} \boldsymbol{C}^{-1}]_{qr} \boldsymbol{U}_j^{(r)} & \text{if } t_j^{n,l} < t_k^{n,l}, \end{cases} \tag{51}$$

where $\Delta t_{k,j}^{n,l} = t_k^{n,l} - t_j^{n,l}$, $\boldsymbol{U}_j^{(r)}$ are the RK stage values on $D^{(j)}$ at $t_j^{n,l}$, and matrices $\boldsymbol{C}$, $\boldsymbol{T}_{\Delta t}$, and $\boldsymbol{H}_{\Delta t}$ are defined by (20), (21), and (31), respectively. Then the time integration is performed on $D^{(k)}$. There are no additional RK steps in the outer buffer, since the coupling values are defined by (51). Therefore, in this algorithm we avoid additional costly computations of fluxes in the outer buffer.

## 5. Numerical experiments

All numerical experiments were completed with double precision on a computer with a four-core Intel i7-4790K CPU. The computational code of the finite volume engine was written in C++ with OpenMP and compiled using GCC. Tetrahedral meshes for all 3D problems considered in this work were generated using the open source software Gmsh version 2.7.1. As was mentioned in Section 4, the MRK3-TW scheme requires more flux computations while MRK3-LLH uses more coupling steps. To assess schemes' efficiency, in our numerical experiments we compared both CPU time and the total number of flux computations for single-rate RK3 and multirate schemes. Flux computation is the most computationally expensive operation and for large meshes it takes over 90% of CPU time (94% in the example on page 83). At the same time, on small meshes with large time-step ratio this percentage is lower and the speedup of MRK3-LLH compared to the RK3-TW scheme is diminished with too many coupling steps. Therefore, in some examples RK3-TW slightly outperforms MRK3-LLH in terms of CPU time.

**Example** (1D linear advection equation). Consider the linear advection problem

$$u_t + u_x = 0, \quad x \in \Omega = (-1, 1), \tag{52}$$

$$u(x, 0) = \sin(\pi x), \tag{53}$$

with periodic boundary conditions. The computational domain consists of two subdomains $D_1 = (-1, 0)$ with grid size $h/2$, and $D_2 = (0, 1)$ with grid size $h$. For the space approximation a finite volume scheme based on a third-order WENO reconstruction [25] is employed. Convergence results for MRK3-TW and MRK3-LLH are compared to the ones by the non-MRK SSP RK3 scheme on a uniform grid (see Table 3).

**Example** (PEC sphere). Consider the classical scattering problem of a plane wave at a PEC sphere for which the analytic series solution is known [22; 5]. The computational domain is represented by a sphere of radius 3 m with a sphere (PEC) of radius 0.5 m cut out at the origin. The $x$ component of the electric field of the incident plane wave $E_x^I$ is given by the derivative of the Gaussian pulse:

$$E_x^I = -2\frac{t - t_0}{b^2} A e^{-(t-t_0)^2/b^2}, \tag{54}$$

where $A = 1.7489 \times 10^{-9}\,\frac{\mathrm{V \cdot s}}{\mathrm{m}}$, $b = 1.5 \times 10^{-9}$ s, and $t_0 = 6 \times 10^{-9}$ s.

The average linear cell size near the PEC surface is 0.0225 m, and on the outer boundary of the domain it is 0.15 m. The resulting nonuniform mesh has linear

| | RK3 | | MRK3-TW | | MRK3-LLH | |
|---|---|---|---|---|---|---|
| $h^{-1}$ | $l_2(u)$ | $r_2(u)$ | $l_2(u)$ | $r_2(u)$ | $l_2(u)$ | $r_2(u)$ |
| 100 | $9.006 \times 10^{-6}$ | | $3.895 \times 10^{-3}$ | | $1.086 \times 10^{-5}$ | |
| 200 | $1.126 \times 10^{-6}$ | 3.00 | $1.963 \times 10^{-3}$ | 0.99 | $1.357 \times 10^{-6}$ | 3.00 |
| 400 | $1.407 \times 10^{-7}$ | 3.00 | $9.863 \times 10^{-4}$ | 0.99 | $1.695 \times 10^{-7}$ | 3.00 |
| 800 | $1.759 \times 10^{-8}$ | 3.00 | $4.945 \times 10^{-4}$ | 1.00 | $2.118 \times 10^{-8}$ | 3.00 |
| 1 600 | $2.198 \times 10^{-9}$ | 3.00 | $2.477 \times 10^{-4}$ | 1.00 | $2.647 \times 10^{-9}$ | 3.00 |
| 3 200 | $2.751 \times 10^{-10}$ | 3.00 | $1.240 \times 10^{-4}$ | 1.00 | $3.310 \times 10^{-10}$ | 3.00 |
| 6 400 | $3.507 \times 10^{-11}$ | 2.97 | $6.203 \times 10^{-5}$ | 1.00 | $4.170 \times 10^{-11}$ | 2.99 |

**Table 3.** Convergence of RK3 and MRK3 schemes for the linear advection equation with initial data $u(x, 0) = \sin(\pi x)$ at $T = 1$. Here $r_2(u) = \log_2(l_2(u^{[h]})/l_2(u^{[h/2]}))$.

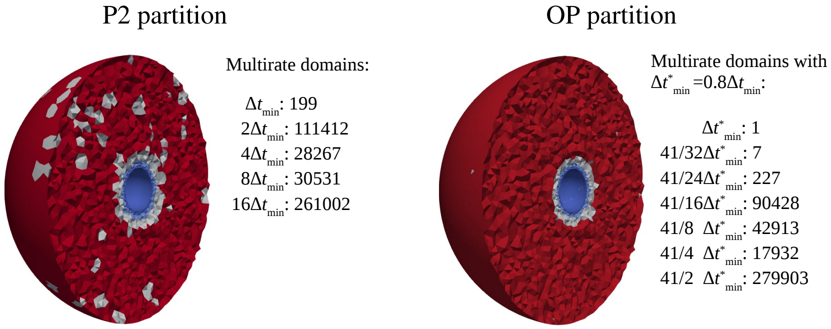

P2 partition
Multirate domains:
$\Delta t_{\min}$: 199
$2\Delta t_{\min}$: 111412
$4\Delta t_{\min}$: 28267
$8\Delta t_{\min}$: 30531
$16\Delta t_{\min}$: 261002
OP partition
Multirate domains with $\Delta t^*_{\min}=0.8\Delta t_{\min}$:
$\Delta t^*_{\min}$: 1
$41/32\Delta t^*_{\min}$: 7
$41/24\Delta t^*_{\min}$: 227
$41/16\Delta t^*_{\min}$: 90428
$41/8\ \Delta t^*_{\min}$: 42913
$41/4\ \Delta t^*_{\min}$: 17932
$41/2\ \Delta t^*_{\min}$: 279903

**Figure 1.** Scattering from PEC sphere: multirate domain partition for the mesh with linear cells size ratio 1 : 6.667.

cell-size ratio 1 : 6.667, and consists of 431 411 tetrahedra with 14 374 of them containing a PEC face. Two types of partitions used in our experiments with the MRK3-TW and MRK3-LLH schemes are shown of Figure 1. The time-domain solutions for the electric field at a side-scatter observation point by two MRK3 schemes and single-rate RK3 are shown in Figure 2. The maximum errors at four observation points are shown in Table 4. The error plots in Figure 2 show that the same accuracy is obtained with the MRK3-LLH-OP scheme as with the single-rate RK3 method. At the same time, the error of the solution obtained by the MRK3-TW-OP scheme is much larger. This demonstrates only first-order accuracy of the MRK3-TW scheme as in 1D analysis. The same conclusions can be drawn from the errors presented in Table 4. A comparison of numerical efficiency for both multirate schemes against the single-rate RK3 is shown in Table 5. While both schemes have faster CPU time than a single-rate scheme, in this example, CPU performance of the MRK3-LLH scheme is higher due to fewer interface flux computations required. It should be noted that P2 partition uses the largest time step as $\Delta t_g$, and in OP

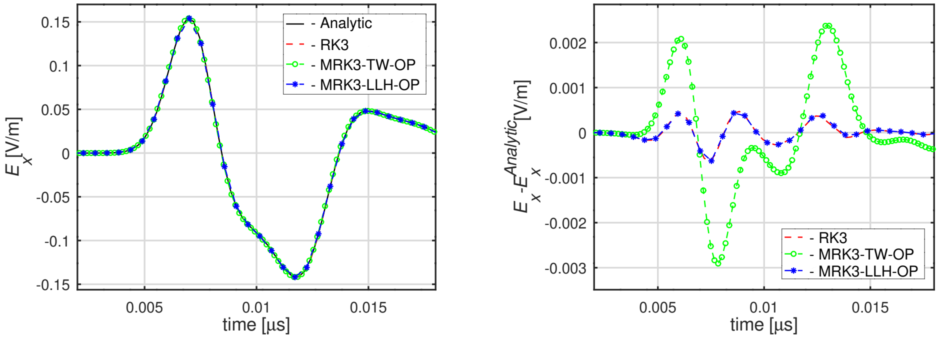


**Figure 2.** Scattering from PEC sphere: time-domain solution at side-scatter observation point $(-1.15, 0, 0)$ using RK3 and MRK3 schemes.

| scheme | side-scatter $(1.15, 0, 0)$ | side-scatter $(-1.15, 0, 0)$ | forward-scatter $(0, 0, 1.15)$ | back-scatter $(0, 0, -1.15)$ |
|---|---|---|---|---|
| RK3 | $1.1842 \times 10^{-3}$ | $1.1788 \times 10^{-3}$ | $8.2511 \times 10^{-3}$ | $3.1686 \times 10^{-3}$ |
| MRK3-TW-P2 | $2.3026 \times 10^{-3}$ | $3.2627 \times 10^{-3}$ | $1.3442 \times 10^{-2}$ | $5.4663 \times 10^{-3}$ |
| MRK3-TW-OP | $2.2546 \times 10^{-3}$ | $2.8729 \times 10^{-3}$ | $1.3193 \times 10^{-2}$ | $5.5634 \times 10^{-3}$ |
| MRK3-LLH-P2 | $1.1519 \times 10^{-3}$ | $9.3847 \times 10^{-4}$ | $8.0727 \times 10^{-3}$ | $3.1817 \times 10^{-3}$ |
| MRK3-LLH-OP | $1.2615 \times 10^{-3}$ | $1.2559 \times 10^{-3}$ | $7.8927 \times 10^{-3}$ | $3.1663 \times 10^{-3}$ |

**Table 4.** PEC sphere: $\max_n |E_x(t^n) - E_x^{\text{Analytic}}(t^n)|$ at observation points for RK3 and MRK3.

| scheme | # of $\Delta t_g$ | $n(LU)$ | $\frac{n(LU)_{\text{RK3}}}{n(LU)}$ | CPU [ms] | $\frac{\text{CPU}_{\text{RK3}}}{\text{CPU}}$ |
|---|---|---|---|---|---|
| RK3 | 7 245 | 9 376 718 085 | 1 | 6 498 449 | 1 |
| MRK3-TW-P2 | 453 | 2 325 701 094 | 4.03 | 1 893 579 | 3.43 |
| MRK3-TW-OP | 221 | 2 182 108 032 | 4.3 | 1 986 033 | 3.27 |
| MRK3-LLH-P2 | 453 | 2 062 756 791 | 4.55 | 1 587 369 | 4.09 |
| MRK3-LLH-OP | 230 | 1 944 214 380 | 4.82 | 1 586 064 | 4.1 |

**Table 5.** PEC sphere: performance of MRK3 schemes compared to single-rate RK3 for domain partitions shown on Figure 1, here $n(LU)$ is the number of flux operations which is the most computationally expensive operation.

partition $\Delta t_g$ (time step to synchronize solutions across multirate domains) is twice as large as the largest time step (see Figure 1). Therefore, the number of $\Delta t_g$ steps in Table 5 is equivalent to the number of synchronization steps, not the number of largest time steps.

**Example** (parallel-plate waveguide). A parallel-plate waveguide is represented by a cubic domain with two faces parallel to the $xy$-plane being PEC plates, and two faces parallel to the $zx$-plane being PMC plates. A plane-wave excited on the port

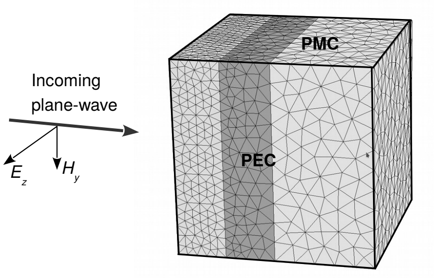


**Figure 3.** Parallel-plate waveguide: geometry and mesh.

| | RK3 | | MRK3-TW | | MRK3-LLH | |
|---|---|---|---|---|---|---|
| # of cells | $L^2$ error | order | $L^2$ error | order | $L^2$ error | order |
| 8 040 | $6.824139 \times 10^{-3}$ | | $1.929949 \times 10^{-2}$ | | $6.890073 \times 10^{-3}$ | |
| 64 076 | $9.177931 \times 10^{-4}$ | 2.89 | $8.110855 \times 10^{-3}$ | 1.25 | $9.246888 \times 10^{-4}$ | 2.9 |
| 554 668 | $1.107034 \times 10^{-4}$ | 3.05 | $4.385090 \times 10^{-3}$ | 0.89 | $1.112703 \times 10^{-4}$ | 3.05 |

**Table 6.** Parallel-plate waveguide: $L^2$ errors at $T = lc_0^{-1}$ ($l = 2\,\text{m}$) using RK3 and MRK3 schemes.

$x = -1$ and propagating in the $x$-direction is given by

$$E_z^{\text{in}} = f(t), \qquad H_y^{\text{in}} = -f(t)\sqrt{\epsilon/\mu}, \qquad E_x^{\text{in}} = E_y^{\text{in}} = H_x^{\text{in}} = H_z^{\text{in}} = 0, \tag{55}$$

where $f(t)$ is defined by the Gaussian pulse

$$f(t) = e^{-(t-t_0)^2/b^2}, \quad b = 1.2 \times 10^{-9}\,[\text{s}],\ t_0 = c_0^{-1}\,[\text{s}]. \tag{56}$$

Experiments are performed on three meshes with fine mesh linear size $\Delta x$ equal to 0.025, 0.05, and 0.1 m, and coarse mesh size $2\Delta x$. An example of problem geometry and mesh is shown in Figure 3. Convergence results are presented in Table 6. On each mesh we compute the $L^2$ error at time $T = 2c_0^{-1}$ by

$$l_2(\boldsymbol{U}(T)) = \frac{\left[\sum_{i=1}^{N} |T_i| \sum_{j=1}^{3} \frac{1}{2}(\epsilon_r \epsilon_0 (\bar{E}_i^j)^2 + \mu_r \mu_0 (\bar{H}_i^j)^2)\right]^{1/2}}{\left[\epsilon_0 \sum_{i=1}^{N} |T_i|\right]^{1/2}}. \tag{57}$$

In another experiment, an inhomogeneous mesh with linear cell-size ratio 1 : 160 was generated similar to the example from [15]. The plane wave (55) uses the pulse given as one wavelength of a cosine function

$$f(t) = \tfrac{1}{2}(1 + \cos(2\pi c_0(t - t_0)))\theta(t - t_s)\theta(t_e - t), \tag{58}$$

where $t_0 = 0.54c_0^{-1}$ [s], $t_s = 1.04c_0^{-1}$ [s], $t_e = 2.04c_0^{-1}$ [s], and $\theta(t)$ is the Heaviside function. A schematic representation of the geometry and resulting mesh are shown

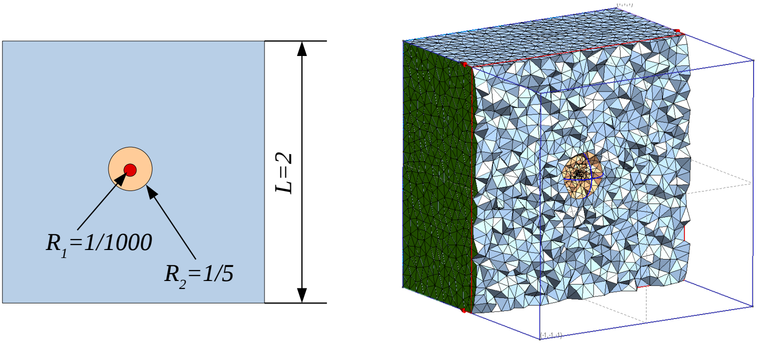


**Figure 4.** Parallel-plate waveguide: inhomogeneous mesh with linear cell-size ratio 1 : 160.

| scheme | # of $\Delta t_g$ | $n(LU)$ | $\frac{n(LU)_{\text{RK3}}}{n(LU)}$ | CPU [ms] | $\frac{\text{CPU}_{\text{RK3}}}{\text{CPU}}$ | $L^2$ error |
|---|---|---|---|---|---|---|
| RK3 | 51 978 | 13 180 477 284 | 1 | 9 130 297 | 1 | $5.7813\times10^{-3}$ |
| MRK3-TW-P2 | 204 | 232 562 448 | 56.68 | 296 687 | 30.77 | $1.3733\times10^{-2}$ |
| MRK3-TW-OP | 136 | 189 863 344 | 69.42 | 271 379 | 33.64 | $1.2976\times10^{-2}$ |
| MRK3-LLH-P2 | 204 | 231 063 048 | 57.04 | 378 350 | 24.13 | $5.6292\times10^{-3}$ |
| MRK3-LLH-OP | 136 | 173 453 040 | 75.99 | 366 928 | 24.88 | $5.7684\times10^{-3}$ |

**Table 7.** Parallel-plate waveguide with mesh-size ratio 1 : 160: performance of MRK3 schemes compared to single-rate RK3.

in Figure 4. The region defined by a sphere with radius $R_1$ has the smallest elements with linear size $R_1/1.6$. Area between spheres with radii $R_1$ and $R_2$ provide gradual transition to the coarsest mesh with average linear cell size 0.1. The resulting mesh has linear cell-size ratio 1 : 160 and contains 84 526 tetrahedra with fewer than 200 of elements of the smallest size. Using the P2 partition, the computational domain is divided into 9 multirate groups with the maximum time-step ratio 1 : 256. In this partition 0.12% of elements belong to multirate group with the smallest time step $\Delta t_{\min}$ and 86% to the group with time step $128\Delta t_{\min}$. OP partition divides the computational domain into 10 multirate groups with time step ratio 1 : 192 and global synchronization time step $384\Delta t^*_{\min}$. In this partition 0.14% of elementsbelong to the multirate group with the smallest time step $\Delta t^*_{\min}$ and 95.6% to the group with time step $192\Delta t^*_{\min}$. Numerical speedup achieved by the MRK3-TW and MRK3-LLH schemes is presented in Table 7. The results demonstrate greater speedup than in [15] even with the third-order scheme where coupling is more expensive due to the flexibility in time-step ratio. Speedup achieved by MRK3-TW is noticeably higher in this example. It can be explained by the fact that this test case uses a small mesh and the weight of additional flux computations needed for coupling in MRK3-TW turns out to be less costly than higher-order coupling used in MRK3-LLH.

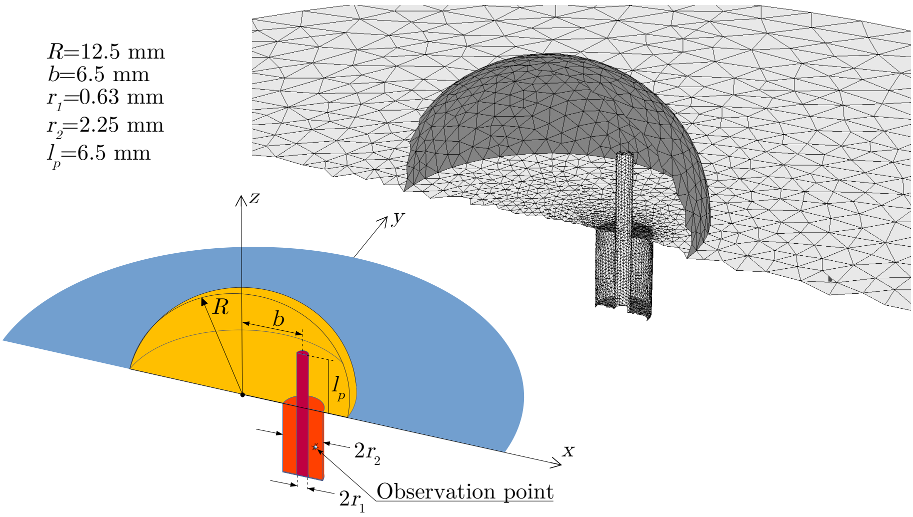


**Figure 5.** DRA: geometry and mesh.

**Example** (probe-fed hemispherical DRA). To evaluate the robustness of multirate schemes on a practical EM problem, we use an example of the coaxial probe-fed hemispherical DRA experiment from [6]. That work contained a comparison of a second-order finite volume time domain scheme against a simulation using the commercial software HFSS for a large dielectric hemisphere antenna withsmall feed coaxial cable. The DRA including all required geometrical parameters is presented in Figure 5. The outer boundary of the computational domain is an ellipsoid with absorbing boundary conditions from (1). Computational results are compared with the ones presented in [6] for the same set of parameters. As in [6] an $S_{11}$ parameter of the antenna is computed. Baumann [6] used the entire port section for the original way of reflection coefficient computations. Because of the third-order accuracy we are able to compute the return loss from a single observation point. To do that as in [6] we impose an analytic field onto the coaxial cable entrance port with wide-enough Gaussian (56) to cover the desirable frequency domain. Then we register the field $E_2^{\text{point}}(t)$ at one observation point with coordinates (0.00144, 0.0065, −0.00475), and compute the analytic field $E_2^{\text{coax}}(t)$ in the coaxial cable in the same point. Then the reflected coefficient is computed at several frequency points using

$$S_{11}(F) = 20\log_{10}\frac{|fE_z^{\text{coax}}(F) - fE_z^{\text{point}}(F)|}{|fE_z^{\text{coax}}(F)|}\ [\text{dB}], \tag{59}$$

where $fE_z(F) = \int_0^T E_z(t)\exp(-2\pi i F t)\,dt$ is computed for a set of frequencies $3\,\text{GHz} \le F \le 6\,\text{GHz}$. Our third-order single point result is closer to the curve

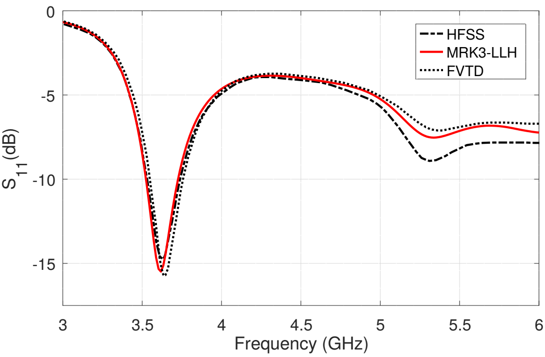


**Figure 6.** Return loss DRA $S_{11}$ coefficient.

| scheme | # of $\Delta t_g$ | $n(\boldsymbol{LU})$ | $\frac{n(\boldsymbol{LU})_{\text{RK3}}}{n(\boldsymbol{LU})}$ | CPU [s] | $\frac{\text{CPU}_{\text{RK3}}}{\text{CPU}}$ | $\frac{\|S_{11}-S_{11}^{\text{RK3}}\|_2}{\|S_{11}^{\text{RK3}}\|_2}$ |
|---|---|---|---|---|---|---|
| RK3 | 170 847 | 97 792 310 259 | 1 | 65 301 | 1 | 0 |
| MRK3-TW-P2 | 2 670 | 26 009 562 030 | 3.76 | 18 965 | 3.44 | $1.48\times10^{-2}$ |
| MRK3-TW-OP | 1 810 | 24 000 306 780 | 4.07 | 18 551 | 3.52 | $2.01\times10^{-2}$ |
| MRK3-LLH-P2 | 2 670 | 24 471 455 130 | 4.00 | 17 150 | 3.8 | $1.29\times10^{-5}$ |
| MRK3-LLH-OP | 2 171 | 21 515 239 590 | 4.55 | 16 834 | 3.88 | $1.97\times10^{-5}$ |

**Table 8.** DRA: performance of MRK3 schemes compared to single-rate RK3, where $n(\boldsymbol{LU})$ is the number of flux operations. The last column shows the relative difference of $S_{11}^{\text{RK3}}$ computed with Runge–Kutta to $S_{11}$ obtained from MRK3 schemes.

obtained with HFSS (Figure 6). Higher-order schemes conduct more high-frequency oscillations, which is visible for higher than 5 GHz reflections. Performance evaluation of our implementation is shown in Table 8. The simulation speedup achieved in our experiments using a third-order scheme is similar to the one reported in [15] for the same problem but using a different mesh and second-order scheme.

## 6. Summary

In this paper two multirate schemes with SSP RK3 base method are tested in application to Maxwell's equations on unstructured tetrahedral meshes. The order conditions for MPRK schemes on linear problems show that the third-order extension of the scheme proposed in [38] has only first-order accurate coupling, while the scheme developed in [27] is third-order accurate for linear problems with three-stage third-order Runge–Kutta methods. For 3D simulations, both schemes are flexible in terms of local time-step partition allowing higher speedup than previously reported in the literature even for more expensive third-order approximation. Solution error comparisons confirm that the analysis based on order conditions is valid in 3D simulations. Moreover, our numerical results show that arbitrary time-step ratio does not compromise the accuracy of simulations. Future work may include extending the implementation of multirate schemes to higher order for 3D simulations.

## Acknowledgments

Kotovshchikova was supported by an NSERC postgraduate scholarship, while Lui was supported by an NSERC Discovery grant.

## References

[1] P. Albrecht, *The Runge–Kutta theory in a nutshell*, SIAM J. Numer. Anal. **33** (1996), no. 5, 1712–1735. MR Zbl

[2] J. F. Andrus, *Numerical solution of systems of ordinary differential equations separated into subsystems*, SIAM J. Numer. Anal. **16** (1979), no. 4, 605–611. MR Zbl

[3] ———, *Stability of a multi-rate method for numerical integration of ODEs*, Comput. Math. Appl. **25** (1993), no. 2, 3–14. MR Zbl

[4] L. D. Angulo, J. Alvarez, F. L. Teixeira, M. F. Pantoja, and S. G. Garcia, *Causal-path local time-stepping in the discontinuous Galerkin method for Maxwell's equations*, J. Comput. Phys. **256** (2014), 678–695. MR Zbl

[5] C. A. Balanis, *Advanced engineering electromagnetics*, Wiley, 1989.

[6] D. Baumann, *A 3-d numerical field solver based on the finite-volume time-domain method*, Ph.D. thesis, ETH Zürich, 2006.

[7] P. Bonnet, X. Ferrieres, F. Issac, F. Paladian, J. Grando, J. C. Alliot, and J. Fontaine, *Numerical modeling of scattering problems using a time domain finite volume method*, J. Electromagnet. Wave. **11** (1997), no. 8, 1165–1189.

[8] F. Collino, T. Fouquet, and P. Joly, *A conservative space-time mesh refinement method for the 1-D wave equation, I: Construction*, Numer. Math. **95** (2003), no. 2, 197–221. MR Zbl

[9] ———, *Conservative space-time mesh refinement methods for the FDTD solution of Maxwell's equations*, J. Comput. Phys. **211** (2006), no. 1, 9–35. MR Zbl

[10] E. M. Constantinescu and A. Sandu, *Multirate timestepping methods for hyperbolic conservation laws*, J. Sci. Comput. **33** (2007), no. 3, 239–278. MR Zbl

[11] C. Dawson and R. Kirby, *High resolution schemes for conservation laws with locally varying time steps*, SIAM J. Sci. Comput. **22** (2000), no. 6, 2256–2281. MR Zbl

[12] S. Descombes, S. Lanteri, and L. Moya, *Locally implicit time integration strategies in a discontinuous Galerkin method for Maxwell's equations*, J. Sci. Comput. **56** (2013), no. 1, 190–218. MR Zbl

[13] J. Diaz and M. J. Grote, *Energy conserving explicit local time stepping for second-order wave equations*, SIAM J. Sci. Comput. **31** (2009), no. 3, 1985–2014. MR Zbl

[14] A. Ezziani and P. Joly, *Local time stepping and discontinuous Galerkin methods for symmetric first order hyperbolic systems*, J. Comput. Appl. Math. **234** (2010), no. 6, 1886–1895. MR Zbl

[15] C. Fumeaux, D. Baumann, P. Leuchtmann, and R. Vahldieck, *A generalized local time-step scheme for efficient FVTD simulations in strongly inhomogeneous meshes*, IEEE T. Microw. Theory **52** (2004), no. 3, 1067–1076.

[16] C. W. Gear and D. R. Wells, *Multirate linear multistep methods*, BIT **24** (1984), no. 4, 484–502. MR Zbl

[17] N. Goedel, S. Schomann, T. Warburton, and M. Clemens, *Local timestepping discontinuous Galerkin methods for electromagnetic RF field problems*, 3rd European Conference on Antennas and Propagation, IEEE, 2009, pp. 2149–2153.

[18] M. J. Grote and T. Mitkova, *Explicit local time-stepping methods for Maxwell's equations*, J. Comput. Appl. Math. **234** (2010), no. 12, 3283–3302. MR Zbl

[19] ———, *High-order explicit local time-stepping methods for damped wave equations*, J. Comput. Appl. Math. **239** (2013), 270–289. MR Zbl

[20] M. Günther, A. Kværnø, and P. Rentrop, *Multirate partitioned Runge–Kutta methods*, BIT **41** (2001), no. 3, 504–514. MR Zbl

[21] E. Hairer, *Order conditions for numerical methods for partitioned ordinary differential equations*, Numer. Math. **36** (1981), no. 4, 431–445. MR Zbl

[22] R. F. Harrington, *Time-harmonic electromagnetic fields*, McGraw-Hill, 1961.

[23] W. Hundsdorfer, A. Mozartova, and V. Savcenco, *Monotonicity conditions for multirate and partitioned explicit Runge–Kutta schemes*, Recent developments in the numerics of nonlinear hyperbolic conservation laws (R. Ansorge, H. Bijl, A. Meister, and T. Sonar, eds.), Notes Numer. Fluid Mech. Multidiscip. Des., no. 120, Springer, 2013, pp. 177–195. MR Zbl

[24] Z. a. Jackiewicz and R. Vermiglio, *Order conditions for partitioned Runge–Kutta methods*, Appl. Math. **45** (2000), no. 4, 301–316. MR Zbl

[25] G.-S. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. **126** (1996), no. 1, 202–228. MR Zbl

[26] M. Kotovshchikova, D. K. Firsov, and S. H. Lui, *A third order finite volume WENO scheme for Maxwell's equations on tetrahedral meshes*, Commun. Appl. Math. Comput. Sci. **13** (2018), no. 1, 87–106. MR Zbl

[27] L. Liu, X. Li, and F. Q. Hu, *Nonuniform time-step Runge–Kutta discontinuous Galerkin method for computational aeroacoustics*, J. Comput. Phys. **229** (2010), no. 19, 6874–6897. MR Zbl

[28] N. Manzanares-Filho, C. A. A. Moino, and A. B. Jorge, *An improved controlled random search algorithm for inverse airfoil cascade design*, 6th World Congress on Structural and Multidisciplinary Optimization (J. Herskovits, S. Mazorche, and A. Canelas, eds.), COPPE, 2005.

[29] E. Montseny, S. Pernet, X. Ferriéres, and G. Cohen, *Dissipative terms and local time-stepping improvements in a spatial high order discontinuous Galerkin scheme for the time-domain Maxwell's equations*, J. Comput. Phys. **227** (2008), no. 14, 6795–6820. MR Zbl

[30] S. Osher and R. Sanders, *Numerical approximations to nonlinear conservation laws with locally varying time and space grids*, Math. Comp. **41** (1983), no. 164, 321–336. MR Zbl

[31] S. Piperno, *Symplectic local time-stepping in non-dissipative DGTD methods applied to wave propagation problems*, M2AN Math. Model. Numer. Anal. **40** (2006), no. 5, 815–841. MR Zbl

[32] J. R. Rice, *Split Runge–Kutta method for simultaneous equations*, J. Res. Nat. Bur. Standards **64B** (1960), 151–170. MR Zbl

[33] A. Sandu and E. M. Constantinescu, *Multirate explicit Adams methods for time integration of conservation laws*, J. Sci. Comput. **38** (2009), no. 2, 229–249. MR Zbl

[34] M. Schlegel, O. Knoth, M. Arnold, and R. Wolke, *Multirate Runge–Kutta schemes for advection equations*, J. Comput. Appl. Math. **226** (2009), no. 2, 345–357. MR Zbl

[35] B. Seny, J. Lambrechts, R. Comblen, V. Legat, and J.-F. Remacle, *Multirate time stepping for accelerating explicit discontinuous Galerkin computations with application to geophysical flows*, Internat. J. Numer. Methods Fluids **71** (2013), no. 1, 41–64. MR Zbl

[36] G. R. Shubin and J. B. Bell, *A modified equation approach to constructing fourth-order methods for acoustic wave propagation*, SIAM J. Sci. Statist. Comput. **8** (1987), no. 2, 135–151. MR Zbl

[37] J. L. Steger and R. F. Warming, *Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods*, J. Comput. Phys. **40** (1981), no. 2, 263–293. MR Zbl

[38] H.-z. Tang and G. Warnecke, *High resolution schemes for conservation laws and convection-diffusion equations with varying time and space grids*, J. Comput. Math. **24** (2006), no. 2, 121–140. MR Zbl

[39] A. Taube, M. Dumbser, C.-D. Munz, and R. Schneider, *A high-order discontinuous Galerkin method with time-accurate local time stepping for the Maxwell equations*, Int. J. Numer. Model. El. **22** (2009), no. 1, 77–103. Zbl

[40] Y.-T. Zhang and C.-W. Shu, *Third order WENO scheme on three dimensional tetrahedral meshes*, Commun. Comput. Phys. **5** (2009), no. 2–4, 836–848. MR Zbl

Received October 28, 2019. Revised February 25, 2020.

MARINA KOTOVSHCHIKOVA: m.a.kotovshchikova@gmail.com
*San Jose, CA United States*

DMITRY K. FIRSOV: d.k.firsov@gmail.com
*San Jose, CA, United States*

SHIU HONG LUI: luish@cc.umanitoba.ca
*Department of Mathematics, University of Manitoba, Winnipeg, MB, Canada*

COMM. APP. MATH. AND COMP. SCI.
Vol. 15, No. 1, 2020
dx.doi.org/10.2140/camcos.2020.15.89

# FAST OPTICAL ABSORPTION SPECTRA CALCULATIONS FOR PERIODIC SOLID STATE SYSTEMS

FELIX HENNEKE, LIN LIN, CHRISTIAN VORWERK, CLAUDIA DRAXL, RUPERT KLEIN AND CHAO YANG

We present a method to construct an efficient approximation to the bare exchange and screened direct interaction kernels of the Bethe–Salpeter Hamiltonian for periodic solid state systems via the interpolative separable density fitting technique. We show that the cost of constructing the approximate Bethe–Salpeter Hamiltonian can be reduced to nearly optimal as $\mathcal{O}(N_k)$ with respect to the number of samples in the Brillouin zone $N_k$ for the first time. In addition, we show that the cost for applying the Bethe–Salpeter Hamiltonian to a vector scales as $\mathcal{O}(N_k \log N_k)$. Therefore, the optical absorption spectrum, as well as selected excitation energies, can be efficiently computed via iterative methods such as the Lanczos method. This is a significant reduction from the $\mathcal{O}(N_k^2)$ and $\mathcal{O}(N_k^3)$ scaling associated with a brute force approach for constructing the Hamiltonian and diagonalizing the Hamiltonian, respectively. We demonstrate the efficiency and accuracy of this approach with both one-dimensional model problems and three-dimensional real materials (graphene and diamond). For the diamond system with $N_k = 2197$, it takes 6 hours to assemble the Bethe–Salpeter Hamiltonian and 4 hours to fully diagonalize the Hamiltonian using 169 cores when the brute force approach is used. The new method takes less than 3 minutes to set up the Hamiltonian and 24 minutes to compute the absorption spectrum on a single core.

## 1. Introduction

The Bethe–Salpeter equation (BSE), derived from the many-body perturbation theory (MBPT), is a widely used method for describing the optical absorption process in molecules and solids [32; 33; 36; 24; 1; 25; 7]. It models the behavior of an electron–hole pair, which is an excitation process with two quasiparticles. Solving the BSE requires constructing and diagonalizing a structured matrix, called the Bethe–Salpeter Hamiltonian (BSH). In the context of optical absorption, the

*MSC2010:* 65F15, 65Z05.
*Keywords:* Bethe–Salpeter equation, interpolative separable density fitting, optical absorption function.

eigenvalues of the BSH are the exciton energies and the corresponding eigenfunctions yield the exciton wavefunctions. The BSH consists of the so-called bare exchange and screened direct interaction kernels that depend on single particle orbitals obtained from a quasiparticle (usually at the GW level) or mean-field calculation. For isolated systems such as molecules, the construction of these kernels requires at least $\mathbb{O}(N_e^5)$ operations in a conventional approach, where $N_e$ is the number of electrons in the system. This is very costly for large systems that contain hundreds or more atoms. Recent efforts have actively explored methods for efficient representation of the BSH, in order to reduce the high computational cost of BSE calculations [4; 3; 16; 21; 30; 27; 28; 31; 23].

In a recent work [13], two of the authors have presented an efficient way to construct the BSH for molecular systems, and to efficiently solve the BSE eigenvalue problem using an iterative scheme. This approach is based on the recently developed interpolative separable density fitting (ISDF) decomposition [19; 20]. The ISDF decomposition has been applied to accelerate a number of applications in computational chemistry and materials science, including the computation of two-electron integrals [19], correlation energy in the random phase approximation [18], density functional perturbation theory [15], and hybrid density functional calculations [12; 8]. In this scheme, a matrix consisting of products of single particle orbital pairs is efficiently approximated as a low-rank matrix product of a matrix built with a small number of auxiliary basis vectors and an expansion coefficient matrix. This decomposition allows us to construct efficient representations of the bare exchange and screened direct kernels. For isolated molecular systems, the construction of the ISDF-compressed BSH matrix only requires $\mathbb{O}(N_e^3)$ operations when the rank of the numerical auxiliary basis is kept at $\mathbb{O}(N_e)$. This results in considerable reduction of the cost compared to the $\mathbb{O}(N_e^5)$ complexity required in a conventional approach. By keeping the interaction kernels in a decomposed form, the matrix–vector multiplications required in the iterative diagonalization procedures of the Hamiltonian $H_{\mathrm{BSE}}$ can be performed efficiently. We can further use these efficient matrix–vector multiplications in a structure-preserving Lanczos algorithm [34] to obtain an approximate absorption spectrum without an explicit diagonalization of the approximate $H_{\mathrm{BSE}}$.

This paper generalizes the work in [13] to periodic solid state systems. According to the Bloch decomposition, each single particle orbital in a periodic system can be characterized by an orbital index $i$ and a Brillouin zone index $\boldsymbol{k}$. Compared to isolated systems, the total number of electrons $N_e$ is equal to the number of electrons per unit cell multiplied by the number of $\boldsymbol{k}$-points denoted by $N_k$. It has been observed that for many extended systems, the number of orbitals (both occupied and virtual orbitals) required for one particular $\boldsymbol{k}$ index can be relatively small, and is independent of $N_e$. Hence, the difficulty of optical absorption spectra calculations for periodic systems mainly arise from the large number of $\boldsymbol{k}$-points. This is particularly the case when the excitons are delocalized in the real space, or when the Fermi-surface is not smooth (such as graphene, and other metallic systems). In such case, $N_k$ can often be rather large (from hundreds to hundreds of thousands; see, e.g., [29], where a $120 \times 120 \times 1$ $\boldsymbol{k}$-grid is used for the quasi-two-dimensional $\mathrm{MoS}_2$ system) in order to properly discretize and sample the Brillouin zone. The cost for constructing the bare exchange and screened direct kernels scales as $\mathcal{O}(N_k^2)$, while the cost for diagonalizing the corresponding BSH scales as $\mathcal{O}(N_k^3)$. This is prohibitively expensive when a dense discretization of the Brillouin zone is needed.

With the help of ISDF for periodic systems [20], we reduce the computational cost for producing optical absorption spectra to a scaling almost linear in $N_k$. First, the complexity of the bare exchange and screened direct kernel construction for extended systems is reduced to the optimal complexity of $\mathcal{O}(N_k)$. A sufficiently reduced representation of the pair product orbitals is possible, thanks to the smoothness of the single particle orbitals with respect to the $\boldsymbol{k}$ index, and the fact that the Brillouin zone is a compact domain. Second, the separable structure of the decomposition makes it possible to exploit a convolutional structure in the screened direct kernel. The complexity of applying the approximated kernels to a vector with respect to $N_k$ is thus only $\mathcal{O}(N_k \log N_k)$. Instead of diagonalizing the BSH directly, we use iterative methods such as the Lanczos method to evaluate the optical absorption spectrum. The same strategy can be applied to evaluate selected excitation energies.

Despite the increasingly wide adoption of the BSE theory in condensed matter physics and quantum chemistry for analyzing optical properties of materials, we could not find a precise mathematical description of how the BSH is constructed for periodic systems in the literature. Therefore, after concise review of the single particle theory and the Bethe–Salpeter equation for periodic systems in Section 2.1, we provide a relatively self-contained derivation of the BSE for periodic systems in Section 2.2 from a numerical linear algebra perspective. We hope our presentation (especially using a discretized Brillouin zone so that all matrices are of finite dimension) is useful to readers not familiar with the matter.

Then the rest of the paper is organized as follows. The interpolative separable density fitting for periodic systems is introduced in Section 3, and the application of the approximate BSH in the ISDF format to a vector in Section 4. The numerical results are presented in Section 5, followed by a conclusion in Section 6.

## 2. Preliminaries

**2.1.** ***Single particle theory for periodic systems.*** To facilitate further discussion we briefly review Bloch–Floquet theory for periodic systems. Without loss of generality we consider a three-dimensional crystal. The *Bravais lattice* with lattice vectors $\boldsymbol{a}_1, \boldsymbol{a}_2, \boldsymbol{a}_3 \in \mathbb{R}^3$ is defined as

$$\mathbb{L} = \{\boldsymbol{R} \mid \boldsymbol{R} = n_1\boldsymbol{a}_1 + n_2\boldsymbol{a}_2 + n_3\boldsymbol{a}_3,\ n_1, n_2, n_3 \in \mathbb{Z}\}. \tag{2-1}$$

In single particle theories such as the Kohn–Sham density functional theory, the self-consistent effective potential $V_{\text{eff}}$ is real-valued and $\mathbb{L}$-periodic, i.e.,

$$V_{\text{eff}}(\boldsymbol{r} + \boldsymbol{R}) = V_{\text{eff}}(\boldsymbol{r}) \quad \text{for all } \boldsymbol{r} \in \mathbb{R}^3 \text{ and } \boldsymbol{R} \in \mathbb{L}.$$

The unit cell is defined as

$$\Omega = \{\boldsymbol{r} = c_1\boldsymbol{a}_1 + c_2\boldsymbol{a}_2 + c_3\boldsymbol{a}_3 \mid 0 \le c_1, c_2, c_3 < 1\}. \tag{2-2}$$

The Bravais lattice induces a reciprocal lattice $\mathbb{L}^*$, with its lattice vectors $\boldsymbol{b}_1, \boldsymbol{b}_2, \boldsymbol{b}_3$ satisfying $\boldsymbol{a}_\alpha \cdot \boldsymbol{b}_\beta = 2\pi\delta_{\alpha\beta}$, $\alpha, \beta \in \{1, 2, 3\}$. The unit cell of the reciprocal lattice is called the (first) Brillouin zone and denoted by $\Omega^*$, defined as

$$\Omega^* = \left\{\boldsymbol{k} = k_1\boldsymbol{b}_1 + k_2\boldsymbol{b}_2 + k_3\boldsymbol{b}_3 \;\middle|\; -\tfrac{1}{2} \le k_1, k_2, k_3 < \tfrac{1}{2}\right\}.$$

The Brillouin zone has a number of special points related to the symmetry of the crystal. The common special point is the $\Gamma$-point, which corresponds to $\boldsymbol{k} = [0, 0, 0]^\top$.

According to the Bloch–Floquet theory, the spectrum of the Hamiltonian $\mathcal{H} = -\frac{1}{2}\nabla_{\boldsymbol{r}}^2 + V_{\text{eff}}(\boldsymbol{r})$ can be relabeled using two indices $(i, \boldsymbol{k})$, where $i \in \mathbb{N}$ is called the band index and $\boldsymbol{k} \in \Omega^*$ is the Brillouin zone index. Each generalized eigenfunction $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ is known as a Bloch orbital and satisfies $\mathcal{H}\psi_{i\boldsymbol{k}}(\boldsymbol{r}) = \epsilon_{i\boldsymbol{k}}\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ with Bloch boundary conditions $\psi_{i\boldsymbol{k}}(\boldsymbol{r} + \boldsymbol{R}) = e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{R}}\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ for any $\boldsymbol{R} \in \mathbb{L}$. Furthermore, $\psi_{i\boldsymbol{k}}$ can be decomposed using the Bloch decomposition

$$\psi_{i\boldsymbol{k}}(\boldsymbol{r}) = e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}u_{i\boldsymbol{k}}(\boldsymbol{r}), \tag{2-3}$$

where $u_{i\boldsymbol{k}}(\boldsymbol{r})$ is the periodic part of $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ satisfying the periodic boundary condition on the unit cell

$$u_{i\boldsymbol{k}}(\boldsymbol{r} + \boldsymbol{R}) = u_{i\boldsymbol{k}}(\boldsymbol{r}) \quad \text{for all } \boldsymbol{R} \in \mathbb{L}. \tag{2-4}$$

It can be directly obtained by solving the eigenvalue problem

$$\mathcal{H}(\boldsymbol{k})u_{i\boldsymbol{k}} = \epsilon_{i\boldsymbol{k}}u_{i\boldsymbol{k}}(\boldsymbol{r}), \quad \boldsymbol{r} \in \Omega,\ \boldsymbol{k} \in \Omega^*, \tag{2-5}$$

where $\mathcal{H}(\boldsymbol{k}) = -\frac{1}{2}(\nabla_{\boldsymbol{r}} + \mathrm{i}\boldsymbol{k})^2 + V_{\text{eff}}(\boldsymbol{r})$. For each $\boldsymbol{k} \in \Omega^*$, the eigenvalues $\epsilon_{i\boldsymbol{k}}$ are ordered nondecreasingly. For a fixed $i$, $\{\epsilon_{i\boldsymbol{k}}\}$ as a function of $\boldsymbol{k}$ is called a *Bloch band*. The collection of all eigenvalues forms the *band structure* of the crystal, which characterizes the spectrum of the operator $\mathcal{H}$.

In the discussion below, we denote by $N_v$ the number of valence bands (i.e., occupied orbitals per unit cell in the ground state) and $N_c$ the number of conduction bands (i.e., unoccupied orbitals per unit cell in the ground state). We also define

$N = N_v + N_c$. We assume the systems to be insulating, in the sense that the following band isolation conditions between the valence and conduction bands are satisfied:

$$\inf|\epsilon_{i\boldsymbol{k}} - \epsilon_{i'\boldsymbol{k}'}| := \epsilon_g > 0, \quad \boldsymbol{k}, \boldsymbol{k}' \in \Omega^*,\ 1 \le i \le N_v,\ N_v + 1 \le i' \le N. \tag{2-6}$$

Denote by $|\Omega|$ the volume of the unit cell, and by

$$|\Omega^*| = \frac{(2\pi)^3}{|\Omega|}$$

the volume of the Brillouin zone. The Bloch orbitals $\{\psi_{i\boldsymbol{k}}\}$ satisfy the orthonormality condition in the distributional sense:

$$\int_{\mathbb{R}^3} \psi^*_{i'\boldsymbol{k}'}(\boldsymbol{r})\psi_{i,\boldsymbol{k}}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = |\Omega^*|\delta_{i',i}\delta(\boldsymbol{k}' - \boldsymbol{k}). \tag{2-7}$$

Here $\delta_{i',i}$ is the Kronecker $\delta$ symbol for a discrete set, while $\delta(\boldsymbol{k}' - \boldsymbol{k})$ is the Dirac delta distribution. Equation (2-7) implies the normalization condition when integrated over the Brillouin zone:

$$\frac{1}{|\Omega^*|}\int_{\Omega^*}\int_{\mathbb{R}^3} \psi^*_{i'\boldsymbol{k}}(\boldsymbol{r})\psi_{i\boldsymbol{k}}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{k} = \delta_{i',i}. \tag{2-8}$$

From the Bloch orbitals, the ground state electron density can be constructed as

$$\rho(\boldsymbol{r}) = \frac{1}{|\Omega^*|}\int_{\Omega^*}\sum_{i=1}^{N_v}|\psi_{i\boldsymbol{k}}(\boldsymbol{r})|^2\,\mathrm{d}\boldsymbol{k} = \frac{1}{|\Omega^*|}\int_{\Omega^*}\sum_{i=1}^{N_v}|u_{i\boldsymbol{k}}(\boldsymbol{r})|^2\,\mathrm{d}\boldsymbol{k}. \tag{2-9}$$

In order to practically perform calculations for periodic systems, the integration with respect to the Brillouin zone $\Omega^*$ needs to be discretized using a quadrature. The most commonly used scheme is based on the Monkhorst–Pack grid [22]

$$\mathcal{K}^\ell_{\boldsymbol{s}} = \left\{\sum_{\alpha=1}^{3}\frac{m_\alpha - s_\alpha}{N^\ell_\alpha}\boldsymbol{b}_\alpha \,\middle|\, m_\alpha = -\frac{N^\ell_\alpha}{2}+1, \dots, \frac{N^\ell_\alpha}{2},\ 0 \le s_\alpha < 1,\ \alpha = 1, 2, 3\right\}. \tag{2-10}$$

It is clear that $\mathcal{K}^\ell_{\boldsymbol{s}} \subset \Omega^*$ and that it corresponds to a uniform discretization of the Brillouin zone. When the shift vector $\boldsymbol{s} = \boldsymbol{0}$, we denote $\mathcal{K}^\ell := \mathcal{K}^\ell_{\boldsymbol{0}}$, and the calculation of periodic systems can be *equivalently* performed using a supercell consisting of $N^\ell_1 \times N^\ell_2 \times N^\ell_3$ unit cells. The supercell is denoted by $\Omega^\ell$, and is further equipped with a periodic boundary condition called the Born–von Karman boundary condition [2]. The calculation of a periodic crystal can thus be recovered by taking the limit $N^\ell_\alpha \to \infty$. We denote by $N_k \equiv N^\ell := N^\ell_1 N^\ell_2 N^\ell_3$ the total number of unit cells, or equivalently the total number of Monkhorst–Pack grid points in the Brillouin zone.

Assuming the Brillouin zone is discretized using $\mathcal{K}^\ell$, the orthogonality condition (2-7) becomes

$$\int_{\Omega^\ell} \psi^*_{i'\boldsymbol{k}'}(\boldsymbol{r})\psi_{i\boldsymbol{k}}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = \delta_{i',i}\delta_{\boldsymbol{k}',\boldsymbol{k}}, \quad \boldsymbol{k},\boldsymbol{k}' \in \mathcal{K}^\ell. \tag{2-11}$$

We also modify the Bloch decomposition as

$$\psi_{i\boldsymbol{k}}(\boldsymbol{r}) = \frac{1}{\sqrt{N^\ell}} e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}} u_{i\boldsymbol{k}}(\boldsymbol{r}), \quad \boldsymbol{k} \in \mathcal{K}^\ell. \tag{2-12}$$

Here the normalization factor $1/\sqrt{N^\ell}$ is introduced so that the orthogonality condition for the periodic part implies

$$\int_{\Omega} u^*_{i'\boldsymbol{k}}(\boldsymbol{r})u_{i\boldsymbol{k}}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = \delta_{i',i}, \quad \boldsymbol{k} \in \mathcal{K}^\ell. \tag{2-13}$$

To facilitate the bookkeeping effort of various relevant constants in practical calculations, in the discussion below we will always assume that the Brillouin zone is discretized into $\mathcal{K}^\ell$ with a corresponding supercell $\Omega^\ell$. The volume of the supercell is $|\Omega^\ell| = N^\ell|\Omega| = N_k|\Omega|$. The unit cell is further discretized into a uniform grid $\{\boldsymbol{r}_i\}_{i=1}^{N_g}$. Practical BSE calculations often truncate the number of conduction bands aggressively, in the sense that $N_g \gg N_v + N_c =: N$. Numerical results indicate that in many cases, the low-lying excitation spectrum is relatively insensitive to $N_c$, and one can often choose $N_c \approx N_v$. Unless otherwise clarified, we may not distinguish a continuous vector $u(\boldsymbol{r})$ and the corresponding discretized vector $\{u(\boldsymbol{r}_i)\}$. Similarly, when the context is clear, we do not distinguish the kernel of an operator $A(\boldsymbol{r},\boldsymbol{r}')$ and its discretized matrix $\{A(\boldsymbol{r}_i,\boldsymbol{r}_j)\}$.

**2.2.** ***Bethe–Salpeter equation for periodic systems.*** The Bethe–Salpeter equation is an eigenvalue problem of the form

$$H_{\mathrm{BSE}}X = EX, \tag{2-14}$$

where $H_{\mathrm{BSE}}$ is the Bethe–Salpeter Hamiltonian (BSH), $X$ is the exciton wavefunction, and $E$ is the corresponding exciton energy. For periodic systems, the BSH has the block structure

$$H_{\mathrm{BSE}} = \begin{bmatrix} D + 2V_A - W_A & 2V_B - W_B \\ -2\overline{V}_B + \overline{W}_B & -D - 2\overline{V}_A + \overline{W}_A \end{bmatrix}, \tag{2-15}$$

where $D(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') = (\epsilon_{i_c\boldsymbol{k}} - \epsilon_{i_v\boldsymbol{k}})\delta_{i_v,j_v}\delta_{i_c,j_c}\delta_{\boldsymbol{k},\boldsymbol{k}'}$ is an $(N_vN_cN_k) \times (N_vN_cN_k)$ diagonal matrix. The quasiparticle energies $\epsilon_{i_v\boldsymbol{k}}, \epsilon_{i_c\boldsymbol{k}}$ are typically obtained from a GW calculation [32]. The $V_A$ and $V_B$ matrices represent the bare *exchange* interaction of electron–hole pairs, and the $W_A$ and $W_B$ matrices are referred to as the screened *direct* interaction of electron–hole pairs. These matrices are defined as

$$\begin{aligned} V_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \int_{\Omega^\ell\times\Omega^\ell} \overline{\psi}_{i_c\boldsymbol{k}}(\boldsymbol{r})\psi_{i_v\boldsymbol{k}}(\boldsymbol{r})V(\boldsymbol{r},\boldsymbol{r}')\overline{\psi}_{j_v\boldsymbol{k}'}(\boldsymbol{r}')\psi_{j_c\boldsymbol{k}'}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \\ V_B(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \int_{\Omega^\ell\times\Omega^\ell} \overline{\psi}_{i_c\boldsymbol{k}}(\boldsymbol{r})\psi_{i_v\boldsymbol{k}}(\boldsymbol{r})V(\boldsymbol{r},\boldsymbol{r}')\overline{\psi}_{j_c\boldsymbol{k}'}(\boldsymbol{r}')\psi_{j_v\boldsymbol{k}'}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \\ W_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \int_{\Omega^\ell\times\Omega^\ell} \overline{\psi}_{i_c\boldsymbol{k}}(\boldsymbol{r})\psi_{j_c\boldsymbol{k}'}(\boldsymbol{r})W(\boldsymbol{r},\boldsymbol{r}')\overline{\psi}_{j_v\boldsymbol{k}'}(\boldsymbol{r}')\psi_{i_v\boldsymbol{k}}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \\ W_B(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \int_{\Omega^\ell\times\Omega^\ell} \overline{\psi}_{i_c\boldsymbol{k}}(\boldsymbol{r})\psi_{j_v\boldsymbol{k}'}(\boldsymbol{r})W(\boldsymbol{r},\boldsymbol{r}')\overline{\psi}_{j_c\boldsymbol{k}'}(\boldsymbol{r}')\psi_{i_v\boldsymbol{k}}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}'. \end{aligned} \tag{2-16}$$

Here $\psi_{i_v\boldsymbol{k}}$ and $\psi_{i_c\boldsymbol{k}}$ are the valence and conduction single particle orbitals typically obtained from a Kohn–Sham density functional theory (KSDFT) calculation, respectively, and $V(\boldsymbol{r},\boldsymbol{r}')$ and $W(\boldsymbol{r},\boldsymbol{r}')$ are the bare and screened Coulomb interactions. Both $V_A$ and $W_A$ are Hermitian, whereas $V_B$ and $W_B$ are complex symmetric. Within the so-called Tamm–Dancoff approximation (TDA) [25], both $V_B$ and $W_B$ are neglected in (2-15). In this case, the $H_{\mathrm{BSE}}$ becomes Hermitian and we can focus on computing the upper left block of $H_{\mathrm{BSE}}$. Both the KSDFT and GW calculations can be challenging in their own right. In this work, however, we consider their output as given and the starting point of our BSE calculation.

In the following discussion, when a single index $i$ is used, it refers to either $i_v$ or $i_c$. Using the Bloch decomposition (2-12), the matrix elements of the BSH can be written using the periodic part of the orbitals as

$$\begin{aligned} V_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k^2}\int_{\Omega^\ell\times\Omega^\ell} \bar{u}_{i_c\boldsymbol{k}}(\boldsymbol{r})u_{i_v\boldsymbol{k}}(\boldsymbol{r})V(\boldsymbol{r},\boldsymbol{r}')\bar{u}_{j_v\boldsymbol{k}'}(\boldsymbol{r}')u_{j_c\boldsymbol{k}'}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \\ V_B(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k^2}\int_{\Omega^\ell\times\Omega^\ell} \bar{u}_{i_c\boldsymbol{k}}(\boldsymbol{r})u_{i_v\boldsymbol{k}}(\boldsymbol{r})V(\boldsymbol{r},\boldsymbol{r}')\bar{u}_{j_c\boldsymbol{k}'}(\boldsymbol{r}')u_{j_v\boldsymbol{k}'}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \\ W_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k^2}\int_{\Omega^\ell\times\Omega^\ell} e^{-\mathrm{i}(\boldsymbol{k}-\boldsymbol{k}')\cdot(\boldsymbol{r}-\boldsymbol{r}')}\bar{u}_{i_c\boldsymbol{k}}(\boldsymbol{r})u_{j_c\boldsymbol{k}'}(\boldsymbol{r}) \\ &\qquad\qquad \times W(\boldsymbol{r},\boldsymbol{r}')\bar{u}_{j_v\boldsymbol{k}'}(\boldsymbol{r}')u_{i_v\boldsymbol{k}}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \\ W_B(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k^2}\int_{\Omega^\ell\times\Omega^\ell} e^{-\mathrm{i}(\boldsymbol{k}-\boldsymbol{k}')\cdot(\boldsymbol{r}-\boldsymbol{r}')}\bar{u}_{i_c\boldsymbol{k}}(\boldsymbol{r})u_{j_v\boldsymbol{k}'}(\boldsymbol{r}) \\ &\qquad\qquad \times W(\boldsymbol{r},\boldsymbol{r}')\bar{u}_{j_c\boldsymbol{k}'}(\boldsymbol{r}')u_{i_v\boldsymbol{k}}(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}'. \end{aligned} \tag{2-17}$$

Note that $V_A$, $V_B$ in (2-17) do not involve the phase factors, since the factor $e^{\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{r}}$ cancels exactly due to the complex conjugate operation. The phase factor only appears in the $W_A$, $W_B$ terms.

Equation (2-17) requires the evaluation of integrals of the form

$$\mathcal{V}(f,g) := \frac{1}{N_k}\int_{\Omega^\ell\times\Omega^\ell} \bar{f}(\boldsymbol{r})V(\boldsymbol{r},\boldsymbol{r}')g(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}', \tag{2-18}$$

and

$$\mathcal{W}_{\boldsymbol{q}}(f,g) := \frac{1}{N_k}\int_{\Omega^\ell\times\Omega^\ell} e^{-\mathrm{i}\boldsymbol{q}\cdot(\boldsymbol{r}-\boldsymbol{r}')}\,\bar{f}(\boldsymbol{r})W(\boldsymbol{r},\boldsymbol{r}')g(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}'. \tag{2-19}$$

Using such notation,

$$\begin{aligned}
V_A(i_v i_c \boldsymbol{k},\, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k}\mathcal{V}(\bar{u}_{i_v\boldsymbol{k}}u_{i_c\boldsymbol{k}},\, \bar{u}_{j_v\boldsymbol{k}'}u_{j_c\boldsymbol{k}'}),\\
V_B(i_v i_c \boldsymbol{k},\, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k}\mathcal{V}(\bar{u}_{i_v\boldsymbol{k}}u_{i_c\boldsymbol{k}},\, \bar{u}_{j_c\boldsymbol{k}'}u_{j_v\boldsymbol{k}'}),\\
W_A(i_v i_c \boldsymbol{k},\, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k}\mathcal{W}_{\boldsymbol{k}-\boldsymbol{k}'}(\bar{u}_{j_c\boldsymbol{k}'}u_{i_c\boldsymbol{k}},\, \bar{u}_{j_v\boldsymbol{k}'}u_{i_v\boldsymbol{k}}),\\
W_B(i_v i_c \boldsymbol{k},\, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k}\mathcal{W}_{\boldsymbol{k}-\boldsymbol{k}'}(\bar{u}_{j_v\boldsymbol{k}'}u_{i_c\boldsymbol{k}},\, \bar{u}_{j_c\boldsymbol{k}'}u_{i_v\boldsymbol{k}}).
\end{aligned} \tag{2-20}$$

In (2-18) and (2-19), $f, g$ are periodic functions in the unit cell, and can be represented using their Fourier representations. For instance,

$$f(\boldsymbol{r}) = \sum_{\boldsymbol{G}\in\mathbb{L}^*} \hat{f}(\boldsymbol{G})e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}}, \tag{2-21}$$

and its Fourier coefficients can be computed as

$$\hat{f}(\boldsymbol{G}) = \frac{1}{|\Omega|}\int_{\Omega} e^{-\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}} f(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}. \tag{2-22}$$

Hence, Parseval's identity reads

$$\int_{\Omega}\bar{f}(\boldsymbol{r})g(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r} = |\Omega|\sum_{\boldsymbol{G}\in\mathbb{L}^*}\bar{\hat{f}}(\boldsymbol{G})\hat{g}(\boldsymbol{G}). \tag{2-23}$$

Both of the kernels $V$, $W$ satisfy the translation symmetry

$$V(\boldsymbol{r}+\boldsymbol{R},\boldsymbol{r}'+\boldsymbol{R}) = V(\boldsymbol{r},\boldsymbol{r}'),\quad W(\boldsymbol{r}+\boldsymbol{R},\boldsymbol{r}'+\boldsymbol{R}) = W(\boldsymbol{r},\boldsymbol{r}')\ \text{ for all } \boldsymbol{R}\in\mathbb{L}. \tag{2-24}$$

Equation (2-24) also defines the values of $V$, $W$ for $\boldsymbol{r}$, $\boldsymbol{r}'$ beyond the supercell $\Omega^\ell$. The Fourier representation of $V$ takes the form

$$V(\boldsymbol{r},\boldsymbol{r}') = \frac{1}{|\Omega^\ell|}\sum_{\boldsymbol{k}\in\mathcal{K}^\ell}\sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}}\,\widehat{V}_{\boldsymbol{k}}(\boldsymbol{G},\boldsymbol{G}')e^{-\mathrm{i}(\boldsymbol{k}+\boldsymbol{G}')\cdot\boldsymbol{r}'}, \tag{2-25}$$

and the Fourier coefficients can be computed as

$$\widehat{V}_{\boldsymbol{k}}(\boldsymbol{G},\boldsymbol{G}') = \frac{1}{|\Omega^\ell|}\int_{\Omega^\ell\times\Omega^\ell}\mathrm{d}\boldsymbol{r}\,\mathrm{d}\boldsymbol{r}'\,e^{-\mathrm{i}(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}}\,V(\boldsymbol{r},\boldsymbol{r}')e^{\mathrm{i}(\boldsymbol{k}+\boldsymbol{G}')\cdot\boldsymbol{r}'}. \tag{2-26}$$

Similarly, the Fourier representation for $W$ can be defined.

It should be noted that the Coulomb kernel $V$ only depends on the distance between $\boldsymbol{r}$ and $\boldsymbol{r}'$, i.e., it has the further translational symmetry property that

$$V(\boldsymbol{r}+\boldsymbol{r}'', \boldsymbol{r}'+\boldsymbol{r}'') = V(\boldsymbol{r}, \boldsymbol{r}') \quad \text{for all } \boldsymbol{r}'' \in \Omega^{\ell}. \tag{2-27}$$

As a result, its Fourier transform $\widehat{V}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}')$ can be simplified into a diagonal matrix

$$\widehat{V}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}') = \frac{4\pi}{|\boldsymbol{k}+\boldsymbol{G}|^2}\delta_{\boldsymbol{G},\boldsymbol{G}'}. \tag{2-28}$$

In fact, the Coulomb kernel periodized with respect to the supercell $\Omega^{\ell}$ is defined to be the inverse Fourier transform of (2-28).

Using such notation, we have

$$\begin{aligned}
&\int_{\Omega^{\ell}} V(\boldsymbol{r}, \boldsymbol{r}')g(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' \\
&= \frac{1}{|\Omega^{\ell}|}\int_{\Omega^{\ell}} \mathrm{d}\boldsymbol{r}' \sum_{\boldsymbol{k}\in\mathcal{K}^{\ell}} \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}') e^{-\mathrm{i}(\boldsymbol{k}+\boldsymbol{G}')\cdot\boldsymbol{r}'} g(\boldsymbol{r}') \\
&= \frac{1}{|\Omega^{\ell}|}\sum_{\boldsymbol{R}\in\mathbb{L}}\int_{\Omega} \mathrm{d}\boldsymbol{r}' \sum_{\boldsymbol{k}\in\mathcal{K}^{\ell}} \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}') e^{-\mathrm{i}(\boldsymbol{k}+\boldsymbol{G}')\cdot(\boldsymbol{r}'+\boldsymbol{R})} g(\boldsymbol{r}'+\boldsymbol{R}) \\
&= \frac{1}{|\Omega^{\ell}|}\int_{\Omega} \mathrm{d}\boldsymbol{r}' \sum_{\boldsymbol{k}\in\mathcal{K}^{\ell}} \sum_{\boldsymbol{R}\in\mathbb{L}} e^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{R}} \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}') e^{-\mathrm{i}(\boldsymbol{k}+\boldsymbol{G}')\cdot\boldsymbol{r}'} g(\boldsymbol{r}').
\end{aligned} \tag{2-29}$$

Here we have used $e^{-\mathrm{i}\boldsymbol{G}'\cdot\boldsymbol{R}} = 1$ and the fact that $g$ is periodic with respect to the unit cell $\Omega$, as well as the identity

$$\int_{\Omega^{\ell}} f(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' = \sum_{\boldsymbol{R}\in\mathbb{L}} \int_{\Omega} f(\boldsymbol{r}'+\boldsymbol{R})\,\mathrm{d}\boldsymbol{r}'. \tag{2-30}$$

Furthermore, from (2-22) and the identity

$$\sum_{\boldsymbol{R}\in\mathbb{L}} e^{-\mathrm{i}\boldsymbol{k}\cdot\boldsymbol{R}} = N_k \delta_{\boldsymbol{k},0}$$

we have

$$\begin{aligned}
\int_{\Omega^{\ell}} V(\boldsymbol{r}, \boldsymbol{r}')g(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' &= \frac{1}{|\Omega|}\int_{\Omega} \mathrm{d}\boldsymbol{r}' \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{0}}(\boldsymbol{G}, \boldsymbol{G}') e^{-\mathrm{i}\boldsymbol{G}'\cdot\boldsymbol{r}'} g(\boldsymbol{r}') \\
&= \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{0}}(\boldsymbol{G}, \boldsymbol{G}') \hat{g}(\boldsymbol{G}').
\end{aligned} \tag{2-31}$$

Compared to (2-28), the definition of $\widehat{V}_{\boldsymbol{0}}$ should be modified to

$$\widehat{V}_{\boldsymbol{0}}(\boldsymbol{G}, \boldsymbol{G}') = \begin{cases} (4\pi/|\boldsymbol{G}|^2)\delta_{\boldsymbol{G},\boldsymbol{G}'}, & \boldsymbol{G} \neq \boldsymbol{0}, \\ 0, & \boldsymbol{G} = \boldsymbol{0}. \end{cases} \tag{2-32}$$

Another way to understand (2-32) is that it can only be applied to a mean-zero function $g(\boldsymbol{r})$, such that $\hat{g}(\boldsymbol{0}) = 0$. In other words, $g$ should be in the range of the Laplacian operator with the periodic boundary condition. This is indeed correct for BSE calculations, due to the orthogonality condition between the valence and conduction bands

$$\int_{\Omega} \bar{u}_{i_c\boldsymbol{k}}(\boldsymbol{r}) u_{i_v\boldsymbol{k}}(\boldsymbol{r}) \,\mathrm{d}\boldsymbol{r} = 0.$$

This implies

$$\begin{aligned}\mathcal{V}(f, g) &= \frac{1}{N_k} \int_{\Omega^\ell} \bar{f}(\boldsymbol{r}) \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{0}}(\boldsymbol{G}, \boldsymbol{G}') \hat{g}(\boldsymbol{G}') \\ &= \int_{\Omega} \bar{f}(\boldsymbol{r}) \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}} \widehat{V}_{\boldsymbol{0}}(\boldsymbol{G}, \boldsymbol{G}') \hat{g}(\boldsymbol{G}') \\ &= |\Omega| \sum_{\boldsymbol{G},\boldsymbol{G}'} \bar{\hat{f}}(\boldsymbol{G}) \widehat{V}_{\boldsymbol{0}}(\boldsymbol{G}, \boldsymbol{G}') \hat{g}(\boldsymbol{G}') \\ &= |\Omega| \sum_{\boldsymbol{G}\neq\boldsymbol{0}} \frac{4\pi}{|\boldsymbol{G}|^2} \bar{\hat{f}}(\boldsymbol{G}) \hat{g}(\boldsymbol{G}). \end{aligned} \tag{2-33}$$

Similarly for the $W$ part,

$$\begin{aligned}&\int_{\Omega^\ell} e^{-\mathrm{i}\boldsymbol{q}\cdot(\boldsymbol{r}-\boldsymbol{r}')} W(\boldsymbol{r}, \boldsymbol{r}') g(\boldsymbol{r}') \,\mathrm{d}\boldsymbol{r}' \\ &= \frac{1}{|\Omega^\ell|} \int_{\Omega^\ell} \mathrm{d}\boldsymbol{r}' e^{-\mathrm{i}\boldsymbol{q}\cdot(\boldsymbol{r}-\boldsymbol{r}')} \sum_{\boldsymbol{k}\in\mathcal{K}^\ell} \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{k}+\boldsymbol{G})\cdot\boldsymbol{r}} \widehat{W}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}') e^{-\mathrm{i}(\boldsymbol{k}+\boldsymbol{G}')\cdot\boldsymbol{r}'} g(\boldsymbol{r}') \\ &= \frac{1}{|\Omega^\ell|} \int_{\Omega} \mathrm{d}\boldsymbol{r}' e^{\mathrm{i}(\boldsymbol{k}-\boldsymbol{q})\cdot(\boldsymbol{r}-\boldsymbol{r}')} \sum_{\boldsymbol{k}\in\mathcal{K}^\ell} \sum_{\boldsymbol{R}\in\mathbb{L}} e^{-\mathrm{i}(\boldsymbol{k}-\boldsymbol{q})\cdot\boldsymbol{R}} \\ &\qquad\qquad \times \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}} \widehat{W}_{\boldsymbol{k}}(\boldsymbol{G}, \boldsymbol{G}') e^{-\mathrm{i}\boldsymbol{G}'\cdot\boldsymbol{r}'} g(\boldsymbol{r}'). \end{aligned} \tag{2-34}$$

In order to obtain a nonvanishing quantity in the equation above, note that the quantity $\sum_{\boldsymbol{R}\in\mathbb{L}} e^{-\mathrm{i}(\boldsymbol{k}-\boldsymbol{q})\cdot\boldsymbol{R}} = N_k$ if $\boldsymbol{k}-\boldsymbol{q} \in \mathbb{L}^*$, and is otherwise 0. Therefore, the summation with respect to $\boldsymbol{k}$ should be restricted to those satisfying

$$\boldsymbol{k}-\boldsymbol{q} = \boldsymbol{G}'', \quad \boldsymbol{G}'' \in \mathbb{L}^*.$$

Since $\boldsymbol{k}$ is restricted to the first Brillouin zone, there is a unique $\boldsymbol{G}''$ (and therefore $\boldsymbol{k}$) for each given $\boldsymbol{q}$ satisfying this relation. Also note that $\boldsymbol{k}-\boldsymbol{q}$ may exceed the first Brillouin zone. In other words, it is indeed possible to have $\boldsymbol{G}'' \neq \boldsymbol{0}$. Then for a given $\boldsymbol{q}$,

$$\begin{aligned}
&\int_{\Omega^\ell} e^{-\mathrm{i}\boldsymbol{q}\cdot(\boldsymbol{r}-\boldsymbol{r}')} W(\boldsymbol{r},\boldsymbol{r}')g(\boldsymbol{r}')\,\mathrm{d}\boldsymbol{r}' \\
&\qquad = \frac{1}{|\Omega|}\int_\Omega \mathrm{d}\boldsymbol{r}' \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{G}+\boldsymbol{G}'')\cdot\boldsymbol{r}}\,\widehat{W}_{\boldsymbol{G}''+\boldsymbol{q}}(\boldsymbol{G},\boldsymbol{G}')e^{-\mathrm{i}(\boldsymbol{G}'+\boldsymbol{G}'')\cdot\boldsymbol{r}'} g(\boldsymbol{r}') \\
&\qquad = \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}(\boldsymbol{G}+\boldsymbol{G}'')\cdot\boldsymbol{r}}\,\widehat{W}_{\boldsymbol{G}''+\boldsymbol{q}}(\boldsymbol{G},\boldsymbol{G}')\hat{g}(\boldsymbol{G}'+\boldsymbol{G}'') \\
&\qquad = \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}}\,\widehat{W}_{\boldsymbol{G}''+\boldsymbol{q}}(\boldsymbol{G}-\boldsymbol{G}'',\boldsymbol{G}'-\boldsymbol{G}'')\hat{g}(\boldsymbol{G}') \\
&\qquad = \sum_{\boldsymbol{G},\boldsymbol{G}'} e^{\mathrm{i}\boldsymbol{G}\cdot\boldsymbol{r}}\,\widehat{W}_{\boldsymbol{q}}(\boldsymbol{G},\boldsymbol{G}')\hat{g}(\boldsymbol{G}'). \qquad (2\text{-}35)
\end{aligned}$$

In the last equality, we have used the definition of the Fourier coefficients in (2-26). We then readily have

$$\mathcal{W}_{\boldsymbol{q}}(f,g) = |\Omega| \sum_{\boldsymbol{G},\boldsymbol{G}'} \bar{\hat{f}}(\boldsymbol{G})\widehat{W}_{\boldsymbol{q}}(\boldsymbol{G},\boldsymbol{G}')\hat{g}(\boldsymbol{G}'). \qquad (2\text{-}36)$$

Therefore, despite that $\mathcal{W}_{\boldsymbol{q}}(f,g)$ is significantly more complex to define, the resulting formula in the Fourier representation is remarkably similar to the form of $\mathcal{V}(f,g)$.

## 3. Interpolative separable density fitting for periodic systems

In order to reduce the computational complexity, we seek to minimize the number of integrals in (2-16). We will use the interpolative separable density fitting decomposition (ISDF) [19; 20]. For periodic systems, we first consider the general form of decomposition

$$Z_{i\boldsymbol{k},j\boldsymbol{k}'}(\boldsymbol{r}) := u_{i\boldsymbol{k}}(\boldsymbol{r})\bar{u}_{j\boldsymbol{k}'}(\boldsymbol{r}) \approx \sum_{\mu=1}^{N_\mu} \zeta_\mu(\boldsymbol{r})u_{i\boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu)\bar{u}_{j\boldsymbol{k}'}(\hat{\boldsymbol{r}}_\mu). \qquad (3\text{-}1)$$

When the unit cell is discretized into a uniform grid $\{\boldsymbol{r}_n\}_{n=1}^{N_g}$, $Z$ can be viewed as a matrix with its row index being $\boldsymbol{r}$, and the column index being a multi-index $(i\boldsymbol{k}, j\boldsymbol{k}')$. The matrix size is thus $N_g \times N^2 N_k^2$ (recall that $N = N_v + N_c$). For a given $\boldsymbol{r}$, $u_{i\boldsymbol{k}}(\boldsymbol{r})\bar{u}_{j\boldsymbol{k}'}(\boldsymbol{r})$ can be viewed as a row vector of size $N^2 N_k^2$. The ISDF decomposition then states that all such matrix rows can be approximately expanded using a linear combination of matrix rows with respect to a selected set of *interpolation points* $\{\hat{\boldsymbol{r}}_\mu\}_{\mu=1}^{N_\mu} \subset \{\boldsymbol{r}_i\}_{i=1}^{N_g}$. The coefficients of such a linear combination, or *interpolating vectors*, are denoted by $\{\zeta_\mu(\boldsymbol{r})\}_{\mu=1}^{N_\mu}$. Here $N_\mu$ can be interpreted as the numerical rank of the ISDF decomposition.

The compression of the pair products $u_{i\boldsymbol{k}}(\boldsymbol{r})\bar{u}_{j\boldsymbol{k}'}(\boldsymbol{r})$ can be understood from the following two limits. First, if only the $\Gamma$-point is used to sample the Brillouin zone, we find that there are $N_v N_c \sim N^2$ pairs of functions. However, the number of grid points $N_g$ only scales linearly with respect to $N$. Hence, the numerical rank of the pair products must scale asymptotically as $\mathbb{O}(N)$. In fact, when all orbitals are smooth functions, we can expect the numerical rank $N_\mu$ to be much lower than $N_g$. This statement has been confirmed by recent analysis [17]. Second, if a large number of $\boldsymbol{k}$-points are used to discretize the Brillouin zone, $N_v$, $N_c$ are often relatively small, and the number of grid points in the unit cell $N_g$ does not increase with respect to $N_k$. Hence, as $N_k$ increases, we may also expect that the numerical rank $N_\mu$ will be determined by smoothness of $u$ with respect to $\boldsymbol{r}$, $\boldsymbol{k}$, and is asymptotically independent of $N_k$. This is indeed what has been observed numerically [20]. Throughout the discussion below, we will focus on the second scenario, i.e., we will explicitly write down the scaling with respect to $N_g$, $N$, and $N_k$, but we will primarily focus on the scaling with respect to $N_k$.

Assuming the interpolation points $\{\hat{\boldsymbol{r}}_\mu\}_{\mu=1}^{N_\mu}$ are already chosen, the interpolation vectors can be efficiently evaluated using a least squares method as follows [12]. Using a linear algebra notation, (3-1) can be written as

$$Z \approx \Theta C. \tag{3-2}$$

Here $\Theta = [\zeta_1, \zeta_2, \dots, \zeta_{N_\mu}]$ contains the interpolating vectors. Each column of $C$ indexed by $(i\boldsymbol{k}, j\boldsymbol{k}')$ is given by

$$[u_{i\boldsymbol{k}}(\hat{\boldsymbol{r}}_1)\bar{u}_{j\boldsymbol{k}'}(\hat{\boldsymbol{r}}_1), \dots, u_{i\boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu)\bar{u}_{j\boldsymbol{k}'}(\hat{\boldsymbol{r}}_\mu), \dots, u_{i\boldsymbol{k}}(\hat{\boldsymbol{r}}_{N_\mu})\bar{u}_{j\boldsymbol{k}'}(\hat{\boldsymbol{r}}_{N_\mu})]^\top.$$

Equation (3-2) is an over-determined linear system with respect to the interpolation vectors $\Theta$. The least squares approximation to the solution is given by

$$\Theta = ZC^*(CC^*)^{-1}. \tag{3-3}$$

Due to the tensor product structure of $Z$ and $C$, the matrix–matrix multiplications $ZC^*$ and $CC^*$ can be carried out efficiently [12], with computational cost $\mathbb{O}(N_g N_\mu N N_k)$ and $\mathbb{O}(N_\mu^2 N N_k)$, respectively. The cost of inverting the matrix $CC^*$ is $\mathbb{O}(N_\mu^3)$, and the overall cost of evaluating $\Theta$ is thus bounded by $\mathbb{O}(N_g N_\mu N N_k + N_\mu^3 + N_g N_\mu^2)$. Hence, the cost scales cubically with respect to the number of electrons in the unit cell, and linearly with respect to the number of $\boldsymbol{k}$-points.

Equation (3-1) is the general form of ISDF. In the BSE calculations, we may further distinguish whether $i$, $j$ should take valence or conduction band indices only, as well as whether $\boldsymbol{k}$, $\boldsymbol{k}'$ can be set to be the same. For instance, (2-17) suggests that in order to compress $V_A$, $V_B$, we only need the ISDF decomposition

$$Z^V_{i_c i_v \boldsymbol{k}}(\boldsymbol{r}) := u_{i_c \boldsymbol{k}}(\boldsymbol{r}) \bar{u}_{i_v \boldsymbol{k}}(\boldsymbol{r}) \approx \sum_{\mu=1}^{N_\mu^V} \zeta_\mu^V(\boldsymbol{r}) u_{i_c \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) \bar{u}_{i_v \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu). \tag{3-4}$$

Note that the number of columns of the matrix $Z^V$ is only $N_v N_c N_k$, and the number of fitting functions $N_\mu^V$ can be chosen to be less than $N_\mu$. The computation of $W_A$, $W_B$ requires the general ISDF format (3-1).

The interpolation points $\{\hat{\boldsymbol{r}}_\mu\}_{\mu=1}^{N_\mu}$ can be chosen in different ways. In this work we employ a randomized variant of QR with column pivoting (QRCP) [19; 20; 9]. Another recently developed method is based on the centroidal Voronoi decomposition (CVT) [8]. We observed that in our examples it is even possible to work with coarse uniform grids as interpolation points, reducing the computational effort for finding the points to essentially zero while only slightly increasing the error. Since the computation of interpolation points is not the bottleneck in our problem, however, we stick to the previously developed techniques.

## 4. Fast algorithm for applying the BSH to a vector

Once the ISDF decomposition is obtained, we may compute the matrix elements

$$\widetilde{V}_{A,\mu\nu} = \mathcal{V}(\zeta_\mu^V, \zeta_\nu^V), \quad \widetilde{V}_{B,\mu\nu} = \mathcal{V}(\zeta_\mu^V, \bar{\zeta}_\nu^V), \quad \mu, \nu = 1, \dots, N_\mu^V, \tag{4-1}$$

and similarly

$$\widetilde{W}_{\boldsymbol{q},\mu\nu} = \mathcal{W}_{\boldsymbol{q}}(\zeta_\mu, \zeta_\nu), \quad \mu, \nu = 1, \dots, N_\mu. \tag{4-2}$$

The expressions in (2-17) can then be approximated in the ISDF format as

$$\begin{aligned}
V_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &\approx \frac{1}{N_k} \sum_{\mu,\nu=1}^{N_\mu^V} \bar{u}_{i_c \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) u_{i_v \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) \widetilde{V}_{A,\mu\nu} \bar{u}_{j_v \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu) u_{j_c \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu),\\
V_B(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &\approx \frac{1}{N_k} \sum_{\mu,\nu=1}^{N_\mu^V} \bar{u}_{i_c \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) u_{i_v \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) \widetilde{V}_{B,\mu\nu} \bar{u}_{j_c \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu) u_{j_v \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu),\\
W_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k} \sum_{\mu,\nu=1}^{N_\mu} \bar{u}_{i_c \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) u_{j_c \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\mu) \widetilde{W}_{\boldsymbol{k}-\boldsymbol{k}',\mu\nu} \bar{u}_{j_v \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu) u_{i_v \boldsymbol{k}}(\hat{\boldsymbol{r}}_\nu),\\
W_B(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') &= \frac{1}{N_k} \sum_{\mu,\nu=1}^{N_\mu} \bar{u}_{i_c \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) u_{j_v \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\mu) \widetilde{W}_{\boldsymbol{k}-\boldsymbol{k}',\mu\nu} \bar{u}_{j_c \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu) u_{i_v \boldsymbol{k}}(\hat{\boldsymbol{r}}_\nu).
\end{aligned} \tag{4-3}$$

In order to use the Fourier representation (2-33) and (2-36), we first need to perform Fourier transforms for $\{\zeta_\mu^V\}$ and $\{\zeta_\mu\}$. Using the fast Fourier transform (FFT), and assuming that the number of Fourier coefficients $\boldsymbol{G}$ is also $N_g$, the computational cost for the Fourier transform scales as $\mathbb{O}(N_\mu^V N_g \log N_g)$ and $\mathbb{O}(N_\mu N_g \log N_g)$, respectively. The Fourier coefficients $\widehat{V}_{\boldsymbol{k}}$ can be obtained analytically, and we assume the coefficients $\widehat{W}_{\boldsymbol{k}}$ are already provided from, e.g., a GW calculation. The cost for computing $\widetilde{V}_A$, $\widetilde{V}_B$ using (2-33) is then $\mathbb{O}((N_\mu^V)^2 N_g)$. Similarly the cost for computing all $\widetilde{W}_{\boldsymbol{q}}$ matrices is $\mathbb{O}(N_\mu^2 N_g N_k)$. In particular, the total cost for the initial setup stage scales as $\mathbb{O}(N_k)$ with respect to the number of $\boldsymbol{k}$-points.

After this initial setup stage, each entry of the BSH can be computed with $\mathbb{O}((N_\mu^V)^2 + N_\mu^2)$ operations. If the entire BSH matrix is to be constructed, the cost will be $\mathbb{O}(N_\mu^2 N_k^2 N_v^2 N_c^2)$.

Below we demonstrate that if we only aim to apply the Hamiltonian $H_{\mathrm{BSE}}$ to an arbitrary vector without ever assembling the full Hamiltonian, the computational cost can be greatly reduced.

For simplicity, let us focus on the case when the Tamm–Dancoff approximation (TDA) is used. Applying the Hamiltonian $H_{\mathrm{BSE}} = D + 2V_A - W_B$ to a vector $X \in \mathbb{C}^{N_v N_c N_k}$ amounts to evaluating the three terms

$$\begin{aligned} [DX](i_v i_c \boldsymbol{k}) &= (\epsilon_{i_c \boldsymbol{k}} - \epsilon_{i_v \boldsymbol{k}'}) X(i_v i_c \boldsymbol{k}), \\ [V_A X](i_v i_c \boldsymbol{k}) &= \sum_{j_v, j_c, \boldsymbol{k}'} V_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') X(j_v j_c \boldsymbol{k}'), \\ [W_A X](i_v i_c \boldsymbol{k}) &= \sum_{j_v, j_c, \boldsymbol{k}'} W_A(i_v i_c \boldsymbol{k}, j_v j_c \boldsymbol{k}') X(j_v j_c \boldsymbol{k}'). \end{aligned} \tag{4-4}$$

Computing the first term for all $(i_v i_c \boldsymbol{k})$ clearly costs $\mathbb{O}(N_v N_c N_k)$ operations. We now show that the second and third terms can also be computed efficiently.

Using (4-3), the second term in (4-5) can be regrouped as

$$\frac{1}{N_k} \sum_\mu \bar{u}_{i_c \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) u_{i_v \boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) \Bigg\{ \sum_\nu \widetilde{V}_{A,\mu\nu} \\ \times \Bigg( \sum_{\boldsymbol{k}'} \Bigg( \sum_{j_c} u_{j_c \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu) \Bigg( \sum_{j_v} \bar{u}_{j_v \boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu) X(j_v j_c \boldsymbol{k}') \Bigg) \Bigg) \Bigg) \Bigg\}. \tag{4-5}$$

This means one can first perform contractions over $j_v$, $j_c$, and $\boldsymbol{k}'$ to obtain a quantity that only depends on $\hat{\boldsymbol{r}}_\nu$. The computational complexity is $\mathbb{O}(N_\mu^V (N_v N_c N_k + N_c N_k))$. The two remaining sums can be computed with $\mathbb{O}((N_\mu^V)^2 + N_\mu^V N_v N_c N_k)$ operations. The total complexity of computing $V_A X$ is bounded by $\mathbb{O}((N_\mu^V)^2 + N_\mu^V N_v N_c N_k)$.

For the third term in (4-4) we obtain

$$\frac{1}{N_k}\sum_{\nu} u_{i_v\boldsymbol{k}}(\hat{\boldsymbol{r}}_\nu)\Bigg\{\sum_{\mu}\bar{u}_{i_c\boldsymbol{k}}(\hat{\boldsymbol{r}}_\mu) \times\Bigg(\sum_{\boldsymbol{k}'}\widetilde{W}_{\boldsymbol{k}-\boldsymbol{k}',\mu\nu}\Bigg(\sum_{j_c} u_{j_c\boldsymbol{k}'}(\hat{\boldsymbol{r}}_\mu)\Bigg(\sum_{j_v}\bar{u}_{j_v\boldsymbol{k}'}(\hat{\boldsymbol{r}}_\nu)X(j_v j_c\boldsymbol{k}')\Bigg)\Bigg)\Bigg)\Bigg\}. \tag{4-6}$$

Here, we exploited the separable structure of the decomposition to reorder the products in such a way that all terms depending on $\boldsymbol{k}$ and $\boldsymbol{k}'$ are to the left and right, respectively, of $\widetilde{W}_{\boldsymbol{k}-\boldsymbol{k}',\mu\nu}$. The two innermost contractions over $j_v$ and $j_c$ result in a quantity that only depends on $\boldsymbol{k}$, $\hat{\boldsymbol{r}}_\mu$, and $\hat{\boldsymbol{r}}_\nu$. The cost for these two steps is $\mathcal{O}(N_\mu N_k N_v N_c + N_\mu^2 N_k N_c)$. The sum over $\boldsymbol{k}'$ then has the structure of a *discrete convolution*, for each fixed $\mu\nu$ pair. Therefore, it can be computed for all $\boldsymbol{k}$ simultaneously in $\mathcal{O}(N_\mu^2 N_k \log N_k)$ operations by fast convolution algorithms, e.g., by using the FFT with zero-padded vectors. The remaining summation operations over $\mu$ and $\nu$ are then obtained with $\mathcal{O}(N_\mu^2 N_c N_k + N_\mu N_v N_c N_k)$ operations. In total the computation of $W_A X$ amounts to $\mathcal{O}(N_\mu N_v N_c N_k + N_\mu^2 N_c N_k + N_\mu^2 N_k \log N_k)$ operations.

Combining the results for the three parts of the Hamiltonian, we see that the computational complexity is given by

$$\mathcal{O}\big((N_\mu + N_\mu^V)N_v N_c N_k + (N_\mu^V)^2 + N_\mu^2 N_c N_k + N_\mu^2 N_k \log N_k\big).$$

In particular, the cost with respect to the number of $\boldsymbol{k}$-points only scales as $\mathcal{O}(N_k \log N_k)$. This allows us to perform BSE calculations for complex materials which require a very large number of $\boldsymbol{k}$-points.

By avoiding the explicit construction of $H_{\text{BSE}}$, the new algorithm also drastically reduces the storage cost. The storage cost for $H_{\text{BSE}}$ alone is $\mathcal{O}((N_v N_c N_k)^2)$. In the new algorithm, the storage cost of $\widehat{W}_{\boldsymbol{q}}$ becomes the dominant component and scales only linearly with respect to $N_k$.

As an example, the matrix-free application of $H_{\text{BSE}}$ can be used to compute the optical absorption spectrum, which requires the evaluation of the quantity

$$\varepsilon_2(\omega) = \operatorname{Im}\left[\frac{8\pi}{|\Omega|} d_r^*((\omega - \mathrm{i}\eta)I - H_{\text{BSE}})^{-1} d_l\right]. \tag{4-7}$$

Here $d_r$ and $d_l$ are called the right and left optical transition vectors, and $\eta$ is a broadening factor used to account for the exciton lifetime. We also compute the smallest eigenvalues of $H_{\text{BSE}}$, which are of interest in their own right, as they represent the transition energies of bound excitons in many semiconducting solid state materials.

To observe the absorption spectrum and identify its main peaks, it is possible to use a structure-preserving iterative method instead of explicitly computing all eigenpairs of $H_{\mathrm{BSE}}$. We refer readers to [6; 34] for details of the structure-preserving Lanczos algorithm, which has been implemented in the BSEPACK [35] library.[1] When TDA is used, the structure-preserving Lanczos reduces to a standard Lanczos algorithm. For the computation of the first eigenvalue we use standard ARPACK [14] routines for Hermitian matrices.

## 5. Numerical examples

To illustrate the efficiency of ISDF for BSE calculations in crystals, we apply the method to compute the excitation modes and absorption spectra of a one-dimensional model problem as well as two real material systems, diamond (3D bulk) and graphene (quasi-2D). For both systems, we determine the optical absorption spectra on $\boldsymbol{k}$-grids close to those employed in previously published calculations to demonstrate that our method is suitable for state-of-the-art calculations, both for 3D and quasi-2D materials. We furthermore provide a numerical scaling analysis and a more detailed analysis of the error in the ISDF in the case of the one-dimensional model and diamond. We show that a good approximation of the spectrum can be obtained with a small number of interpolation vectors.

The method was implemented in the programming language Julia [5] and the source code is available.[2] As the input to our method for the actual materials, we employ the KSDFT single particle orbitals, quasiparticle energies, and screened Coulomb potential computed by `exciting` [10; 37], an all-electron full-potential code with implementations of density functional theory and many-body perturbation theory. The Tamm–Dancoff approximation is used in all calculations.

All calculation for the proposed method were carried out on a single core of an Intel Core i5-8250U CPU at 1.60 GHz.

**5.1.** ***One-dimensional problems.*** For the one-dimensional problem, we take the single particle orbitals $\psi_{i\boldsymbol{k}}(\boldsymbol{r})$ in (2-16) to be eigenfunctions of a single particle Hamiltonian $\mathscr{H}(\boldsymbol{k})$ in which the effective potential is defined as

$$V_{\mathrm{eff}}(r) = 20\cos(4\pi r/L) + 0.2\sin(2\pi r/L),$$

where the unit cell size is $|\Omega| \equiv L = 1.5$.

The bare Coulomb potential used in (2-16) is chosen to be

$$V(r, r') = \frac{1}{\sqrt{(r-r')^2 + 0.01}}, \tag{5-1}$$

[1] https://sites.google.com/a/lbl.gov/bsepack/

[2] https://github.com/fhenneke/BSE_k_ISDF.jl/

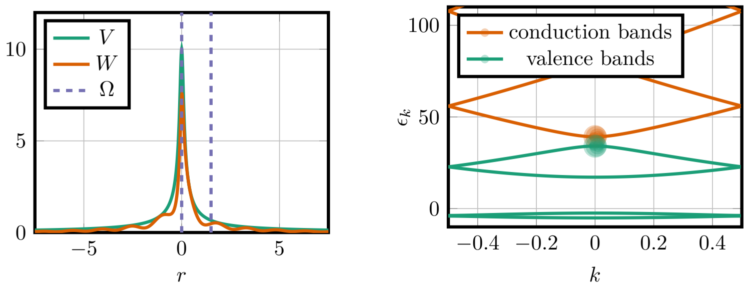


**Figure 1.** Left: the potentials $V(r,0)$ and $W(r,0)$. Right: band structure with coefficients of the lowest eigenfunction for $N_k = 128$. The areas of the circles on the valence and conduction bands at position $\boldsymbol{k}$ are proportional to $\sum_{i_c} |X(i_v i_c \boldsymbol{k})|^2$ and $\sum_{i_v} |X(i_v i_c \boldsymbol{k})|^2$.

and the screened interaction is chosen as

$$W(r,r') = \frac{(3+\sin(2\pi r/L))(3+\cos(4\pi r'/L))}{16} e^{-(r-r')^2/(32L^2)} V(r,r'). \tag{5-2}$$

Compared to the smoothed-out Coulomb potential $V$, the chosen screened interaction $W$ decays exponentially and also contains lattice periodic contributions. The potentials are shown in Figure 1. Both potentials are periodically extended $N_k - 1$ times outside of the unit cell. The particular structure of the potentials has an influence on the band structure and spectrum of the BSH, but was observed to not significantly impact the convergence behavior or the run time scaling of the ISDF method.

The Bloch functions $u_{i\boldsymbol{k}}$ are sampled on $N_g = 128$ uniformly distributed grid points within the unit cell, and the number of $\boldsymbol{k}$-points $N_k$ ranges from 16 to 4096 in our experiments.

For each $\boldsymbol{k}$-point, the first four eigenstates are treated as the valence states in this model, while the remaining eigenstates are considered as the conduction states, separated by an energy gap from the former. We use all $N_v = 4$ valence bands and $N_c = 5$ conduction bands to construct the approximate $H_{\text{BSE}}$. The number of $\boldsymbol{k}$-points was chosen to be $N_k = 256$ in the error analysis of the ISDF approximation, and varies from 16 to 4096 in the run time analysis and the analysis of the error in the absorption spectrum. The largest resulting Hamiltonian is of size 81 920 × 81 920.

Figure 2 shows how the ISDF approximation error varies with respect to the truncation parameter $N_\mu^{ij}$ and how the accuracy of the approximate spectrum of $H_{\text{BSE}}$ changes with respect to the ISDF approximation error.

In the left subfigure, we plot the relative error $\|\Theta^{\alpha\beta} C^{\alpha\beta} - Z^{\alpha\beta}\|_F / \|Z^{\alpha\beta}\|_F$, $\alpha, \beta \in \{v, c\}$, where $\|\cdot\|_F$ is the Frobenius norm, for different choices of truncation levels $N_\mu$ (or number of interpolation points). As expected, when $N_\mu$ is too small, ISDF results in relatively large error. As $N_\mu$ becomes slightly larger, the ISDF approximation error decays exponentially with respect to $N_\mu$ up to $N_\mu = 20 \sim 30$.

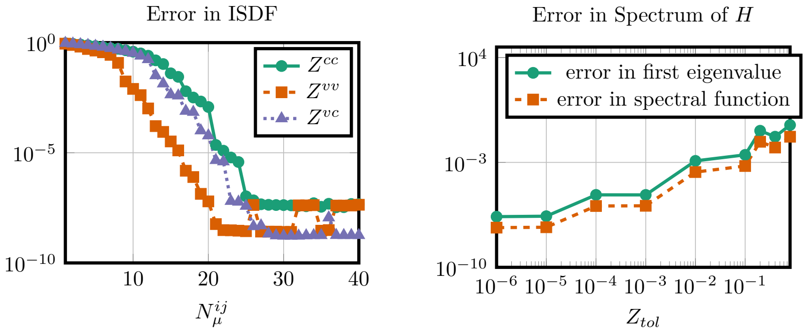


**Figure 2.** Left: ISDF approximation error $\|Z - \Theta C\|_F / \|Z\|_F$ for different choices of $N_\mu$. Right: resulting errors in the spectrum of $H_{\text{BSE}}$ for different ISDF error tolerances.

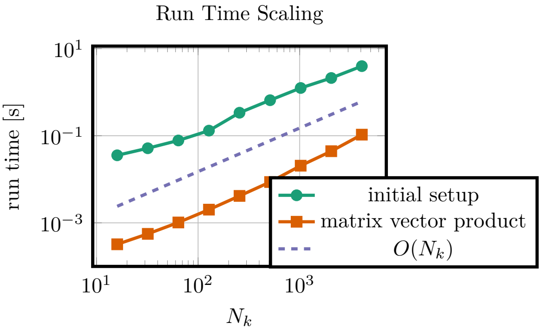


**Figure 3.** Run times for the initial setup and individual matrix-free matrix-vector products.

At this truncation level, the error is on the order of $10^{-8}$, which is sufficiently small for obtaining a highly accurate approximation of the spectrum of $H_{\text{BSE}}$ as shown in the right subfigure. In this subfigure, we plot the relative error in the first eigenvalue and in the overall optical absorption spectrum against the ISDF error tolerance $Z_{\text{tol}}$. For each $Z_{\text{tol}}$, we choose the smallest truncation parameters $N_\mu$ with the resulting error in $Z^{\alpha,\beta}$ being less than or equal to $Z_{\text{tol}}$ for $\alpha, \beta \in \{v, c\}$.

In Figure 3, we plot the timing measurements for both the construction of $\widetilde{V}$ and $\widetilde{W}$ and the multiplication of the approximate $H_{\text{BSE}}$ with a vector with respect to $N_k$. In these calculations, the ISDF truncation parameters $N_\mu$ are chosen so that the relative error in $Z^{\alpha\beta}$ is below $Z_{\text{tol}} = 10^{-5}$. This error tolerance resulted in the choices of $N_\mu^{vv} = 17$, $N_\mu^{cc} = 23$, and $N_\mu^{vc} = 21$.

As we can see in Figure 3, the scaling of the run time for the construction of $\widetilde{V}$ and $\widetilde{W}$ is nearly linear with respect to $N_k$, which is in excellent agreement with the theoretical computational complexity presented in the preceding section. The scaling of the run time for the multiplication of the approximate $H_{\text{BSE}}$ with a vector also looks linear in $N_k$. In fact, a more detailed investigation showed that the

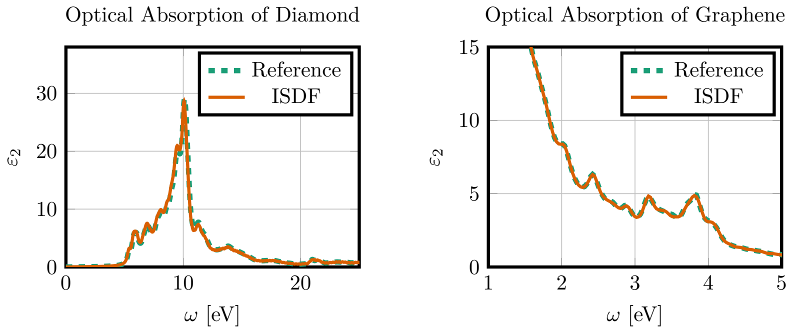


**Figure 4.** Optical absorption spectrum for diamond (left) and graphene (right).

| parameters | diamond | graphene |
|---|---|---|
| $N_v$ | 4 | 4 |
| $N_c$ | 10 | 5 |
| $N_k$ | $13 \times 13 \times 13$ | $42 \times 42 \times 1$ |
| $N_r$ | $20 \times 20 \times 20$ | $15 \times 15 \times 50$ |
| $N_\mu^{vv}$ | 70 | 50 |
| $N_\mu^{cc}$ | 220 | 180 |
| $N_\mu^{vc}$ | 100 | 60 |
| $N_{\text{iter}}$ | 150 | 100 |

**Table 1.** Parameters used in the computation of spectra and the benchmarks.

convolutions in $\boldsymbol{k}$ in the application of $W$ dominate the cost of the matrix-vector multiplications, in good agreement with the theoretical $\mathbb{O}(N_k \log N_k)$ complexity shown earlier.

For comparison, without the use of ISDF, the construction of $H_{\text{BSE}}$ is estimated to take about 460 000 seconds for $N_k = 4096$. With our method it took less than 10 seconds.

**5.2.** ***Three-dimensional problems.*** We now compare optical absorption spectra for diamond and graphene computed from the approximate $H_{\text{BSE}}$ constructed via ISDF with corresponding reference spectra. The reference spectra are obtained from the exact $H_{\text{BSE}}$ from the `exciting` code [10; 37]. The comparison is shown in Figure 4. The reference spectrum for diamond is constructed on a $13 \times 13 \times 13$ $\boldsymbol{k}$-grid using all 4 valence and 10 conduction states. Fourier components $\widehat{W}_{\boldsymbol{q}}(\boldsymbol{G}, \boldsymbol{G}')$ in (2-35) are calculated up to a cutoff $|\boldsymbol{G} + \boldsymbol{q}| \leq 2.5\,\mathrm{a}_0^{-1}$, where $\mathrm{a}_0$ is the Bohr radius. The screened Coulomb interaction is calculated within the random-phase approximation (RPA) including 100 conduction states. For graphene, the reference spectrum is obtained on a $42 \times 42 \times 1$ $\boldsymbol{k}$-grid using all 4 valence and 5 conduction states. Fourier

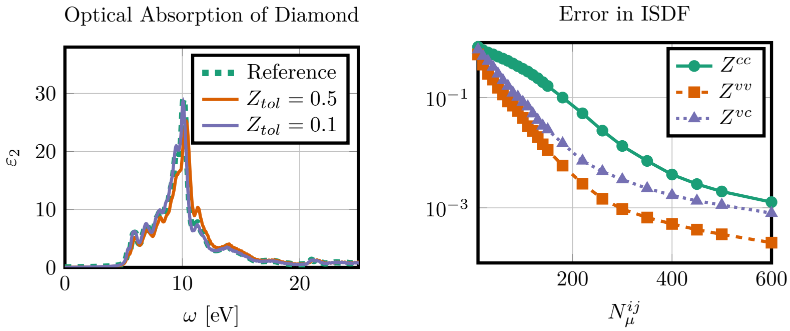


**Figure 5.** Left: optical absorption spectrum for diamond with differently accurate ISDF approximations. Right: estimated errors in ISDF approximation with different numbers of interpolation points.

| | error in | | |
|---|---|---|---|
| $Z_{\text{tol}}$ | absorption function | first eigenvalue | |
| 0.5 | 0.199 | 0.0038 | (20.7 meV) |
| 0.1 | 0.056 | 0.0011 | (6.2 meV) |
| 0.05 | 0.040 | 0.0006 | (3.3 meV) |

**Table 2.** Relative (and absolute) errors in the spectrum of $H_{\text{BSE}}$ for different ISDF error tolerances.

components $\widehat{W}_{\boldsymbol{q}}(\boldsymbol{G}, \boldsymbol{G}')$ in (2-35) are calculated up to a cutoff $|\boldsymbol{G}+\boldsymbol{q}| \leq 2.0\,\mathrm{a}_0^{-1}$, and 80 conduction states are included in the RPA calculations for the screened Coulomb potential. The numerical parameters of the reference and approximate calculations are shown in Table 1. The number of interpolation vectors was chosen such that the relative ISDF error was around 0.1.

We can clearly see that for both diamond and graphene, the approximate optical absorption spectrum matches well with the reference spectrum. In particular, the positions and heights of all major peaks are in good agreement. We should note that, in the case of diamond, the absorption spectrum produced by a $13 \times 13 \times 13$ $\boldsymbol{k}$-grid is in good agreement with measurements [26] and previous BSE calculations [11]. In the case of graphene, however, larger $\boldsymbol{k}$-grids have been reported for BSE calculations [38] to produce an optical absorption spectrum in good agreement with the experimental result.

Figure 5 shows that the ISDF approximation error can be systematically reduced as we increase the number interpolating vectors $N_\mu$. However, Figure 4 shows that the approximate absorption spectrum is already in good agreement with the reference spectrum, when the relative ISDF approximation error is at 0.1. Thus, it seems unnecessary to use a larger number of interpolation vectors in these cases. This observation is corroborated by the relative difference between the first eigenvalue

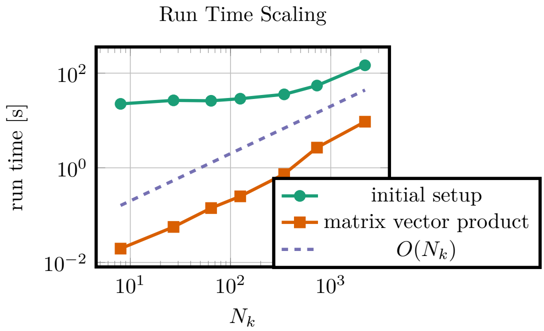


**Figure 6.** Run times for the initial setup and individual matrix-free matrix-vector products.

of the approximate $H_{\mathrm{BSE}}$ computed using ARPACK and that of reference $H_{\mathrm{BSE}}$ constructed in `exciting` shown in Table 2. With a relative ISDF approximation error of $Z_{\mathrm{tol}} = 0.1$, the error in the first BSE eigenvalue is below 10 meV in both examples shown here.

To illustrate the run time scaling of the method in the 3D examples, we measure the time it takes to construct the approximate $H_{\mathrm{BSE}}$ via ISDF as well as the time it takes to multiply the resulting $H_{\mathrm{BSE}}$ with vectors for the diamond example. We use $\boldsymbol{k}$-grids of sizes $N_k = n_k \times n_k \times n_k$ for $n_k \in \{2, 3, 4, 5, 7, 9, 13\}$. The resulting timing measurements are plotted in Figure 6. It can be seen that the run time for constructing the approximate $H_{\mathrm{BSE}}$ scales linearly with the number of $\boldsymbol{k}$-points. The multiplication of $H_{\mathrm{BSE}}$ with vectors scales as $\mathbb{O}(N_k \log N_k)$ for sufficiently large $N_k$. As in the model problem, the convolutions in $\boldsymbol{k}$ in the application of $W$ dominate the cost of the matrix–vector multiplications. For comparison, computing the ISDF decomposition of the Hamiltonian for the case $N_k = 13^3$ took 147 seconds, whereas the full assembly of the Hamiltonian took about 6 hours in `exciting` on 13 compute nodes with 13 cores each. The optical absorption function was obtained by running about 150 Lanczos steps, which amounts to about 24 minutes for each fixed direction ($x$, $y$, and $z$), compared to almost 4 hours required in the `exciting` code for the full diagonalization on 13 compute nodes.

## 6. Conclusion

In this paper, we examined the possibility of using the ISDF technique to reduce the computational complexity of BSH construction and the subsequent iterative approximation of the optical absorption spectrum and excitation energies of electron-hole (exciton) pairs for solids. For periodic systems, a fine $\boldsymbol{k}$-point sampling in the Brillouin zone is often required to produce accurate results, whereas the number of bands per $\boldsymbol{k}$-point required to construct the bare exchange and screened direct kernels of the BSH is relatively small. We showed that the complexity of the ISDF procedure scales linearly with respect to the number of $\boldsymbol{k}$-points ($N_k$) when the ranks of the approximate bare exchange and screened direct kernels produced by the ISDF procedure are chosen to be independent of $N_k$. By keeping the bare exchange and screened direct kernels in the low-rank decomposed form produced by the ISDF procedure, an iterative method used to obtain the optical absorption spectrum and selected excitation energies (eigenvalues of the BSH) can be implemented with cost scaling as $\mathbb{O}(N_k \log N_k)$. Our numerical experiments, which were performed on a 1D model as well as two different types of actual materials (diamond and graphene), confirm our complexity analysis. They demonstrate that the ISDF technique can indeed significantly reduce the cost of BSE calculation for solids while maintaining the same accuracy provided by a standard BSE calculation implemented in the software `exciting`. Our current implementation of the ISDF technique is done using the Julia programming language for a single node. A distributed parallel implementation is needed to accommodate a much finer $\boldsymbol{k}$-point sampling which is required in the case of the graphene example to produce a computed absorption spectrum that matches with experimental results.

## Acknowledgments

This work was partially supported by the U.S. Department of Energy (DOE) under grant DE-SC0017867 (Lin), by the Center for Computational Study of Excited-State Phenomena in Energy Materials (C2SEPEM) at the Lawrence Berkeley National Laboratory, which is funded by the DOE, Office of Science, Basic Energy Sciences, Materials Sciences and Engineering Division, under contract number DE-AC02-05CH11231 (Yang), by the Scientific Discovery Through Advanced Computing (SciDAC) program, and by the CAMERA program (Lin and Yang). Within a framework of cooperation between the University of California, Berkeley and the Freie Universität Berlin, the latter sponsored an extended visit of Henneke and Klein in Berkeley. We thank Wei Hu, Meiyue Shao, and Kyle Thicke for helpful discussions. Draxl and Klein thank the Institute for Pure and Applied Mathematics at the University of California, Los Angeles for its support during the 2013 fall program on “Materials for a sustainable energy future” and for creating the inspiring scientific atmosphere that initiated their collaboration.

## References

[1] S. Albrecht, G. Onida, and L. Reining, *Ab initio calculation of the quasiparticle spectrum and excitonic effects in* $\mathrm{Li_2O}$, Phys. Rev. B **55** (1997), no. 16, 10278–10281.

[2] N. W. Ashcroft and N. D. Mermin, *Solid state physics*, Harcourt, New York, 1976.

[3] P. Benner, V. Khoromskaia, and B. N. Khoromskij, *A reduced basis approach for calculation of the Bethe–Salpeter excitation energies by using low-rank tensor factorisations*, Mol. Phys. **114** (2016), no. 7–8, 1148–1161.

[4] P. Benner, S. Dolgov, V. Khoromskaia, and B. N. Khoromskij, *Fast iterative solution of the Bethe–Salpeter eigenvalue problem using low-rank and QTT tensor approximation*, J. Comput. Phys. **334** (2017), 221–239. MR Zbl

[5] J. Bezanson, A. Edelman, S. Karpinski, and V. B. Shah, *Julia: a fresh approach to numerical computing*, SIAM Rev. **59** (2017), no. 1, 65–98. MR Zbl

[6] J. Brabec, L. Lin, M. Shao, N. Govind, C. Yang, Y. Saad, and E. G. Ng, *Efficient algorithms for estimating the absorption spectrum within linear response TDDFT*, J. Chem. Theory Comput. **11** (2015), no. 11, 5197–5208.

[7] J. Deslippe, G. Samsonidze, D. A. Strubbe, M. Jain, M. L. Cohen, and S. G. Louie, *BerkeleyGW: a massively parallel computer package for the calculation of the quasiparticle and optical properties of materials and nanostructures*, Comput. Phys. Commun. **183** (2012), no. 6, 1269–1289.

[8] K. Dong, W. Hu, and L. Lin, *Interpolative separable density fitting through centroidal Voronoi tessellation with applications to hybrid functional electronic structure calculations*, J. Chem. Theory Comput. **14** (2018), no. 3, 1311–1320.

[9] G. H. Golub and C. F. Van Loan, *Matrix computations*, 4th ed., Johns Hopkins University, Baltimore, MD, 2013. MR Zbl

[10] A. Gulans, S. Kontur, C. Meisenbichler, D. Nabok, P. Pavone, S. Rigamonti, S. Sagmeister, U. Werner, and C. Draxl, *exciting: a full-potential all-electron package implementing density-functional theory and many-body perturbation theory*, J. Phys. Condens. Mat. **26** (2014), no. 36, art. id. 363202.

[11] P. H. Hahn, K. Seino, W. G. Schmidt, J. Furthmüller, and F. Bechstedt, *Quasiparticle and excitonic effects in the optical spectra of diamond,* SiC*,* Si*,* GaP*,* GaAs*,* InP*, and* AlN, Phys. Status Solidi B **242** (2005), no. 13, 2720–2728.

[12] W. Hu, L. Lin, and C. Yang, *Interpolative separable density fitting decomposition for accelerating hybrid density functional calculations with applications to defects in silicon*, J. Chem. Theory Comput. **13** (2017), no. 11, 5420–5431.

[13] W. Hu, M. Shao, A. Cepellotti, F. H. da Jornada, L. Lin, K. Thicke, C. Yang, and S. G. Louie, *Accelerating optical absorption spectra and exciton energy computation via interpolative separable density fitting*, ICCS 2018, II (Y. Shi, H. Fu, Y. Tian, V. V. Krzhizhanovskaya, M. H. Lees, J. Dongarra, and P. M. A. Sloot, eds.), Lecture Notes in Comput. Sci., no. 10861, Springer, 2018, pp. 604–617. MR

[14] R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted arnoldi methods*, Software, Environments, and Tools, no. 6, Society for Industrial and Applied Mathematics, Philadelphia, PA, 1998. MR Zbl

[15] L. Lin, Z. Xu, and L. Ying, *Adaptively compressed polarizability operator for accelerating large scale ab initio phonon calculations*, Multiscale Model. Simul. **15** (2017), no. 1, 29–55. MR Zbl

[16] M. P. Ljungberg, P. Koval, F. Ferrari, D. Foerster, and D. Sánchez-Portal, *Cubic-scaling iterative solution of the Bethe–Salpeter equation for finite systems*, Phys. Rev. B **92** (2015), no. 7, art. id. 075422.

[17] J. Lu, C. D. Sogge, and S. Steinerberger, *Approximating pointwise products of Laplacian eigenfunctions*, J. Funct. Anal. **277** (2019), no. 9, 3271–3282. MR Zbl

[18] J. Lu and K. Thicke, *Cubic scaling algorithms for RPA correlation using interpolative separable density fitting*, J. Comput. Phys. **351** (2017), 187–202. MR Zbl

[19] J. Lu and L. Ying, *Compression of the electron repulsion integral tensor in tensor hypercontraction format with cubic scaling cost*, J. Comput. Phys. **302** (2015), 329–335. MR Zbl

[20] ———, *Fast algorithm for periodic density fitting for Bloch waves*, Ann. Math. Sci. Appl. **1** (2016), no. 2, 321–339. MR Zbl

[21] M. Marsili, F. Mosconi, Edoardo De Angelis, and P. Umari, *Large-scale $GW$-BSE calculations with $N^3$ scaling: excitonic effects in dye-sensitized solar cells*, Phys. Rev. B **95** (2017), no. 7, art. id. 075415.

[22] H. J. Monkhorst and J. D. Pack, *Special points for Brillouin-zone integrations*, Phys. Rev. B **13** (1976), no. 12, 5188–5192. MR

[23] N. L. Nguyen, H. Ma, M. Govoni, F. Gygi, and G. Galli, *Finite-field approach to solving the Bethe–Salpeter equation*, Phys. Rev. Lett. **122** (2019), no. 23, art. id. 237402. MR

[24] G. Onida, L. Reining, R. W. Godby, R. Del Sole, and W. Andreoni, *Ab initio calculations of the quasiparticle and absorption spectra of clusters: the sodium tetramer*, Phys. Rev. Lett. **75** (1995), no. 5, 818–821.

[25] G. Onida, L. Reining, and A. Rubio, *Electronic excitations: density-functional versus many-body Green's-function approaches*, Rev. Mod. Phys. **74** (2002), no. 2, 601–659.

[26] H. R. Phillip and E. A. Taft, *Kramers–Kronig analysis of reflectance data for diamond*, Phys. Rev. **136** (1964), no. 5A, A1445–A1448.

[27] Y. Ping, D. Rocca, and G. Galli, *Electronic excitations in light absorbers for photoelectrochemical energy conversion: first principles calculations based on many body perturbation theory*, Chem. Soc. Rev. **42** (2013), 2437–2469.

[28] Y. Ping, D. Rocca, D. Lu, and G. Galli, *Ab initio calculations of absorption spectra of semiconducting nanowires within many-body perturbation theory*, Phys. Rev. B **85** (2012), no. 3, art. id. 035316.

[29] D. Y. Qiu, F. H. da Jornada, and S. G. Louie, *Optical spectrum of* $\mathrm{MoS}_2$*: many-body effects and diversity of exciton states*, Phys. Rev. Lett. **111** (2013), no. 21, art. id. 216805.

[30] D. Rocca, D. Lu, and G. Galli, *Ab initio calculations of optical absorption spectra: solution of the Bethe–Salpeter equation within density matrix perturbation theory*, J. Chem. Phys. **133** (2010), no. 16, art. id. 164109.

[31] D. Rocca, Y. Ping, R. Gebauer, and G. Galli, *Solution of the Bethe–Salpeter equation without empty electronic states: application to the absorption spectra of bulk systems*, Phys. Rev. B **85** (2012), no. 4, art. id. 045116.

[32] M. Rohlfing and S. G. Louie, *Electron-hole excitations and optical spectra from first principles*, Phys. Rev. B **62** (2000), no. 8, 4927–4944.

[33] E. E. Salpeter and H. A. Bethe, *A relativistic equation for bound-state problems*, Phys. Rev. **84** (1951), 1232–1242. MR Zbl

[34] M. Shao, F. H. da Jornada, L. Lin, C. Yang, J. Deslippe, and S. G. Louie, *A structure preserving Lanczos algorithm for computing the optical absorption spectrum*, SIAM J. Matrix Anal. Appl. **39** (2018), no. 2, 683–711. MR Zbl

[35] M. Shao and C. Yang, *BSEPACK user's guide*, user manual, 2016. arXiv

[36] G. Strinati, *Application of the Green's functions method to the study of the optical properties of semiconductors*, Riv. Nuovo Cimento **11** (1988), no. 12, 1–86.

[37] C. Vorwerk, B. Aurich, C. Cocchi, and C. Draxl, *Bethe–Salpeter equation for absorption and scattering spectroscopy: implementation in the `exciting` code*, Electron. Struct. **1** (2019), no. 3, art. id. 037001.

[38] L. Yang, J. Deslippe, C.-H. Park, M. L. Cohen, and S. G. Louie, *Excitonic effects on the optical response of graphene and bilayer graphene*, Phys. Rev. Lett. **103** (2009), no. 18, art. id. 186802.

Received December 10, 2019.

FELIX HENNEKE: felix.henneke@fu-berlin.de
*Institut für Mathematik, Freie Universität Berlin, Berlin, Germany*

LIN LIN: linlin@math.berkeley.edu
*Department of Mathematics, University of California, Berkeley, Berkeley, CA, United States*

and

*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, United States*

CHRISTIAN VORWERK: vorwerk@physik.hu-berlin.de
*Institut für Physik, IRIS Adlershof, Humboldt-Universität zu Berlin, Berlin, Germany*

CLAUDIA DRAXL: claudia.draxl@physik.hu-berlin.de
*Institut für Physik, IRIS Adlershof, Humboldt-Universität zu Berlin, Berlin, Germany*

RUPERT KLEIN: rupert.klein@fu-berlin.de
*Institut für Mathematik, Freie Universität Berlin, Berlin, Germany*

CHAO YANG: cyang@lbl.gov
*Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA, United States*

# Guidelines for Authors

Authors may submit manuscripts in PDF format on-line at the Submission page at msp.org/camcos.

**Originality**. Submission of a manuscript acknowledges that the manuscript is original and and is not, in whole or in part, published or under consideration for publication elsewhere. It is understood also that the manuscript will not be submitted elsewhere while under consideration for publication in this journal.

**Language**. Articles in CAMCoS are usually in English, but articles written in other languages are welcome.

**Required items**. A brief abstract of about 150 words or less must be included. It should be self-contained and not make any reference to the bibliography. If the article is not in English, two versions of the abstract must be included, one in the language of the article and one in English. Also required are keywords and subject classifications for the article, and, for each author, postal address, affiliation (if appropriate), and email address.

**Format**. Authors are encouraged to use LaTeX but submissions in other varieties of TeX, and exceptionally in other formats, are acceptable. Initial uploads should be in PDF format; after the refereeing process we will ask you to submit all source material.

**References**. Bibliographical references should be complete, including article titles and page ranges. All references in the bibliography should be cited in the text. The use of BibTeX is preferred but not required. Tags will be converted to the house format, however, for submission you may use the format of your choice. Links will be provided to all literature with known web locations and authors are encouraged to provide their own links in addition to those supplied in the editorial process.

**Figures**. Figures must be of publication quality. After acceptance, you will need to submit the original source files in vector graphics format for all diagrams in your manuscript: vector EPS or vector PDF files are the most useful.

Most drawing and graphing packages (Mathematica, Adobe Illustrator, Corel Draw, MATLAB, etc.) allow the user to save files in one of these formats. Make sure that what you are saving is vector graphics and not a bitmap. If you need help, please write to graphics@msp.org with details about how your graphics were generated.

**White space**. Forced line breaks or page breaks should not be inserted in the document. There is no point in your trying to optimize line and page breaks in the original manuscript. The manuscript will be reformatted to use the journal's preferred fonts and layout.

**Proofs**. Page proofs will be made available to authors (or to the designated corresponding author) at a Web site in PDF format. Failure to acknowledge the receipt of proofs or to return corrections within the requested deadline may cause publication to be postponed.

# Communications in Applied Mathematics and Computational Science

vol. 15 no. 1 2020