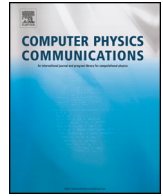




ELSEVIER

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpcCOSE ν : A collective oscillation simulation engine for neutrinos $\star, \star\star$ Manu George ^{a,*}, Chun-Yu Lin ^b, Meng-Ru Wu ^{a,c}, Tony G. Liu ^a, Zewei Xiong ^d^a Institute of Physics, Academia Sinica, Taipei, 11529, Taiwan^b National Center for High-performance computing, National Applied Research Laboratories, Hsinchu Science Park, Hsinchu City 30076, Taiwan^c Institute of Astronomy and Astrophysics, Academia Sinica, Taipei, 10617, Taiwan^d GSI Helmholtzzentrum für Schwerionenforschung, Planckstraße 1, 64291 Darmstadt, Germany

ARTICLE INFO

Article history:

Received 31 March 2022

Received in revised form 26 October 2022

Accepted 5 November 2022

Available online 11 November 2022

Keywords:

Neutrino

Flavor conversion

Etc.

ABSTRACT

We introduce the implementation details of the simulation code COSE ν , which numerically solves a set of non-linear partial differential equations that govern the dynamics of neutrino collective flavor conversions. We systematically provide the details of both finite difference method supported by Kreiss-Oliger dissipation and finite volume method with seventh order weighted essentially non-oscillatory scheme. To ensure the reliability of the code, we perform comparison of the simulation results with theoretically obtainable solutions. In order to understand and characterize the error accumulation behavior of the implementations when neutrino self-interactions are switched on, we also analyze the evolution of the deviation of the conserved quantities for different values of simulation parameters. We report the performance of our code with both CPUs and GPUs. The public version of the COSE ν package is available at <https://github.com/COSEnu/COSEnu>.

Program summary

Program Title: COSE ν CPC Library link to program files: <https://doi.org/10.17632/d4d4hg2sths.1>Developer's repository link: <https://github.com/COSEnu/COSEnu>Code Ocean capsule: <https://codeocean.com/capsule/7929908>

Licensing provisions: BSD 3-clause

Programming language: C++, Python 3.0+

Supplementary material: https://cosenu.github.io/cosenu_docs/

Nature of problem: Collective neutrino flavor oscillations is a novel but generic phenomenon that occurs in cosmological and astrophysical environments dense in neutrinos. Understanding this phenomenon requires solving the corresponding quantum kinetic transport equation, which is a set of nonlinear and hyperbolic partial differential equations.

Solution method: We adopt the method of lines to solve the hyperbolic partial differential equations in our program. The spatial derivatives are evaluated using either the finite difference method supported by 3rd order Kreiss-Oliger dissipation or the finite volume method with 7th order WENO scheme. The time integration is performed using the explicit 4th order Runge-Kutta method. Parallelization with OpenMP or OpenACC is supported.

© 2022 Elsevier B.V. All rights reserved.

 \star The review of this paper was arranged by Prof. Z. Was. $\star\star$ This paper and its associated computer program are available via the Computer Physics Communications homepage on ScienceDirect (<http://www.sciencedirect.com/science/journal/00104655>).

* Corresponding author.

E-mail address: manug@gate.sinica.edu.tw (M. George).

1. Introduction

Neutrinos are the most elusive fundamental particles in the Standard Model of particle physics. They have no electric charge and tiny masses (not yet known experimentally), and only interact with other particles via the weak force. Despite the weakly-interacting nature, they play a substantial role in deciding the evolution and the matter composition of different physical systems such as core-collapse supernovae (CCSNe), neutron star (NS) mergers, and the early Universe. Added to their elusiveness, terrestrial based experiments and astrophysical observations revealed

that neutrinos are capable of undergoing a phenomenon known as flavor transition, by virtue of which they oscillate between e , μ , and τ flavors, due to the mixing of their flavor and mass quantum eigenstates [1]. Associated with these properties comes a set of parameters: the masses of different mass eigenstates and their ordering, as well as the mixing angles and the CP-violating phase(s) in the neutrino mixing matrix. Most of these parameters have been measured precisely in recent years [1] while the ongoing and planned experiments are expected to shed more light on the remaining ones.

Recent theoretical studies revealed that neutrinos behave differently when surrounded by dense matter. Apart from the Mikheyev-Smirnov-Wolfenstein (MSW) [2,3] mechanism in which neutrinos experience resonant flavor transition due to their neutral and charged current interactions with matter, they can also undergo forward scattering among themselves via neutral current interactions resulting in processes such as $\nu_e \bar{\nu}_e \rightleftharpoons \nu_x \bar{\nu}_x$ [4,5], where $\nu_x(\bar{\nu}_x)$ represent $\nu_\mu(\bar{\nu}_\mu)$ or $\nu_\tau(\bar{\nu}_\tau)$ neutrinos, leading to the collective conversion of neutrino flavors; see Ref. [6] for a recent review and references therein. The conversion rates associated with this mechanism can be as fast as $\omega_{\text{coll}} \propto G_F n_{\nu_e}$, where G_F is the Fermi constant and n_{ν_e} is the local number density of ν_e . For typical neutrino densities ($10^{30} - 10^{35} \text{ cm}^{-3}$) around the neutrino emission surfaces of CCSNe and coalescing NSs, the corresponding $\omega_{\text{coll}} \sim \mathcal{O}(\text{cm}^{-1})$ greatly exceeds the vacuum oscillation rate $\omega_{\text{vac}} = |\Delta m^2|/2E \sim \text{km}^{-1}$, where Δm^2 is the mass-squared difference of the relevant neutrino mass eigenstates and E is the neutrino energy. Various analytical and numerical studies have been carried out to understand the behavior of these “fast” neutrino flavor conversions [7–29]. Besides, several studies on systems such as the NS merger and CCSNe [17,30–40] have inferred that the fast conversions may occur in the most dense regions of them and can potentially affect the dynamics and the nucleosynthesis of these systems.

Because of the highly non-linear structure of the neutrino-neutrino effective interaction Hamiltonian, it is impossible to have a complete analytical solution for the neutrino flavor evolution in environments dense in neutrinos. However, it is possible to linearize the system of equations and carry out the normal mode analysis [8,10,13,41]. Such analyses show that dense neutrino system supports collective flavor runaway modes when certain criteria are satisfied. Although the linear analysis helps us understand the behavior of fast oscillations to some extent, the assumption that the correlations between different flavors are much smaller compared to the self correlation put a very stringent constraint on the applicability of this method. Thus, to study the neutrino oscillations in dense environments to full extent, several numerical simulations have been carried out in recent years. Refs. [15,20,22] studied the neutrino system in a one-dimensional (1D) box in the z -direction with periodic boundaries and translational symmetry in the x and y directions using numerical methods which evolves directly neutrino correlations in discretized grids. Ref. [25] used the spectral method to solve the same set of equation in 1D. Meanwhile, the authors of Ref. [27] adopted the particle-in-cell method and have recently performed simulations for systems with higher spatial dimensions, including both 2D and 3D cases. A study which compares in detail simulation outcome from these groups will be published soon [42].

The intention of this article is to systematically discuss the computational aspects of COSEV (Collective Oscillation Simulation Engine for Neutrinos), the code base developed for simulating the collective neutrino flavor conversions and used in Ref. [22]. COSEV is written completely in C++ to ensure the high performance. At present, it provides two advanced numerical methods for solving the neutrino flavor evolution equations. The first one uses the 4th order finite-difference (central) scheme (FD) sup-

ported by third-order Kreiss-Oliger (KO3) numerical dissipation to treat the advection effectively. The second method adopts the finite-volume (FV) formalism. In this scheme, the flux reconstruction is implemented using the 7th order weighted essentially non-oscillatory (WENO) scheme. For both cases, we have used the 4th order Runge-Kutta method (RK4) for time integration. We shall provide implementation details of both methods in the following sections. In-depth analysis of the physics of the simulation results and their comparison with previous numerical studies for some bench-mark cases was reported in Ref. [22].

This article is organized as follows. In section 2, we briefly discuss the equations governing the neutrino flavor evolution. Section 3 is dedicated to detailed descriptions of the numerical schemes used in COSEV. In section 4, we summarize the results from the advection tests which provide some insights into the numerical error accumulation behavior of both FD and FV implementations. We also carry out tests when the vacuum oscillation is present and compare the results with the analytical solutions. In section 5, we present the results from the simulations when collective neutrino flavor transitions occur. We provide the computational performance comparison in section 6. Discussion and conclusions follow in section 7. Throughout this paper we set $\hbar = c = 1$ to adopt the natural units.

2. Theoretical setup

We considered a simplified set of quantum kinetic equations to describe the transport of neutrinos in a physical system [43,44]. First, we assume for simplicity a two flavor neutrino system in which the single particle flavor state can be represented by a 2×1 column vector $(\nu_e, \nu_x)^T$, where ν_e represents the electron type neutrino and ν_x represent the μ type or τ type neutrino or an appropriate linear combination of them. Then, we use a 2×2 complex valued (Hermitian) density matrix $\varrho(t, \mathbf{x}, \mathbf{v})$ [$\bar{\varrho}(t, \mathbf{x}, \mathbf{v})$] to represent the quantum statistical phase-space distribution of (anti-)neutrinos in our system. The diagonal elements of ϱ , labelled as $\varrho_{ee}(t, \mathbf{x}, \mathbf{v})$ and $\varrho_{xx}(t, \mathbf{x}, \mathbf{v})$ are related to the number densities of e and x type neutrinos respectively for a given velocity mode \mathbf{v} . The off-diagonal component $\varrho_{ex}(t, \mathbf{x}, \mathbf{v})$ is related to the correlation between them. The same definition applies to the antineutrino density matrix $\bar{\varrho}$. For this first version of COSEV, we choose to ignore other potential correlations such as spin coherence and the neutrino-antineutrino correlators [43,44], as well as the momentum-changing collisions of neutrinos [45–48]. We also assume that the system has a perfect translational symmetry in both x and y directions and axial symmetry about the z -axis such that the density matrices depend only on t , z and the z component of the velocity v_z . Then, the space-time evolution of $\varrho(t, z, v_z)$ and $\bar{\varrho}(t, z, v_z)$ are determined by the following equations in 1+1+1(time+space+velocity) dimensions.

$$\frac{\partial}{\partial t} \varrho(t, z, v_z) + v_z \frac{\partial}{\partial z} \varrho(t, z, v_z) = -i[H(t, z, v_z), \varrho(t, z, v_z)], \quad (1a)$$

$$\frac{\partial}{\partial t} \bar{\varrho}(t, z, v_z) + v_z \frac{\partial}{\partial z} \bar{\varrho}(t, z, v_z) = -i[\bar{H}(t, z, v_z), \bar{\varrho}(t, z, v_z)], \quad (1b)$$

where the square bracket $([,])$ stands for the commutation operation. The explicit form for ϱ (for neutrinos) and $\bar{\varrho}$ (anti-neutrinos) are given by,

$$\varrho(t, z, v_z) = \begin{bmatrix} \varrho_{ee} & \varrho_{ex} \\ \varrho_{ex}^* & \varrho_{xx} \end{bmatrix}, \quad \bar{\varrho}(t, z, v_z) = \begin{bmatrix} \bar{\varrho}_{ee} & \bar{\varrho}_{ex} \\ \bar{\varrho}_{ex}^* & \bar{\varrho}_{xx} \end{bmatrix}. \quad (2)$$

The Hamiltonian H (and \bar{H}) in general have contributions stemming from the vacuum oscillations (H_{vac}), neutrino-matter forward scattering (H_{m}) and neutrino-neutrino forward scattering ($H_{\nu\nu}$). In the numerical setup we omit the neutrino-matter forward scattering contribution as it can be removed, assuming H_{m} is space-time independent over the scales determined by $H_{\nu\nu}$, by making appropriate transformations of the density matrices [49,50].¹ In that case the Hamiltonian relevant for our present discussion takes following form in the flavor basis,

$$H(t, z, v_z) = H_{\text{vac}} + \mu \int_{-1}^1 dv'_z (1 - v_z v'_z) \times [\varrho(t, z, v'_z) - \alpha \bar{\varrho}^*(t, z, v'_z)], \quad (3a)$$

$$\bar{H}(t, z, v_z) = \bar{H}_{\text{vac}} - \mu \int_{-1}^1 dv'_z (1 - v_z v'_z) \times [\varrho^*(t, z, v'_z) - \alpha \bar{\varrho}(t, z, v'_z)], \quad (3b)$$

where

$$H_{\text{vac}} = \bar{H}_{\text{vac}} = \frac{\omega_{\text{vac}}}{2} \begin{bmatrix} -\cos 2\theta_{\text{vac}} & \sin 2\theta_{\text{vac}} \\ \sin 2\theta_{\text{vac}} & \cos 2\theta_{\text{vac}} \end{bmatrix}$$

is the vacuum mixing Hamiltonian with θ_{vac} the mixing angle. The quantity $\mu = \sqrt{2} G_{\text{F}} n_{\nu_e}^0$ represents the effective strength of $H_{\nu\nu}$, where $n_{\nu_\beta}^0$ denotes the number density of β flavor neutrinos. $\alpha = (n_{\nu_e}^0/n_{\bar{\nu}_e}^0)$ is the initial neutrino-antineutrino asymmetry parameter. Note here that the quantities ϱ and $\bar{\varrho}$ are normalized with respect to the initial number densities of ν_e and $\bar{\nu}_e$ respectively.

In literature, another equivalent way of describing the dynamics of the collective neutrino flavor evolution is via the so-called polarization vectors, \mathbf{P} , defined by

$$\rho = (P_0 \mathcal{I} + \mathbf{P} \cdot \boldsymbol{\sigma})/2, \quad (4)$$

where $\boldsymbol{\sigma}$ are the Pauli matrices. For cases where the number densities of neutrinos and antineutrinos are conserved per phase-space volume, both P_0 and the length of the polarization vector $|\mathbf{P}| = \sqrt{\sum_{i=1}^3 P_i^2}$ are constants of motion. Due to its ease of implementation and also keeping potential future extensions in mind, COSEV is implemented in terms of the density matrix formalism. We make use of the conserved quantities while testing simulation (see Sec. 5).

3. Numerical implementation

In this section we focus on the numerical implementation of COSEV which solves the partial differential equations (PDEs) (1a) and (1b) when supplemented with the Hamiltonian in Eq. (3). The idea in both FD and FV implementations is to divide the spatial domain which extends from z_0 to z_1 into N_z grid points and carry out the time integration using RK4 from an initial time t_0 to a final time t_f in steps of Δt . Then the size of each cell $\Delta z = (z_1 - z_0)/N_z$ is related to Δt through $\Delta t = C_{\text{CFL}}(\Delta z/\max(|v_z|))$, where C_{CFL} is the Courant–Friedrichs–Lewy number. Since the interaction Hamiltonian is velocity dependent and v_z takes values from -1 to 1 , the velocity space is also divided into N_{v_z} discrete points such that the neutrino beams with v_z values ranging between v_j and

$v_j + \Delta v_z$ for some $0 \leq j < N_{v_z}$, are treated as a single beam with v_z value $v_j + \Delta v_z/2$. Then the contribution to the Hamiltonian from the velocity integrals in the Eqs. (3a) and (3b) are carried out using simple Riemann sum.

3.1. Finite difference method

The numerical technique used to solve hyperbolic PDEs in this work is based on the “method of lines” (See Sec. 6.7 of Ref. [51] for instance) where the spatial and temporal discretizations are treated conceptually in a separated style. We discretized the advection term with the 4th order central finite difference scheme. The resulting equations become ordinary differential equations (ODEs) of time and can be solved as the initial-value problem via the explicit, 4th order Runge-Kutta method. For the integration over velocities in the right hand side, the standard trapezoid rule of second-order accuracy is used with the fixed-width, vertex-center grid points over velocities in $[-1, 1]$. We have also used the basic Simpson’s rule of fourth-order accuracy and obtained similar results. The framework of “method of lines” allows us to treat the advection term with other high order methods such as the WENO scheme, which will be discussed in the next section.

To suppress the high-frequency instability arising from the FD approximation of equations (see Sec. 5), we add Kreiss-Oliger dissipation on the right hand side of each evolving variables u , i.e., letting $\partial_t u \rightarrow \partial_t u + Q u$. The general form of the $2r$ -order Kreiss-Oliger dissipation can be expressed as

$$Q = \varepsilon_{\text{ko}} (-1)^r h^{2r-1} (D_+)^r (D_-)^r / 2^{2r}, \quad (5)$$

where D_{\pm} are one-sided FD operators for $\partial/\partial x$, defined by $D_+ \equiv (u_{+1} - u_0)/h$ and $D_- \equiv (u_0 - u_{-1})/h$. With this convention, the central FD of $\partial^2/\partial x^2$ can be written as $D_+ D_- = (u_{+1} - 2u_0 + u_{-1})/h^2$. A sufficiently large dissipation strength ε_{ko} can suppress the instability without destroying the convergence of a symmetric hyperbolic system of equations, as discussed in [51] via the Fourier-based analysis. In our work, we have used $r = 2$ for most numerical experiments.

3.2. Finite volume method

Unlike the FD method where the values on the grids points are evolved, the FV scheme evolves the cell-averaged values. Let us consider the following simple one dimensional hyperbolic equation,

$$\frac{\partial u(t, z)}{\partial t} + \frac{\partial f(u(t, z))}{\partial z} = 0, \quad (6)$$

where $u(t, z)$ represents the quantity that we want to evolve, and $f(u(t, z))$ is the associated flux function. Given the values of the functions u and f at the grid points and assuming we can appropriately interpolate them to a required order of accuracy, we have

$$\frac{d\bar{u}_i(t)}{dt} = -\frac{1}{\Delta z} [f_{i+1/2}(t) - f_{i-1/2}(t)]. \quad (7)$$

In the above expression we have used the definition $\bar{A}_i(t) = (1/\Delta z) \int_{i-1/2}^{i+1/2} dz A(t, z)$ for some function $A(t, z)$. The upper and lower limits of the integral denoted by $i \pm 1/2$ are the representatives of $z_i \pm (\Delta z/2)$, with z_i being the coordinate of the i th grid point. The PDE in Eq. (6) now becomes an ODE in t , which can be solved, given the values on the right hand side of Eq. (7), using any standard ODE solver. Then the problem is down to reconstructing the flux values at the cell boundaries $i \pm 1/2$ to a required order of accuracy. For reconstructing the flux values $f_{i \pm 1/2}$, the FV implementation of COSEV uses WENO scheme [52–54] such that the

¹ This amounts to the redefinition of the frames in the flavor space. Note that under such a transformation, the vacuum Hamiltonian becomes an effective one ($H_{\text{vac}} \rightarrow H_{\text{vac}}^{\text{eff}}$). For simplicity, we omit the superscript “eff” hereafter.

fluxes are 7-th order accurate for a smooth function while they are at least 4-th order accurate for functions with discontinuities. The discussion below closely follows Refs. [52,53].

Given the cell averaged values of the flux function $f(z)$ at n contiguous locations with a uniform separation of Δz , we can construct a polynomial $\hat{f}^{(n)}$ of degree $n-1$ such that the reconstruction of the function $f(z)$ is n -th order accurate. That is,

$$f(z_i) = \hat{f}_i^{(n)} + O(\Delta z^n). \quad (8)$$

This allows us to write down the n -th order approximation of Eq. (7) as,

$$\frac{d}{dt} \bar{u}_i(t) = -\frac{1}{\Delta z} \left[\hat{f}_{i+1/2} - \hat{f}_{i-1/2} \right] + O(\Delta z^n). \quad (9)$$

We have omitted the superscript n in Eq. (9). It can be shown that, given the cell averaged values of the function \bar{f}_i at each cell of the stencil, values of the function at $i \pm 1/2$ can be expressed as a linear combination of the values of \bar{f}_i ,

$$\hat{f}_{i+1/2} = \sum_{j=0}^{n-1} c_{r,j} \bar{f}_{i-r+j}, \quad (10a)$$

$$\hat{f}_{i-1/2} = \sum_{j=0}^{n-1} \tilde{c}_{r,j} \bar{f}_{i-r+j}, \quad (10b)$$

with the following definition of the coefficients c_{rj} (see Ref. [53] for instance),

$$c_{r,j} = \sum_{m=j+1}^k \frac{\sum_{l=0, l \neq m}^k \prod_{q=0, q \neq m, l}^k (r-q+1)}{\prod_{l=0, l \neq m}^k (m-l)}, \quad (11)$$

where r is the shift parameter that can be thought as the measure of the left-shift of the stencil under consideration from the point i . Since we are reconstructing the values of f at the exact locations $i \pm 1/2$, the relation $\tilde{c}_{r,j} = c_{r-1,j}$ holds true. Note here that the definition of $c_{r,j}$ in the Eq. (11) is valid only for a grid with uniform cell size Δz .

The n -th order WENO scheme considers a stencil S containing n grid points in order to approximate the value of flux at $i \pm 1/2$. Then this stencil S is divided into smaller sub-stencils of order k such that $n = 2k - 1$. The purpose of this decomposition is that, the reconstructed values at the given location are n -th order accurate for smooth functions and at least k -th order accurate if there exists a discontinuity. Each sub-stencil of length k can then be labelled using the shift parameter r (see Fig. A.13). As a result, we have

$$\hat{f}_{i \pm 1/2}^{(n)} = \sum_r d_r \hat{f}_{r, i \pm 1/2}^{(k)}. \quad (12)$$

In order to treat the discontinuities appropriately while making the above linear combination [Eq. (12)], WENO uses the weighted averaging of the coefficients d_r . The weight factor for the contribution of the sub-stencil r is estimated with respect to a smoothness index (SI_r), which as the name implies, measures the smoothness of the sub-stencil r . In our implementation, we have used the results from Refs. [55] and [56] to compute the SI_r . Given the values of SI_r , we replace d_r in Eq. (12) with the corresponding weighted coefficients,

$$\hat{f}_{i \pm 1/2}^{(n)} = \sum_r \frac{w_r}{w} \hat{f}_{r, i \pm 1/2}^{(k)}, \quad (13)$$

where $w = \sum_r w_r$ and

$$w_r = \frac{d_r}{(SI_r + \varepsilon)^2}. \quad (14)$$

The ε in Eq. (14) is to avoid the possibility of the denominator becoming zero. A typical value of ε can be $\sim 10^{-6}$. The explicit forms of the quantities $\hat{f}^{(n)}$, $\hat{f}^{(k)}$, d_r , and SI_r for 7th order WENO are provided in the Appendix A. Note that, when SI_r has discontinuity SI_r becomes large resulting in smaller value of w_r which as we can see from the Eq. (13) reduces the contribution of $\hat{f}_{r, i \pm 1/2}^{(k)}$ to $\hat{f}_{i \pm 1/2}^{(n)}$.

4. Tests for advection and vacuum oscillations

Before discussing collective neutrino oscillations in the next section, we use this section to discuss the results of some preliminary tests carried out with COSEV when $H_{\nu\nu}$ is turned off. In Sec. 4.1 we illustrate the results only with pure advection and in Sec. 4.2 we discuss the results when vacuum oscillation term is turned on. In both cases we will show that the simulation results match very well with what expected theoretically. Also, to quantify the numerical error, we define the quantity

$$E_2 = \sqrt{\frac{1}{L} \int dz \varepsilon(z)^2}, \quad (15)$$

where L is the length of spatial domain and $\varepsilon(z) = f_{\text{Exact}}(z) - f_{\text{Sim}}(z)$ with subscripts Exact and Sim denoting results from simulations and from the analytical solutions. In the following discussions we use FD and FV as the abbreviations for the fourth order finite-difference method with third order KO dissipation and finite volume method with seventh order WENO flux reconstruction schemes, respectively. Note that we adopt the periodic boundary condition in z for all the simulations discussed later.

4.1. Advection

To test the numerical implementation of the advection term, we set the right hand side of the Eq. (1) to zero. The resulting PDE has the solution of the form $f(z - v(t - t_0))$ for an initial profile $f(z, t_0)$ when the advection velocity is set to v . To illustrate the behavior of the advection implementations, we consider two different initial profiles $f(z, t_0)$: a unit Gaussian to represent continuous profiles and a boxcar profile of unit height representing profiles with discontinuities.

Fig. 1 shows the results of the advection tests with the Gaussian profile of unit amplitude initially centered at $z = 0$. We have chosen $N_z = 2000$, $C_{\text{CFL}} = 0.2$ and $v = 0.5$ mode. The panel (a) shows the exact solution and the results from both FV and FD at the end of the simulation at $t = 600$ while panel (b) shows the corresponding errors. Here we see that both implementations of the advection are capable of producing the expected profile with errors at the order of $\mathcal{O}(10^{-8})$. The panel (c) demonstrates the error accumulation behavior of FD and FV. It is evident that even though the magnitudes of the errors remain small, FD accumulates the errors faster compared to FV. The fact that the error remains almost constant in FV reflects the conservative nature of the FV scheme.

In panels (a) and (b) of Fig. 2, we show the E_2 errors and the corresponding numerical order of convergence (see Appendix C for more details) respectively for different values of N_z . Blue colored lines are used to indicate the results from FV while red colored lines (dotted and solid lines for results with and without Kriess-Oliger dissipation respectively) are used for FD. As can be observed from Fig. 2, the errors in FD and FV reduce with increasing value of N_z . The panel (b) shows that both implementations asymptotically reach the expected order of accuracy (7 for 7th order WENO

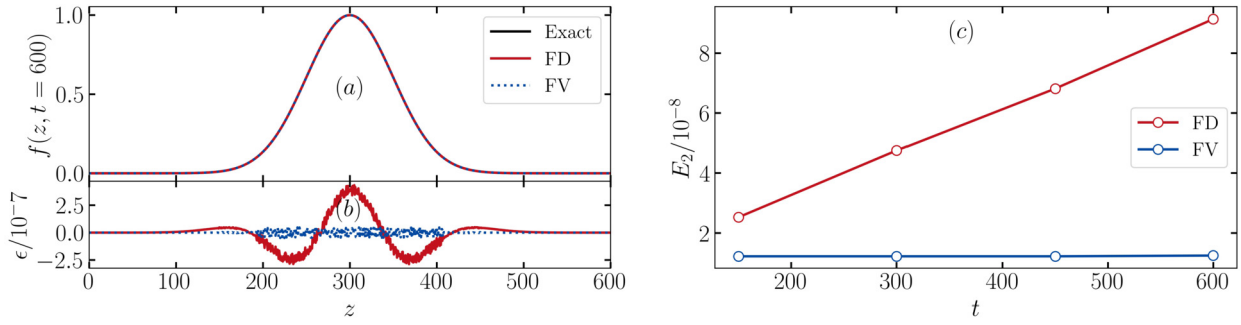


Fig. 1. Results of the advection tests for a traveling Gaussian packet initially centered at $z = 0$ with velocity $v_z = 0.5$. Panel (a): numerical solutions obtained with FV and FD schemes as well as the exact solution at $t = 600$. Panel (b): the spatial distribution of the error $\epsilon(z)$ at the same $t = 600$ as in panel (a). Panel (c): the error indicator E_2 (see text for definition) obtained with FV and FD at different times during the evolution. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

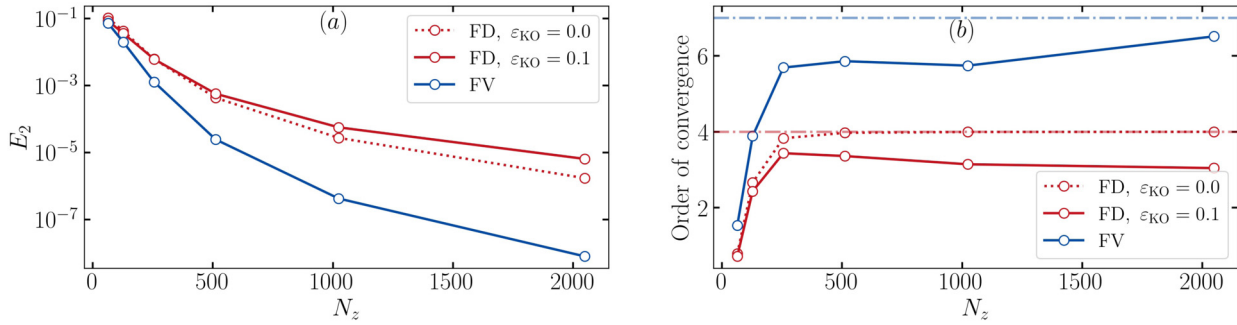


Fig. 2. E_2 [panel (a)] at $t=600$ and the corresponding convergence rate [panel (b)] from advection tests for different values of N_z . Blue and red lines are used to represent the results obtained with FV and FD schemes, respectively. The dotted and solid red lines distinguish the results with and without the Kriess-Oliger dissipation. The light dot-dashed blue and red lines in panel (b) represent the theoretical asymptotic values for the order of convergence.

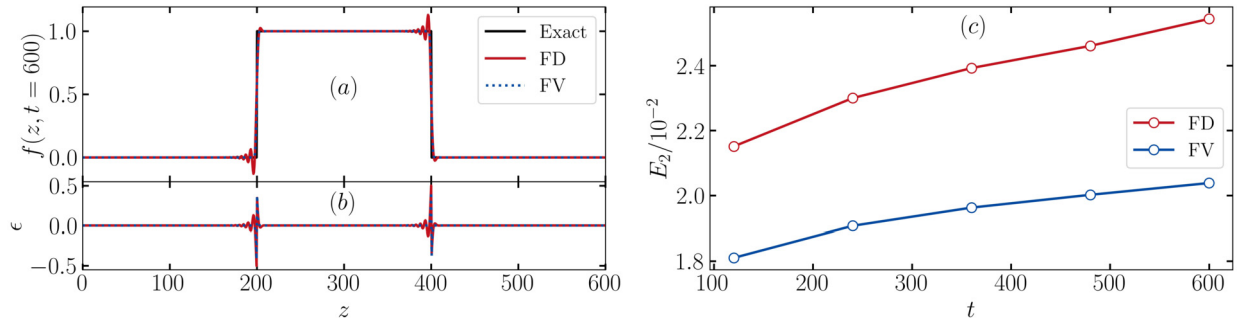


Fig. 3. Results of advection tests with a box centered initially at $z = 0$ with $v_z = 0.5$. Panel (a)–(c) show the same quantities as in Fig. 1.

and 4 for 4th order FD for a smooth function). Note that the numerical accuracy gets reduced when we include the KO dissipation for FD (for the tests we have used KO3 with $\epsilon_{KO} = 0.1$). However, this sacrifice in the numerical accuracy is compensated by the numerical stability as will be illustrated in the section 5.

Next, we demonstrate the nature of the COSEV implementation when some part of the domain contains discontinuity. For this purpose, we have chosen an initial box car profile,

$$f(z, t = 0) = \begin{cases} 1 & \text{for } -100 \leq z \leq 100, \\ 0 & \text{otherwise.} \end{cases} \quad (16)$$

The results obtained are shown in Fig. 3. The sub-panel (a) shows the comparison of the results from FD and FV with the exact solution. The sub-panel (b) shows the nature of error accumulation of both FD and FV. Unlike the Gaussian profile, the maximum values of the error are ~ 0.5 throughout the evolution for both

cases. This is pertained to the slight increase in the width at the base and a slight decrease in the width at the top of the profile obtained from the simulation compared to the exact one. Furthermore, the FD scheme produces small oscillation errors across the discontinuities due to the Gibbs phenomenon [51]. When increasing the resolution, we find that the maximum value of the errors in both FD and FV across the discontinuities remains unchanged. However, the E_2 error can be reduced with larger N_z .

We additionally compute the Fourier transformed spectra for all cases discussed above. In Fig. 4, panel (a) compares the discrete Fourier spectrum of the exact solution with the same of the simulation result for an initial Gaussian profile. Similarly panel (b) illustrates those for an initial box profile. As evident from the figure, the simulations produce nearly identical Fourier spectrum as the exact one when the Gaussian profile is used. For cases with the box profile, the Fourier spectra at the low frequencies compare well with the exact one for both the FD and FV schemes. Both

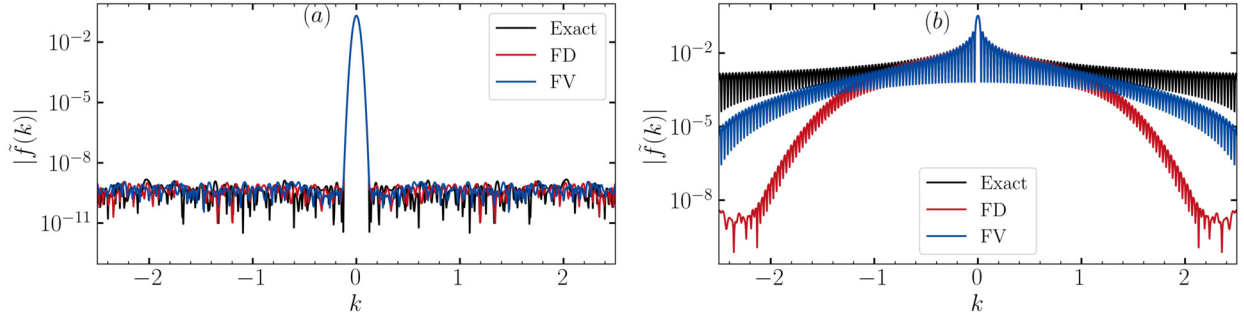


Fig. 4. Comparison of the discrete Fourier spectrum of the solutions obtained using FD and FV simulations with that of the exact solutions at $t = 600$. Panel (a): cases with initial Gaussian profile shown in Fig. 1. Panel (b): cases with initial box car profile shown in Fig. 3.

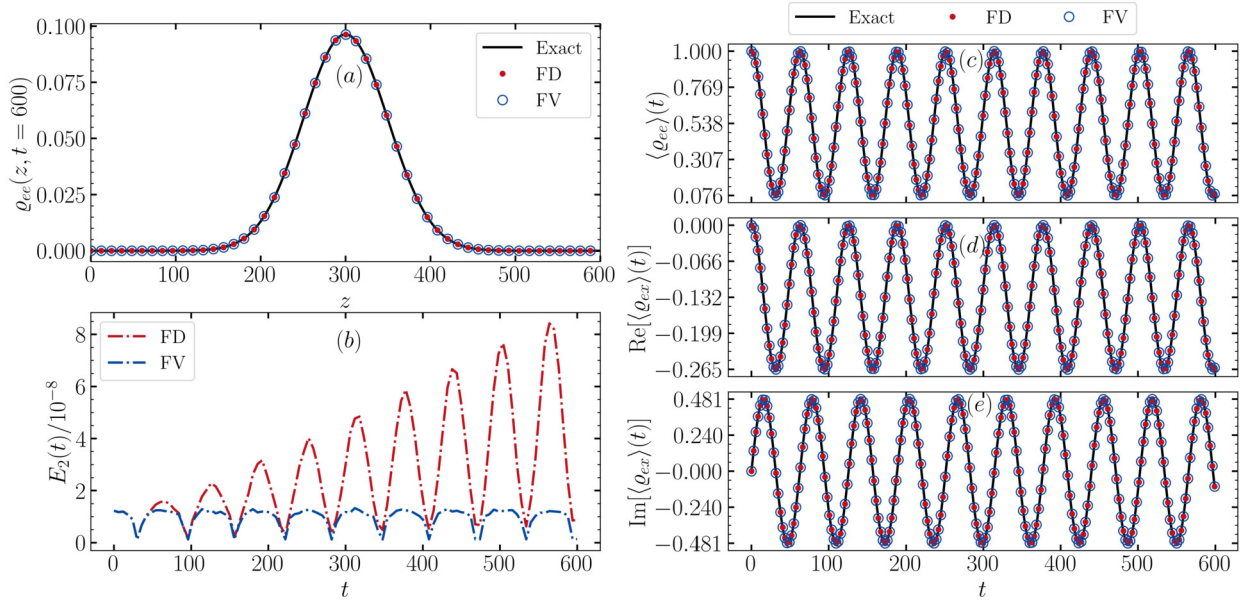


Fig. 5. Results from the vacuum oscillation tests (see text for details). Panel (a): comparison between the simulation and the exact solutions for Q_{ee} . Panel (b): the error accumulation behavior for simulations with FD and FV schemes. Panel (c)–(e): the time evolution of the quantities $\langle Q_{ee} \rangle$, $\text{Re}[\langle Q_{ex} \rangle]$ and $\text{Im}[\langle Q_{ex} \rangle]$.

implementations clearly suppress the high frequency part of the Fourier spectrum due to the slight deviation from perfect discontinuities at the edges of the box. Moreover, the FD scheme here shows further suppression of the high $|k|$ power when compared to the FV scheme, as a result of the KO dissipation discussed in the section 3.1. A comparison of the advection test results with and without the KO dissipation for FD is shown in Appendix B.

4.2. Vacuum oscillations

The second set of tests that we do for COSEV is to consider the cases including both the advection and the vacuum oscillations of neutrinos. Since the vacuum Hamiltonian H_{vac} is independent of ϱ , this allows us to obtain an analytical solution for the space-time evolution of $\varrho(t, z)$. For a pure initial state consisting of only electron neutrinos, the analytical solution of the components of density matrix under the action of H_{vac} takes the following form,

$$Q_{ee}(t, z) = Q_{ee}^{t_0}(z - v(t - t_0)) \times \left[1 - \sin^2(2\theta_{\text{vac}}) \sin^2\left(\frac{\omega_{\text{vac}}}{2}(t - t_0)\right) \right], \quad (17a)$$

$$Q_{xx}(t, z) = Q_{xx}^{t_0}(z - v(t - t_0)) \times \left[\sin^2(2\theta_{\text{vac}}) \sin^2\left(\frac{\omega_{\text{vac}}}{2}(t - t_0)\right) \right], \quad (17b)$$

$$\text{Re}[Q_{ex}(t, z)] = -Q_{ee}^{t_0}(z - v(t - t_0)) \times \left[\sin(2\theta_{\text{vac}}) \cos(2\theta_{\text{vac}}) \sin^2\left(\frac{\omega_{\text{vac}}}{2}(t - t_0)\right) \right], \quad (17c)$$

$$\text{Im}[Q_{ex}(t, z)] = Q_{ee}^{t_0}(z - v(t - t_0)) \times \frac{1}{2} \left[\sin(2\theta_{\text{vac}}) \sin(\omega_{\text{vac}}(t - t_0)) \right], \quad (17d)$$

where $Q_{ab}^{t_0}(z)$ is the value of the of density matrix components at $t = t_0$ for the matrix component ab . Note that here we are only interested in testing if the simulation results match with the exact solutions given in the Eqs. (17). For this purpose, we consider a wave packet of pure electron neutrinos with a Gaussian profile initially centered at $z = 0$. The entire wave packet travels with $v_z = 0.5$ in the z -direction. We set ω_{vac} to 0.1 and θ_{vac} to 37 degrees and choose $N_z = 2000$ and $C_{\text{CFL}} = 0.2$.

Fig. 5 shows the results from vacuum oscillation tests. Panel (a) plots the spatial profiles of Q_{ee} obtained from simulations with FD and FV on top of the analytical solution. Panel (b) shows the temporal evolution of the E_2 error associated with the same for both FD and FV. Similar to earlier results from the advection tests, E_2 remains at the level of $\mathcal{O}(10^{-8})$ for both FD and FV schemes. Once again, the error grows faster in the FD case than FV. The envelope of the error for the latter remains almost constant, also due to the conservative nature of the FV scheme. The oscillatory be-

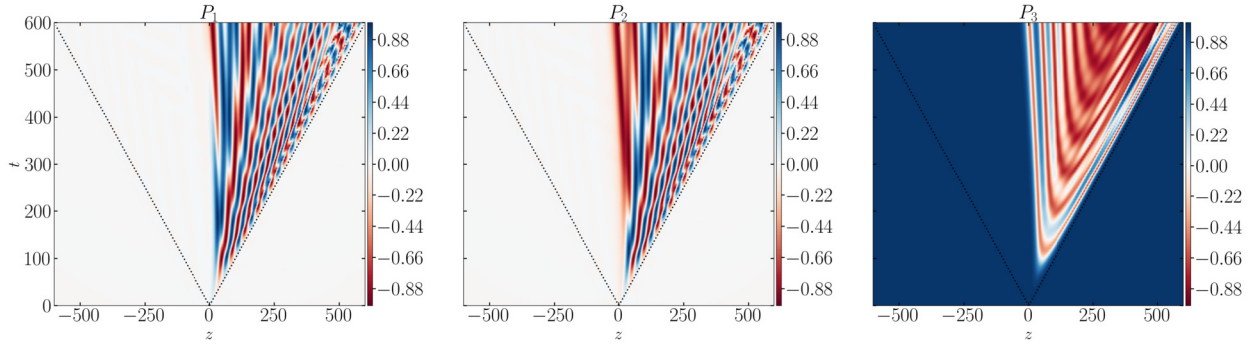


Fig. 6. Evolution of the components of the polarization vector with $v_z = 0.99$ for the fiducial case discussed in the sub-section 5 using $N_z = 2000$, $N_{v_z} = 100$ and $C_{\text{CFL}} = 0.2$. The diagonal dotted lines are used to indicate the light cone.

havior of the error accumulation shown in the figure is related to the oscillatory behavior of $\langle Q_{ee} \rangle(t)$: when $\langle Q_{ee} \rangle(t)$ approaches possible minimum value (0.076 for our choice of vacuum oscillation parameters), the profile $Q_{ee}(z, t)$ becomes smoother, resulting in relatively lower truncation error compared to the profile when $\langle Q_{ee} \rangle(t) = 1$. Note that, here we have used the notation $\langle Q_{ab} \rangle(t)$ to indicate $\int Q_{ab}(z, t) dz / \int Q_{ee}^0(z) dz$. For clarity, we show $\langle Q_{ee} \rangle(t)$, $\text{Re}[\langle Q_{ex} \rangle(t)]$ and $\text{Im}[\langle Q_{ex} \rangle(t)]$ respectively in panels (c), (d) and (e). With this definition, what is shown in panel (c) corresponds to the time evolution of the survival probability of neutrinos. Once again, the results obtained here clearly match the exact solution well.

5. Collective oscillations

In the previous two sections, we have performed tests for advection and vacuum oscillations. We now examine the results obtained from the simulations when collective oscillations occur. We switch off the vacuum oscillations and set $\mu = 1$ so that t and z are expressed in terms of μ^{-1} hereafter. The nonlinear and multi-angle nature of the $\nu\nu$ interaction makes the evolution of collective neutrino flavor conversions complicated. As a result, we cannot compare our simulation results to analytic solutions like in previous sections. Instead, we consider the deviation of several conserved quantities from their initial values to quantify the numerical errors.

Following the convention used in Refs. [15,22], we rewrite the density matrix ϱ in the following way,

$$\varrho(t, z, v_z) \rightarrow g_\nu(v_z)\rho(t, z, v_z), \quad (18a)$$

and

$$\bar{\varrho}(t, z, v_z) \rightarrow g_{\bar{\nu}}(v_z)\bar{\rho}(t, z, v_z), \quad (18b)$$

where $g_{\nu(\bar{\nu})}(v_z) = \frac{1}{4\pi^2 n_{\nu(\bar{\nu})}} \int E^2 dE f_{\nu(\bar{\nu})}$ is the normalized neutrino (anti-neutrino) angular distribution function, i.e., $\int_{-1}^1 g_{\nu(\bar{\nu})}(v_z) dv_z = 1$. In this notation the angular distribution of electron lepton number (ELN) of the neutrinos takes the form $G(v_z) = g_\nu(v_z) - \alpha g_{\bar{\nu}}(v_z)$, where $\alpha = n_{\bar{\nu}_e}/n_{\nu_e}$. When the neutrino and antineutrino number densities are homogeneous in z , the length of the polarization vector $|\mathbf{P}| = 1$ is conserved for any given z and v_z (\mathbf{P} is defined by the decomposition $\rho = (P_0 \mathcal{I} + \mathbf{P} \cdot \boldsymbol{\sigma})/2$). Furthermore, by virtue of the nature of the interaction, we can also show that the quantity

$$\mathbf{M}_0(t) = \int dz \int dv_z n_{\nu_e}(z, t) G(v_z) \mathbf{P}(t, z, v_z), \quad (19)$$

whose third component corresponds to the net neutrino lepton number, is also conserved when periodic boundary condition is considered [16].

We have chosen the same angular distribution functions

$$g_{\nu(\bar{\nu})}(v_z) \propto \exp[-(v_z - 1)^2/2\sigma_{\nu(\bar{\nu})}^2], \quad (20)$$

as in Refs. [15,22]. The corresponding neutrino ELN distributions for different values of α can be found in Ref. [22]. For following discussion we consider $\alpha = 0.9$ as our fiducial value for the asymmetry parameter. We also use the values $\sigma_\nu = 0.6$ and $\sigma_{\bar{\nu}} = 0.5$ such that the $\bar{\nu}$ velocity distribution is more forward-peaked than that of ν . Note that in the absence of the vacuum oscillation term, this configuration does not spontaneously produce any flavor transitions. Thus, one needs to provide initial perturbations to the off-diagonal components of the density matrix. In our tests, the perturbations are supplied in the following manner,

$$\rho_{ee}(z, v_z) = \left(1 + \sqrt{1 - \epsilon^2(z)}\right)/2 = \bar{\rho}_{ee}(z, v_z), \quad (21a)$$

$$\rho_{xx}(z, v_z) = \left(1 - \sqrt{1 - \epsilon^2(z)}\right)/2 = \bar{\rho}_{xx}(z, v_z), \quad (21b)$$

$$\rho_{ex}(z, v_z) = \epsilon(z)/2 = \bar{\rho}_{ex}(z, v_z), \quad (21c)$$

where $\epsilon(z) = 10^{-2} \exp[-z^2/50]$. We then impose the periodic boundary condition in z and evolve the system with COSEV.

In Fig. 6, we show the time evolution of the components of the polarization vector with $v_z = 0.99$ for the above mentioned fiducial case. For this simulation we have chosen the parameters $N_z = 2000$, $N_{v_z} = 100$ and $C_{\text{CFL}} = 0.2$. We can see that collective neutrino oscillations, signified by the change of P_i from their initial values, occur close to the center of the simulation domain where the initial perturbation is maximal. Flavor waves are produced and propagate primarily toward the positive z direction, but never cross the light cone represented by the dotted lines.

Both FV and FD with Kriess-Oliger dissipation produces nearly identical results. In order to emphasize the importance of the KO dissipation scheme in FD, we additionally show in Fig. 7 a comparison of P_3 with $v_z = 0.99$ obtained with and without Kriess-Oliger dissipation scheme. As can be seen from the left panel, the FD simulation without dissipation leads to numerical instabilities starting at $t \simeq 800$, indicated by the back-propagating pattern which violates the causality and produces unphysically large values of $P_3 > 1$.

We now examine the numerical error accumulation in simulations for collective oscillations. We first check the maximum (δP_{max}) and the average ($\langle \delta P \rangle$) of the deviation of the norm of the polarization vector from unity (δP). These quantities are defined as follows,

$$\delta P_{\text{max}} = \max(|\delta P_\nu|_{\text{max}}, |\delta P_{\bar{\nu}}|_{\text{max}}), \quad (22)$$

$$\langle \delta P \rangle = \frac{\int dz \int dv_z |\delta P_\nu| g_\nu(v_z)}{\int dz \int dv_z g_\nu(v_z)}, \quad (23)$$

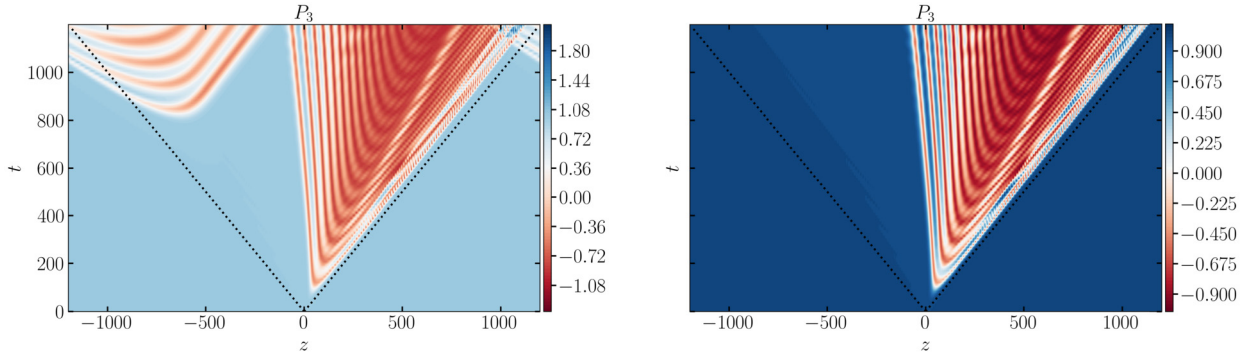


Fig. 7. Comparison of FD results without (left panel) and with (right panel) the Kreiss-Oliger dissipation scheme for the evolution of P_3 . The dotted lines are used to indicate the light cone.

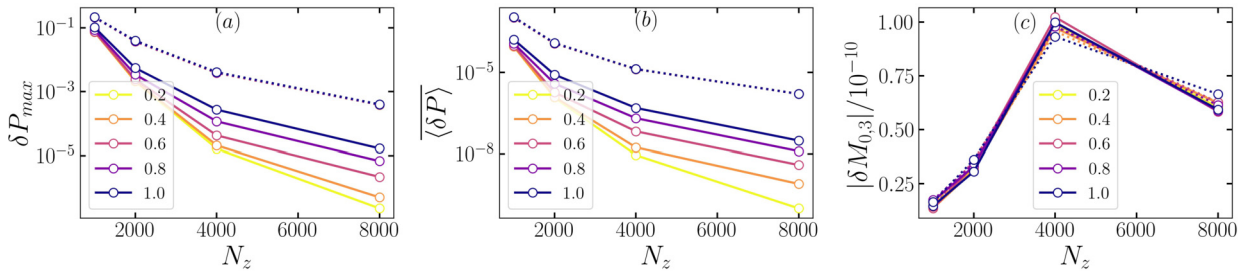


Fig. 8. Error of the conserved quantities in COSEV simulations for different values of N_z and C_{CFL} . Panel (a) shows the δP_{max} , panel (b) shows the $\overline{\langle \delta P \rangle}$ and panel (c) shows the deviation of the third component of $\delta \mathbf{M}_0$. Solid and dotted lines are for simulations with FV and FD schemes respectively. Different colors are used for results with different values of C_{CFL} . Note here that the dotted lines for different values of C_{CFL} are indistinguishable in the sub-panels (a) and (b) as they are on top of each other.

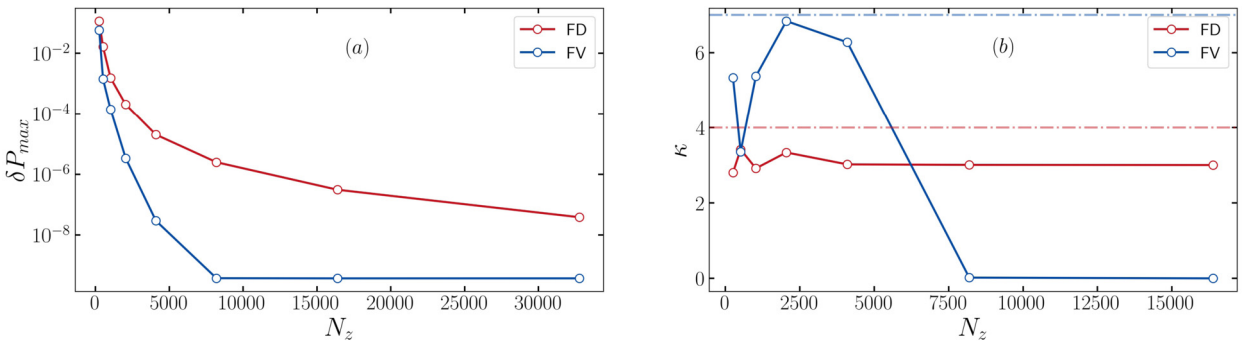


Fig. 9. Values of δP_{max} [panel (a)] and κ [panel (b)] obtained from COSEV simulations for different values of N_z while keeping dt fixed. The light dot-dashed blue and red lines in panel (b) represent the expected asymptotic values for the order of convergence.

where $|\delta P_\nu|_{\text{max}}$ and $|\delta P_{\bar{\nu}}|_{\text{max}}$ are the maximum values of $|\delta P|$ among all velocity modes for ν and $\bar{\nu}$ respectively. We then carry out the following two studies to understand the error accumulations from the spatial derivative and time integration separately. In the first one we vary the value of C_{CFL} for each value of N_z and in the second case we keep Δt constant while varying N_z . The panels (a) and (b) of Fig. 8 show the results for the first case, where we plotted the errors obtained at the end of the tests. The panel (c) shows the deviation of the third component of \mathbf{M}_0 from its initial value. We have used dotted and solid lines to indicate the results from FD and FV respectively while different colors are used to indicate results obtained using different values of C_{CFL} . From panels (a) and (b), we can see that doubling the resolution leads to considerable improvements in the errors for both FD and FV. However, Changing the value of C_{CFL} alone while keeping N_z unchanged (equivalent to changing only Δt) does not affect the errors in the case of FD whereas the same

improves the numerical accuracy of FV. This implies that in the case of FD, the major contribution to the error arises from the treatment of the spatial derivative while in the case of FV the error from the spatial part is similar to or slightly less than that from the time integration. To further illustrate this we consider the second case in which we keep Δt constant and choose a set of N_z and C_{CFL} as explained in the Appendix C. Δt here is chosen such that $N_z = 32768$ has $C_{\text{CFL}} = 1.0$. Fig. 9(a) shows the values of δP_{max} at the end of 1000 iterations from this test. For FD, the error decreases monotonically as we increase the value of N_z . For FV, however, the error decreases until a saturation value $N_z^{\text{sat}} (= 8962 \text{ here})$, after which it stays constant. The reason for this is that when N_z reaches N_z^{sat} , the error from time integration becomes similar in magnitude as that from the spatial part. For N_z larger than the saturation value, the predominant contribution to the error [see Eq. (C.2)] comes from the time integration, which is a constant since Δt is a constant here. This is more

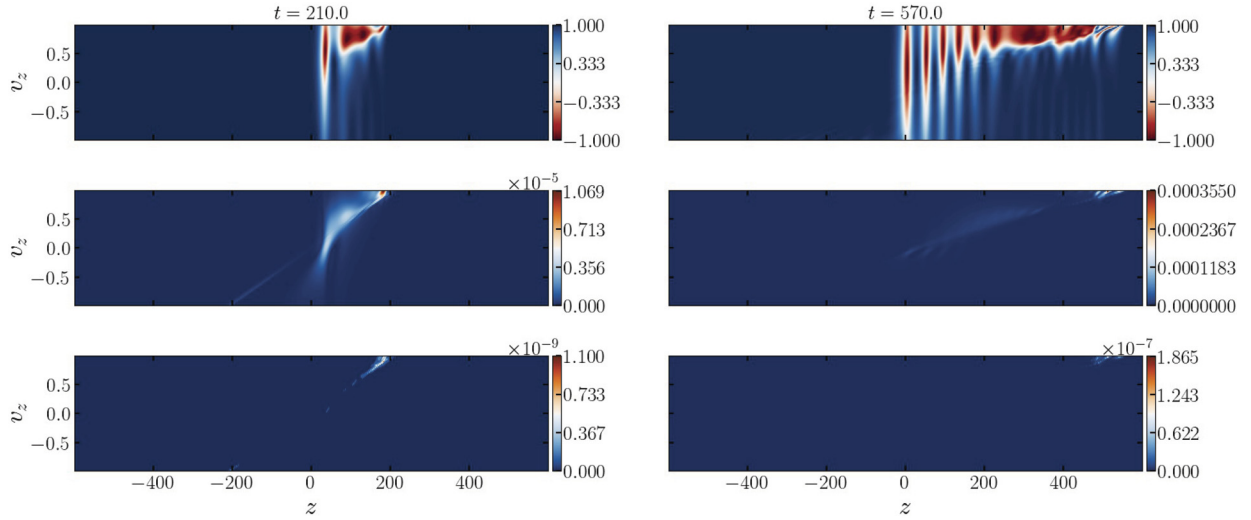


Fig. 10. $P_3(v_z, z)$ (top panels) at different times and the corresponding errors of $|P|$ with FD [middle] and FV [bottom] schemes.

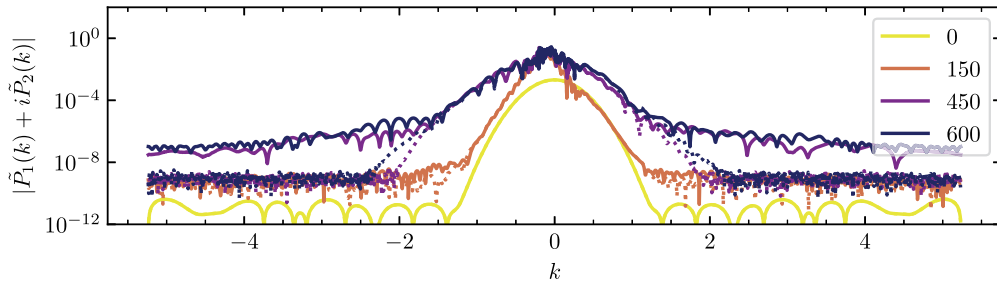


Fig. 11. The Fourier spectrum computed using the perpendicular components of the polarization vector at different times indicated by different colors obtained with the FD (dotted lines) and FV (solid lines) schemes.

evident from Fig. 9(b) where we show the quantity κ , defined as

$$\kappa = \frac{\log[\delta P_{\max}(N_{z,i})/\delta P_{\max}(N_{z,i+1})]}{\log(2)}, \quad (24)$$

where $N_{z,i}$ refers to the i th N_z value adopted in this set of calculations with $N_{z,i+1} = 2N_{z,i}$. Since we kept Δt constant, κ corresponds to the order of numerical convergence when the error from time integration is negligible compared to the numerical error from the spatial derivative. If the situation is the other way around, κ approaches zero. Thus, the approximate constant value of κ for FD indicates the domination of the error from spatial derivative in the total error. Note here that κ assumes similar value that we obtained for the order of accuracy in the Sec. 4.1. On the other hand, κ for FV initially increases with the increase in N_z and then quickly approaches zero as a result of the aforementioned reasons.

From our simulations, we also observe that the δP_{\max} is associated with the collective mode which propagates with maximum speed. This is because damping occurs most severely at the spatial regions where the smoothness is the lowest (see also the previous advection tests in Sec. 4.1). This is illustrated in Fig. 10 where the error distribution as a function of z and v_z is plotted. The top panels show $P_3(z, v_z)$ at two different times and serve as a reference to locate the collective wave fronts. The middle and the bottom panels show the deviation $\delta P(z, v_z)$ obtained from FD and FV, respectively. This plot clearly shows that the errors are larger at the edge of the wave fronts with largest values occurring at $v_z \simeq 1$ as discussed above.

Finally, in Fig. 11 we show the time evolution of the Fourier spectra for $v_z = 0.99$ obtained from our fiducial simulation ($N_z = 2000$ and $C_{\text{CFL}} = 0.2$) with FD (dotted lines) and FV (solid lines). Both FD and FV produce similar Fourier spectra in the range of $-1 \leq k \leq 1$. For larger values of $|k|$, we see that the FV scheme results in faster rise in spectrum when compared to the case with FD. Nevertheless, we confirm that this difference does not lead to any discernible impact on other quantities examined here or in Ref. [22]. We note that the reason for causing this difference likely differs from that discussed in Sec. 4.1 for advection tests, but we do not pursue the exact reason in this paper and leave it for future work.

6. Performance comparison

In the previous sections we have discussed the implementation details and the numerical behavior of COSEV. We now briefly comment on the parallelization strategy as well as the computational performance of the code. COSEV exploits multi-core and GPU accelerations with the help of the directive based parallelization provided by OpenMP [57] and OpenACC [58], respectively, with a single source code. For OpenMP, we extensively used `parallel for collapse(n)` clause for every n -level tightly nested loop without data dependency, which is most of our case. The `reduction` clause is appended to the parallel directives whenever necessary (mostly while computing integrals and doing the analysis). Similarly in OpenACC, we extensively used `parallel loop collapse(n)` clause for tightly nested loops and `reduction` clause in analysis routines. In the computation of the right-hand side of the Eq. (1), we further exploit the levels of parallelism such

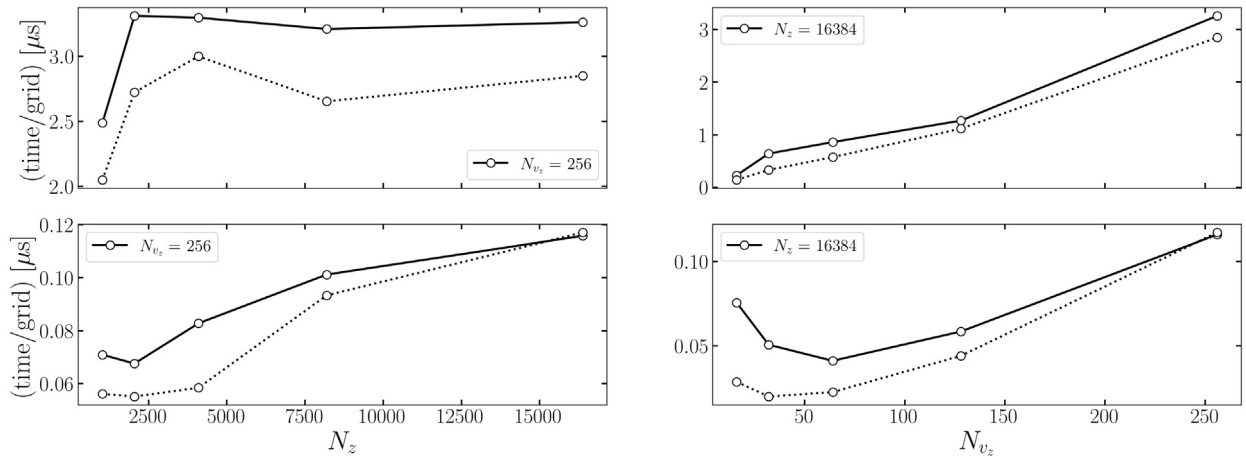


Fig. 12. Comparison of the average computational time taken by COSEv package per grid per iteration for different grid configurations. Left panels display the time per grid when N_{v_z} is kept constant and the right panels show the same when N_z is kept constant. The top and bottom panels show the multi-core and GPU performances respectively. Note that solid and dotted lines are used to differentiate the results with FV and FD respectively and different colors for the lines are used to indicate different values of N_{v_z} .

as `gang`, `worker` and `vector` provided by OpenACC. For data management on GPU, we currently rely on the feature of unified memory allowing a compiler handle the data allocation on the GPU and its movement between the CPU when necessary. Using the minimal set of the compiler directives mentioned above already attain acceptable time-to-solution for the scale of problems in this work and serve as the baseline for future extension. We note that the performance can be optimized by more sophisticated controls on data movement and asynchronous execution, which will be explored in future iterations of COSEv.

Fig. 12 compares the computational time per iteration per grid number ($N_z \times N_{v_z}$) for both FD and FV using different values of N_z and N_{v_z} . The left and right panels show the results obtained with fixed N_{v_z} and N_z , while the top and the bottom panels are with a 20 core Intel@Xeon@E5-2620 processor and NVIDIA P100 GPU, respectively. Solid and dotted lines are used to distinguish the results from FV and FD. Note that here we have switched off all the parts which analyze and output the simulation results in the code, but only considered the computational part, which consumes the major part of the total computing time. First, for nearly all cases the FD version performs slightly better than the FV version on both the GPU and CPU (note that the FV version provides better accuracy as described in previous sections). Second, it is clear that the performance with a single GPU card used here provides much better ($\gtrsim 20$ times faster) than that with a single CPU node. Third, the cases with constant N_{v_z} show nearly perfect scaling when varying N_z in CPU. The scaling is slightly less than ideal in GPU mainly due to tests performed in the under-saturated region of N_z . On the other hand, the computing time per grid per iteration increases nearly linearly as N_{v_z} becomes large, as shown in the right panels of Fig. 12. The linear dependency is consistent with the estimated computing costs of the integration over N_{v_z} in the right-hand side of Eq. (1), which grows linearly over N_{v_z} and is the most time-consuming part of the program.

7. Conclusion and future plans

We have provided details of the numerical implementation in the simulation code, the Collective Simulation Engine for Neutrinos (COSEv), which numerically solves a set of partial differential equations that dictates the dynamics of the collective neutrino flavor conversions in 1+1 dimensions. In-depth details for both the finite difference method supported by the third order Kriess-Oliger dissipation scheme as well as the finite volume method with

seventh order weighted essentially non-oscillatory scheme were discussed. We have also tested the code against advection and vacuum oscillations and shown that COSEv is capable of reproducing the analytical results to a very good precision.

For collective neutrino oscillations, we discussed a fiducial case and demonstrated that adopting the dissipation scheme in the finite difference version of the implementation is essential to prevent the growth of numerical instabilities. The analysis of the conserved quantities showed that COSEv can simulate collective oscillations with very small numerical errors when appropriate spatial and temporal resolutions are chosen. We have also evaluated and provided the performance of COSEv on both CPUs and GPU. The public version of the COSEv package is available at <https://github.com/COSEnu/COSEnu>.

Beyond what was described in this work, we plan to extend COSEv to include other spatial and phase-space dimensions, as well as the collisions between neutrinos and matter. We will also explore other numerical schemes to further suppress the associated errors, and pursue better speed-up with cross nodes and/or multiple GPU cards. All these improvements will be released in future version of COSEv.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgement

M.-R. W. and M. G. acknowledge supports from the National Science and Technology Council, Taiwan under Grant No. 110-2112-M-001-050, No. 111-2628-M-001-003-MY4, and the Academia Sinica under Project No. AS-CDA-109-M11. M.-R. W. also acknowledges supports from the Physics Division, National Center for Theoretical Sciences, Taiwan. M. G. and M.-R. W. appreciate the computing resources provided by the Academia Sinica Grid-computing Center. C.-Y. L. thanks the National Center for High-performance Computing (NCHC) for providing computational and storage resources. Z. X. acknowledge supports from the European

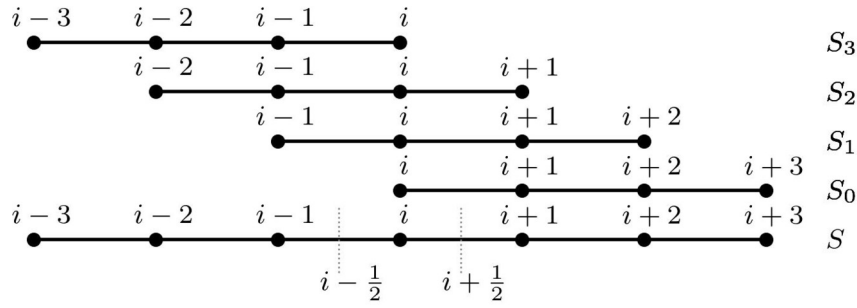


Fig. A.13. The pictorial representation of the stencils used to compute the fluxes at the locations $x_i - \Delta x/2$ and $x_i + \Delta x/2$ (locations of these points on the stencil are represented using vertical dotted lines). The stencil labelled S corresponds to the base stencil. The WENO scheme implemented in the `COSEV` uses the linear combination of 4th order accurate fluxes computed using sub-stencils labelled S_r , $r = 0, 1, 2, 3$.

Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (ERC Advanced Grant KILO-NOVA No. 885281). Authors would also like to thank Taiwan Computing Cloud and NVIDIA for organizing GPU hackathon and also would like to acknowledge the help from Mathew Colgrove and Mason Wu of NVIDIA.

Appendix A. Seventh order WENO-explicit form

In order to reconstruct the flux at $i + 1/2$, we consider a stencil $S = \{i - 3, i - 2, i - 1, i, i + 1, i + 2, i + 3\}$ and divide it into four sub-stencils of consists of four grid points each (see the Fig. A.13 for a pictorial representation of the stencil S and the sub-stencils used for the WENO reconstruction) [53].

$$S_0 = \{i, i + 1, i + 2, i + 3\}, \text{ for } r = 0,$$

$$S_1 = \{i - 1, i, i + 1, i + 2\}, \text{ for } r = 1,$$

$$S_2 = \{i - 2, i - 1, i, i + 1\}, \text{ for } r = 2,$$

$$S_3 = \{i - 3, i - 2, i - 1, i\}, \text{ for } r = 3.$$

Then we have

$$\begin{aligned} \tilde{f}_{i+1/2}^{(7)} = & \left(\frac{-1}{140}\right)\bar{f}_{i-3} + \left(\frac{5}{84}\right)\bar{f}_{i-2} + \left(\frac{-101}{420}\right)\bar{f}_{i-1} + \\ & \left(\frac{319}{420}\right)\bar{f}_i + \left(\frac{107}{210}\right)\bar{f}_{i+1} + \left(\frac{-19}{210}\right)\bar{f}_{i+2} + \left(\frac{1}{105}\right)\bar{f}_{i+3}, \end{aligned} \quad (\text{A.1})$$

and

$$\hat{f}_{r=0,i+1/2}^{(4)} = \left(\frac{3}{12}\right)\bar{f}_i + \left(\frac{13}{12}\right)\bar{f}_{i+1} + \left(\frac{-5}{12}\right)\bar{f}_{i+2} + \left(\frac{1}{12}\right)\bar{f}_{i+3}, \quad (\text{A.2a})$$

$$\hat{f}_{r=1,i+1/2}^{(4)} = \left(\frac{-1}{12}\right)\bar{f}_{i-1} + \left(\frac{7}{12}\right)\bar{f}_i + \left(\frac{7}{12}\right)\bar{f}_{i+1} + \left(\frac{-1}{12}\right)\bar{f}_{i+2}, \quad (\text{A.2b})$$

$$\hat{f}_{r=2,i+1/2}^{(4)} = \left(\frac{1}{12}\right)\bar{f}_{i-2} + \left(\frac{-5}{12}\right)\bar{f}_{i-1} + \left(\frac{13}{12}\right)\bar{f}_i + \left(\frac{3}{12}\right)\bar{f}_{i+1}, \quad (\text{A.2c})$$

$$\hat{f}_{r=3,i+1/2}^{(4)} = \left(\frac{-3}{12}\right)\bar{f}_{i-3} + \left(\frac{13}{12}\right)\bar{f}_{i-2} + \left(\frac{-23}{12}\right)\bar{f}_{i-1} + \left(\frac{25}{12}\right)\bar{f}_i. \quad (\text{A.2d})$$

From Eqs. (12), (A.1), (A.2), we get $d_0 = \frac{1}{35}$, $d_1 = \frac{12}{35}$, $d_2 = \frac{18}{35}$ and $d_3 = \frac{4}{35}$. Using these values of d_r we can construct the weight factors using Eq. (14). Furthermore, for each sub-stencil, one need to estimate the smoothness indices (SI_r). In our implementation we have used the following SI_r [55,56] for $r = 0, 1, 2, 3$.

$$\begin{aligned} SI_0 = & \bar{f}_i(2107\bar{f}_i - 9402\bar{f}_{i+1} + 7042\bar{f}_{i+2} - 1854\bar{f}_{i+3}) \\ & + \bar{f}_{i+1}(11003\bar{f}_{i+1} - 17246\bar{f}_{i+2} + 4642\bar{f}_{i+3}) \\ & + \bar{f}_{i+2}(7043\bar{f}_{i+2} - 3882\bar{f}_{i+3}) + 547\bar{f}_{i+3}^2, \end{aligned} \quad (\text{A.3a})$$

$$\begin{aligned} SI_1 = & \bar{f}_{i-1}(547\bar{f}_{i-1} - 2522\bar{f}_i + 1922\bar{f}_{i+1} - 494\bar{f}_{i+2}) \\ & + \bar{f}_i(3443\bar{f}_i - 5966\bar{f}_{i+1} + 1602\bar{f}_{i+2}) \\ & + \bar{f}_{i+1}(2843\bar{f}_{i+1} - 1642\bar{f}_{i+2}) + 267\bar{f}_{i+2}^2, \end{aligned} \quad (\text{A.3b})$$

$$\begin{aligned} SI_2 = & \bar{f}_{i-2}(267\bar{f}_{i-2} - 1642\bar{f}_{i-1} + 1602\bar{f}_i - 494\bar{f}_{i+1}) \\ & + \bar{f}_{i-1}(2843\bar{f}_{i-1} - 5966\bar{f}_i + 1922\bar{f}_{i+1}) \\ & + \bar{f}_i(3443\bar{f}_i - 2522\bar{f}_{i+1}) + 547\bar{f}_{i+1}^2, \end{aligned} \quad (\text{A.3c})$$

$$\begin{aligned} SI_3 = & \bar{f}_{i-3}(547\bar{f}_{i-3} - 3882\bar{f}_{i-2} + 4642\bar{f}_{i-1} - 1854\bar{f}_i) \\ & + \bar{f}_{i-2}(7043\bar{f}_{i-2} - 17246\bar{f}_{i-1} + 7042\bar{f}_i) \\ & + \bar{f}_{i-1}(11003\bar{f}_{i-1} - 9402\bar{f}_i) + 2107\bar{f}_i^2. \end{aligned} \quad (\text{A.3d})$$

Simple pseudo code for the above described steps for calculating the value of the flux at $x_i + \Delta x/2$ is given in Algorithm 1.

Algorithm 1 Estimation of flux at $i + 1/2$ using 7th order accurate WENO scheme.

- 1: For each spatial grid i assign:
 - $S_0 \leftarrow \{\bar{f}_i, \bar{f}_{i+1}, \bar{f}_{i+2}, \bar{f}_{i+3}\}$
 - $S_1 \leftarrow \{\bar{f}_{i-1}, \bar{f}_i, \bar{f}_{i+1}, \bar{f}_{i+2}\}$
 - $S_2 \leftarrow \{\bar{f}_{i-2}, \bar{f}_{i-1}, \bar{f}_i, \bar{f}_{i+1}\}$
 - $S_3 \leftarrow \{\bar{f}_{i-3}, \bar{f}_{i-2}, \bar{f}_{i-1}, \bar{f}_i\}$
- 2: Compute 4th order accurate values $\hat{f}_{r,i+1/2}^{(4)}$ for each $S_r, r = 0, 1, 2, 3$ using Eq. (A.2)
- 3: Compute smoothness SI_r for each S_r using Eq. (A.3)
- 4: Compute the weight factor w_r using d_r and SI_r using Eq. (14)
- 5: Compute average weight factor $\bar{w}_r = \frac{w_r}{\sum_r w_r}, r = 0, 1, 2, 3$
- 6: Compute the 7th order accurate values at $\hat{f}_{i+1/2}^{(7)}$ using \bar{w}_r and $\hat{f}_{r,i+1/2}^{(4)}$ with Eq. (13)

We can follow the similar procedure to reconstruct the value of flux at $i - 1/2$ as well.

Appendix B. Effect of Kreiss-Oliger dissipation

As discussed in Sec. 3, our implementation of the `COSEV` simulation with the FD method uses the third order KO dissipation scheme to take care of the possible numerical instabilities. To illustrate the advantage of the KO scheme, in Fig. B.14 we show the results and the Fourier transform of the results from the FD simulations with and without KO dissipation.

As shown in the bottom left panel of Fig. B.14, the Fourier power spectrum from the simulation without KO dissipation has the same amplitude as that of the exact solution. However, the phases are only retained relatively well for $|k| \lesssim 1$ but become random for $|k| \gtrsim 1$. This leads to a numerical instability clearly visible in the top left panel. When we apply the KO scheme with $\varepsilon_{KO} = 0.1$, although the phases remain random for $|k| \gtrsim 1$, it clearly suppresses the amplitudes of these modes and thus the corresponding numerical artifact.

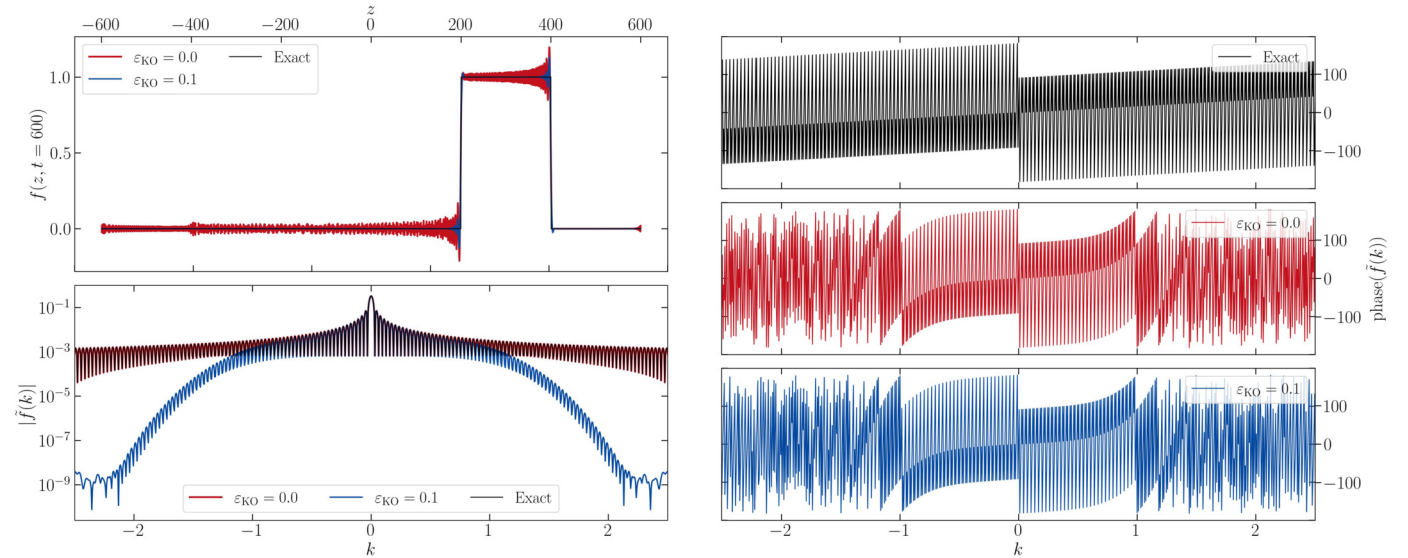


Fig. B.14. Comparison of the effect of KO dissipation scheme. Top-left panel shows results from the pure advection for a box profile with (blue) and without (red) Kreiss-Oliger dissipation. The solid black line represents the exact solution. Bottom-left panel displays the absolute value of the discrete Fourier spectra corresponding to the profiles shown in the top left panel and the right panels show the phase values (in degrees) of the same Fourier spectra.

Appendix C. Numerical estimation of the order of accuracy

The general form of the numerical truncation error, E , can be expressed as,

$$E \simeq C_z \Delta z^\alpha + C_t \Delta t^\beta, \quad (\text{C.1})$$

where C_z and C_t are some constants and α and β are the spatial and temporal order of accuracy respectively. Now, if we have a set $S = \{\Delta z_0, \Delta z_1, \dots, |\Delta z_{i+1}| = \frac{\Delta z_i}{r}, r > 1\}$, then

$$E_i = C_z \left(\frac{\Delta z_0}{r^i}\right)^\alpha + C_t \Delta t_i^\beta. \quad (\text{C.2})$$

If we choose a set of C_{CFL} such that Δt_i is a constant for all values of $\Delta z_i \in S$ then the spatial order of accuracy α can be estimated using

$$\alpha = \frac{\log((E_{i+1} - E_i)/(E_{i+2} - E_{i+1}))}{\log(r)}. \quad (\text{C.3})$$

References

- [1] P. Zyla, et al., PTEP 2020 (8) (2020) 083C01, <https://doi.org/10.1093/ptep/ptaa104>.
- [2] L. Wolfenstein, Phys. Rev. D 20 (1979) 2634–2635, <https://doi.org/10.1103/PhysRevD.20.2634>.
- [3] S. Mikhaev, A. Smirnov, Sov. J. Nucl. Phys. 42 (913) (1985).
- [4] J.T. Pantaleone, Phys. Lett. B 287 (1992) 128–132, [https://doi.org/10.1016/0370-2693\(92\)91887-F](https://doi.org/10.1016/0370-2693(92)91887-F).
- [5] G. Sigl, G. Raffelt, Nucl. Phys. B 406 (1993) 423–451, [https://doi.org/10.1016/0550-3213\(93\)90175-O](https://doi.org/10.1016/0550-3213(93)90175-O).
- [6] F. Capozzi, N. Saviano, Universe 8 (2) (2022) 94, <https://doi.org/10.3390/universe8020094>, arXiv:2202.02494.
- [7] R.F. Sawyer, Phys. Rev. Lett. 116 (2016) 081101, <https://doi.org/10.1103/PhysRevLett.116.081101>.
- [8] I. Izaguirre, G. Raffelt, I. Tamborra, Phys. Rev. Lett. 118 (2017) 021101, <https://doi.org/10.1103/PhysRevLett.118.021101>.
- [9] B. Dasgupta, A. Mirizzi, M. Sen, J. Cosmol. Astropart. Phys. 2017 (02) (2017) 019, <https://doi.org/10.1088/1475-7516/2017/02/019>.
- [10] F. Capozzi, B. Dasgupta, E. Lisi, A. Marrone, A. Mirizzi, Phys. Rev. D 96 (2017) 043016, <https://doi.org/10.1103/PhysRevD.96.043016>.
- [11] S. Abbar, H. Duan, Phys. Rev. D 98 (2018) 043014, <https://doi.org/10.1103/PhysRevD.98.043014>.
- [12] F. Capozzi, G. Raffelt, T. Stirner, J. Cosmol. Astropart. Phys. 2019 (09) (2019) 002, <https://doi.org/10.1088/1475-7516/2019/09/002>.
- [13] C. Yi, L. Ma, J.D. Martin, H. Duan, Phys. Rev. D 99 (2019) 063005, <https://doi.org/10.1103/PhysRevD.99.063005>.
- [14] M. Chakraborty, S. Chakraborty, J. Cosmol. Astropart. Phys. 01 (2020) 005, <https://doi.org/10.1088/1475-7516/2020/01/005>, arXiv:1909.10420.
- [15] J.D. Martin, C. Yi, H. Duan, Phys. Lett. B 800 (2020) 135088, <https://doi.org/10.1016/j.physletb.2019.135088>.
- [16] S. Bhattacharyya, B. Dasgupta, Phys. Rev. D 102 (2020) 063018, <https://doi.org/10.1103/PhysRevD.102.063018>.
- [17] L. Johns, H. Nagakura, G.M. Fuller, A. Burrows, Phys. Rev. D 102 (2020) 103017, <https://doi.org/10.1103/PhysRevD.102.103017>.
- [18] I. Padilla-Gay, S. Shalgar, I. Tamborra, J. Cosmol. Astropart. Phys. 01 (2021) 017, <https://doi.org/10.1088/1475-7516/2021/01/017>, arXiv:2009.01843.
- [19] F. Capozzi, M. Chakraborty, S. Chakraborty, M. Sen, Phys. Rev. Lett. 125 (2020) 251801, <https://doi.org/10.1103/PhysRevLett.125.251801>, arXiv:2005.14204.
- [20] S. Bhattacharyya, B. Dasgupta, Phys. Rev. Lett. 126 (2021) 061302, <https://doi.org/10.1103/PhysRevLett.126.061302>.
- [21] S. Richers, D.E. Willcox, N.M. Ford, A. Myers, Phys. Rev. D 103 (8) (2021) 083013, <https://doi.org/10.1103/PhysRevD.103.083013>, arXiv:2101.02745.
- [22] M.-R. Wu, M. George, C.-Y. Lin, Z. Xiong, Phys. Rev. D 104 (2021) 103003, <https://doi.org/10.1103/PhysRevD.104.103003>.
- [23] T. Morinaga, Fast neutrino flavor instability and neutrino flavor lepton number crossings, arXiv:2103.15267, 3 2021.
- [24] Z. Xiong, Y.-Z. Qian, Phys. Lett. B 820 (2021) 136550, <https://doi.org/10.1016/j.physletb.2021.136550>.
- [25] M. Zaizen, T. Morinaga, Phys. Rev. D 104 (8) (2021) 083035, <https://doi.org/10.1103/PhysRevD.104.083035>, arXiv:2104.10532.
- [26] C. Kato, H. Nagakura, T. Morinaga, Astrophys. J. Suppl. 257 (2) (2021) 55, <https://doi.org/10.3847/1538-4365/ac2aa4>, arXiv:2108.06356.
- [27] S. Richers, D.E. Willcox, N.M. Ford, A. Myers, Phys. Rev. D 103 (2021) 083013, <https://doi.org/10.1103/PhysRevD.103.083013>.
- [28] B. Dasgupta, Phys. Rev. Lett. 128 (8) (2022) 081102, <https://doi.org/10.1103/PhysRevLett.128.081102>, arXiv:2110.00192.
- [29] S. Abbar, F. Capozzi, arXiv:2111.14880, 11 2021.
- [30] M.-R. Wu, I. Tamborra, Phys. Rev. D 95 (2017) 103007, <https://doi.org/10.1103/PhysRevD.95.103007>.
- [31] M.-R. Wu, I. Tamborra, O. Just, H.-T. Janka, Phys. Rev. D 96 (2017) 123015, <https://doi.org/10.1103/PhysRevD.96.123015>.
- [32] T. Morinaga, H. Nagakura, C. Kato, S. Yamada, Phys. Rev. Res. 2 (1) (2020) 012046, <https://doi.org/10.1103/PhysRevResearch.2.012046>, arXiv:1909.13131.
- [33] H. Nagakura, T. Morinaga, C. Kato, S. Yamada, Fast-pairwise collective neutrino oscillations associated with asymmetric neutrino emissions in core-collapse supernova, arXiv:1910.04288, 10 2019, <https://doi.org/10.3847/1538-4357/ab4cf2>.
- [34] M. Delfan Azari, S. Yamada, T. Morinaga, H. Nagakura, S. Furusawa, A. Harada, H. Okawa, W. Iwakami, K. Sumiyoshi, Phys. Rev. D 101 (2) (2020) 023018, <https://doi.org/10.1103/PhysRevD.101.023018>, arXiv:1910.06176.
- [35] S. Abbar, H. Duan, K. Sumiyoshi, T. Takiwaki, M.C. Volpe, Phys. Rev. D 101 (4) (2020) 043016, <https://doi.org/10.1103/PhysRevD.101.043016>, arXiv:1911.01983.
- [36] R. Glas, H.T. Janka, F. Capozzi, M. Sen, B. Dasgupta, A. Mirizzi, G. Sigl, Phys. Rev. D 101 (6) (2020) 063001, <https://doi.org/10.1103/PhysRevD.101.063001>, arXiv:1912.00274.

- [37] Z. Xiong, A. Sieverding, M. Sen, Y.-Z. Qian, *Astrophys. J.* 900 (2) (2020) 144, <https://doi.org/10.3847/1538-4357/abac5e>.
- [38] M. George, M.-R. Wu, I. Tamborra, R. Ardevol-Pulpillo, H.-T. Janka, *Phys. Rev. D* 102 (2020) 103015, <https://doi.org/10.1103/PhysRevD.102.103015>.
- [39] X. Li, D.M. Siegel, *Phys. Rev. Lett.* 126 (2021) 251101, <https://doi.org/10.1103/PhysRevLett.126.251101>.
- [40] H. Nagakura, L. Johns, A. Burrows, G.M. Fuller, *Phys. Rev. D* 104 (8) (2021) 083025, <https://doi.org/10.1103/PhysRevD.104.083025>, arXiv:2108.07281.
- [41] A. Banerjee, A. Dighe, G. Raffelt, *Phys. Rev. D* 84 (2011) 053013, <https://doi.org/10.1103/PhysRevD.84.053013>, arXiv:1107.2308.
- [42] S. Richers, H. Duan, M.-R. Wu, S. Bhattacharyya, M. Zaizen, M. George, C.-Y. Lin, Z. Xiong, *Phys. Rev. D* 106 (4) (2022) 043011, <https://doi.org/10.1103/PhysRevD.106.043011>.
- [43] A. Vlasenko, G.M. Fuller, V. Cirigliano, *Phys. Rev. D* 89 (2014) 105004, <https://doi.org/10.1103/PhysRevD.89.105004>.
- [44] C. Volpe, D. Väänänen, C. Espinoza, *Phys. Rev. D* 87 (2013) 113010, <https://doi.org/10.1103/PhysRevD.87.113010>.
- [45] S. Shalgar, I. Tamborra, *Phys. Rev. D* 103 (2021) 063002, <https://doi.org/10.1103/PhysRevD.103.063002>, arXiv:2011.00004.
- [46] L. Johns, *Collisional flavor instabilities of supernova neutrinos*, arXiv:2104.11369, 4 2021.
- [47] J.D. Martin, J. Carlson, V. Cirigliano, H. Duan, *Phys. Rev. D* 103 (2021) 063001, <https://doi.org/10.1103/PhysRevD.103.063001>, arXiv:2101.01278.
- [48] G. Sigl, *Phys. Rev. D* 105 (4) (2022) 043005, <https://doi.org/10.1103/PhysRevD.105.043005>, arXiv:2109.00091.
- [49] H. Duan, G.M. Fuller, Y.-Z. Qian, *Phys. Rev. D* 74 (2006) 123004, <https://doi.org/10.1103/PhysRevD.74.123004>.
- [50] H. Duan, G.M. Fuller, Y.-Z. Qian, *Annu. Rev. Nucl. Part. Sci.* 60 (2010) 569–594, <https://doi.org/10.1146/annurev.nucl.012809.104524>, arXiv:1001.2799.
- [51] B. Gustafsson, H.-O. Kreiss, J. Oliger, *Time Dependent Problems and Difference Methods*, vol. 24, John Wiley & Sons, 1995.
- [52] X.-D. Liu, S. Osher, T. Chan, *J. Comput. Phys.* 115 (1) (1994) 200–212, <https://doi.org/10.1006/jcph.1994.1187>.
- [53] C.-W. Shu, *Essentially Non-oscillatory and Weighted Essentially Non-oscillatory Schemes for Hyperbolic Conservation Laws*, Springer Berlin Heidelberg, Berlin, Heidelberg, 1998, pp. 325–432.
- [54] C.-W. Shu, *Int. J. Comput. Fluid Dyn.* 17 (2) (2003) 107–118, <https://doi.org/10.1080/1061856031000104851>.
- [55] G.-S. Jiang, C.-W. Shu, *J. Comput. Phys.* 126 (1) (1996) 202–228, <https://doi.org/10.1006/jcph.1996.0130>.
- [56] D.S. Balsara, C.-W. Shu, *J. Comput. Phys.* 160 (2) (2000) 405–452, <https://doi.org/10.1006/jcph.2000.6443>.
- [57] OpenMP, <http://www.openmp.org/>.
- [58] OpenACC, <http://www.openacc.org/>.