

Classical-quantum simulation of non-equilibrium Marshak waves

C.J. Myers¹, Nick Gentile², Hunter Rouillard¹, Ryan Vogt¹,
F. Graziani² and F. Gaitan^{1,†}

¹Laboratory for Physical Science, 8050 Greenmead Dr, College Park, MD 20740, USA

²Lawrence-Livermore National Laboratory, 7000 East Avenue, Livermore, CA 94550, USA

(Received 29 April 2024; revised 22 July 2024; accepted 24 July 2024)

In the radiation hydrodynamic simulations used to design inertial confinement fusion (ICF) and pulsed power experiments, nonlinear radiation diffusion tends to dominate CPU time. This raises the interesting question of whether a quantum algorithm can be found for nonlinear radiation diffusion which provides a quantum speedup. Recently, such a quantum algorithm was introduced based on a quantum algorithm for solving systems of nonlinear partial differential equations (PDEs) which provides a quadratic quantum speedup. Here, we apply this quantum PDE (QPDE) algorithm to the problem of a non-equilibrium Marshak wave propagating through a cold, semi-infinite, optically thick target, where the radiation and matter fields are not assumed to be in local thermodynamic equilibrium. The dynamics is governed by a coupled pair of nonlinear PDEs which are solved using the QPDE algorithm, as well as two standard PDE solvers: (i) Python's py-pde solver; and (ii) the KULL ICF simulation code developed at Lawrence-Livermore National Laboratory. We compare the simulation results obtained using the QPDE algorithm and the standard PDE solvers and find excellent agreement.

Keywords: plasma nonlinear phenomena, fusion plasma

1. Introduction

Inertial confinement fusion (ICF) and pulse power experiments are complex and financially expensive to execute. There is thus a clear need for design work that tests whether an experimental concept will perform as expected. Radiation-hydrodynamics (RH) computer simulation codes are the primary computational tool used for such design work, with the equations of RH (Castor 2004) providing the foundation for all such codes. Constructed for accuracy, robustness and efficiency, they are elaborate, multi-physics and massively parallel codes. Simulation runs can take hours to weeks to complete, depending on the number of spatial dimensions, resolution and physics model chosen, as well as the particular design being tested. The equations of RH are a coupled set of nonlinear partial

† Email address for correspondence: fgaitan@lps.umd.edu

differential equations (PDEs) expressing the conservation of fluid mass, linear momentum and energy, as well as nonlinear PDEs for the radiation and matter fields.

For our purposes, it is the radiation package used in RH codes that is of interest. These packages come in several variants. The first, and most accurate, describes full photon transport coupled to the matter fields. This is the most expensive variant in which the radiation field is non-Planckian, spatially inhomogeneous and anisotropic. It is rarely used since the matter conditions are dense enough that the second and third variants are excellent approximations for ICF capsule implosion and burn. The second variant, and next on the accuracy scale, is multi-group diffusion coupled to matter fields. Here, the radiation is also non-Planckian and spatially inhomogeneous, although now is assumed to be nearly isotropic. Finally, the third variant is the two-temperature (2 T) diffusion approximation where the radiation is assumed to be Planckian, although still inhomogeneous, nearly isotropic and coupled to matter fields. It is most frequently used when simulating the burn stage of an ICF capsule where the plasma is very dense, and has the virtue of allowing relatively fast simulation runs (when compared with the first two variants). Due to its robustness and wide use, we focus on the third variant here.

Given the difficulty of RH simulations, it is natural to ask whether a quantum computer might provide a quantum speedup for such simulations. A first step towards a quantum RH algorithm is already possible as a quantum algorithm for simulating hydrodynamic flows exists which provides a quadratic speedup (Gaitan 2020). This quantum algorithm was soon generalized to a quantum algorithm for solving systems of nonlinear PDEs with a quadratic speedup (Gaitan 2021). Recently, the quantum PDE (QPDE) algorithm was used to simulate nonlinear radiation diffusion (NRD) in the case where the radiation and matter fields are in local thermodynamic equilibrium (LTE) (Gaitan, Graziani & Porter 2024). The algorithm was verified by applying it to a standard NRD test problem – a Marshak wave propagating through a cold, optically thick, semi-infinite slab of matter. A numerical simulation of the QPDE algorithm applied to this problem was carried out and the results found were compared with those produced by a standard PDE solver. Excellent agreement was found.

In this paper we show how the QPDE algorithm can be applied to the 2 T diffusion approximation described above. This approximation is relevant for ICF and pulse power experiments where the radiation and matter fields can go out of LTE. Consequently, although our test problem will again be a Marshak wave, we allow for non-LTE so that the matter and radiation local temperatures need not be equal. The Marshak wave dynamics is now determined by a coupled pair of nonlinear PDEs which describe the exchange of energy between the radiation and matter fields. We numerically simulated application of the QPDE algorithm to the 2 T Marshak wave problem and compared the results found with those obtained using two well-known PDE solvers. The first comparison uses Python's `py-pde` solver (Zwicker 2020) as the standard PDE solver, while the second uses the KULL ICF simulation code developed at Lawrence-Livermore National Laboratory (LLNL) (Rathkopf *et al.* 2000).

We shall see below that the QPDE algorithm is a hybrid classical-quantum algorithm which uses a quantum computer to evaluate definite integrals from which a global PDE solution is constructed. The hybrid nature of this algorithm produces a number of advantages over fully quantum nonlinear PDE algorithms.

- (i) The only data that need to be sent to a quantum computer to execute the QPDE algorithm are the integrand values used to approximate the above definite integrals. How this is done using quantum circuits is explained in Gaitan (2024). These values are computed using a classical computer.

- (ii) As explained in Gaitan (2020, 2021), nonlinear and/or dissipative terms present in a system of PDEs contribute terms to the above mentioned integrand. Such processes only affect its algebraic form. Their presence causes no difficulties when determining the integrand values since they are easily evaluated using a classical computer. Thus the challenges faced by fully QPDE algorithms due to nonlinear and dissipative processes do not arise in our QPDE algorithm.
- (iii) As will be seen below, our QPDE algorithm, at its coarsest level of description, has two steps. The first reduces the nonlinear PDEs to nonlinear ODEs. This step can be done in a number of different ways using standard numerical methods for PDEs. As will be shown in forthcoming papers, the numerical toolbox used by applied mathematicians, computational scientists and engineers to implement stable, high-resolution numerical computations can be straightforwardly incorporated into our QPDE algorithm. These papers show how finite volume, finite element and spectral methods can be incorporated, as well as weighted effectively non-oscillatory methods that can handle PDEs giving rise to solutions with discontinuities such as shock waves. With our QPDE algorithm, PDE subject matter experts are not required to reinvent the wheel to use a quantum computer to produce stable, high-resolution PDE solutions, as their existing powerful toolbox can be straightforwardly incorporated into this algorithm.

The structure of this paper is as follows. In § 2 we briefly summarize the QPDE algorithm and explain how it is applied to a system of nonlinear PDEs. In § 3 we apply the QPDE algorithm to the coupled pair of PDEs driving the 2 T Marshak wave dynamics. We present the results of a numerical simulation of the QPDE algorithm applied to this problem, and compare them with those found using the `py-pde` Python PDE solver. The comparison provides strong evidence that the QPDE algorithm correctly solves the 2 T Marshak wave problem. Specifically, and as expected: (i) the radiation and matter fields are seen to evolve into LTE; and (ii) the equilibrium/1 T Marshak wave (Gaitan *et al.* 2024) is seen to emerge as the large time limit of the non-equilibrium/2 T Marshak wave dynamics. In § 4 we repeat the comparison of § 3, this time using the LLNL/KULL ICF radiation diffusion solver as the standard PDE solver. We again find: (i) clear evidence that the QPDE algorithm correctly solves the 2 T Marshak wave problem; and (ii) that the radiation and matter fields evolve into LTE. Finally, we make closing remarks in § 5.

2. The QPDE algorithm

We briefly describe how the QPDE algorithm is constructed. For a detailed presentation, including a discussion of the Russian doll-like hierarchy of supporting quantum algorithms, see Gaitan (2024).

The task of the QPDE algorithm is to find an approximate, bounded solution of a system of nonlinear PDEs

$$\frac{\partial U}{\partial t} = F[U, U_i, \dots, U_{i_1 \dots i_n}], \quad (2.1)$$

where the exact solution $U(x, t)$ is a d -component vector field defined on a: (i) spatial region D with boundary ∂D ; and (ii) time interval $0 \leq t \leq T$. The function F is assumed to be nonlinear, and depends on U and its spatial partial derivatives up to order n , where $U_i = \partial U / \partial x_i$; \dots ; $U_{i_1 \dots i_n} = \partial^n U / \partial x_{i_1} \dots \partial x_{i_n}$. We hold off on stating the initial and boundary conditions until § 3. The Supporting Information for Gaitan (2021) showed how systems of nonlinear PDEs containing time partial derivatives of order greater than one and/or a non-autonomous function $F[t, U, U_i, \dots, U_{i_1 \dots i_n}]$ can be reduced to the form given in

(2.1). The QPDE algorithm is thus applicable to a large family of systems of nonlinear PDEs which includes those typically encountered in science and engineering applications.

The QPDE algorithm, at its coarsest level of description, consists of two steps. The first step is to discretize space while leaving time as a continuous parameter. Thus the spatial continuum parameterized by x is replaced by a spatial grid containing M uniformly spaced grid points (in the simplest case) parameterized by $x(I) = x_0 + I\Delta$, where $I = (i_1, i_2, i_3)$, $i_k \in \{1, \dots, m_k\}$ ($k = 1, 2, 3$), $M = m_1 m_2 m_3$ and Δ is the spacing between grid points along any coordinate direction. The solution $U(x, t)$ is now restricted to grid points: $U_I(t) \equiv U(x(I), t)$. Absent a spatial continuum, spatial partial derivatives no longer exist and must be approximated. Here, we use a finite difference (FD) approximation (Pletcher, Tannehill & Anderson 2013). This replaces spatial partial derivatives of $U(x, t)$ with algebraic expressions of $\{U_I(t)\}$. Inserting the FD approximations into $F[U, U_{i_1}, \dots, U_{i_1, \dots, i_n}]$ yields an algebraic expression $f_I(U)$ at each grid point $x(I)$. Time t is now the only continuous parameter and so the partial time derivative in (2.1) becomes a total derivative. The result of the spatial discretization is the reduction of the system of nonlinear PDEs to a coupled set of nonlinear ordinary differential equations (ODEs),

$$\frac{dU_I}{dt} = f_I(U), \quad (2.2)$$

with an ODE associated with each grid point $x(I)$ in the interior of D . We explain how the driver function $f_I(U)$ is determined in § 3. The initial and boundary conditions for (2.2) are obtained by evaluating the initial and boundary conditions for (2.1) at the grid points $x(I)$ in the interior of D , and on the boundary ∂D , respectively.

The second step is to use a quantum nonlinear ODE algorithm to solve (2.2). In Gaitan (2020, 2021) a quantum algorithm due to Kacewicz (Kacewicz 2006) was used, although any quantum nonlinear ODE algorithm will do. To flesh out this second step, we briefly describe application of Kacewicz' algorithm to (2.2). To unclutter the notation, we suppress the subscripts on $U_I(t)$ and $f_I(t)$. The algorithm: (i) returns an approximate, bound solution $\alpha(t)$ to the exact solution $U(t)$ of (2.2) over the time interval $0 \leq t \leq T$; (ii) guarantees the error ε in the approximate solution satisfies an upper bound (see (2.4)) with probability $1 - \delta$; and (iii) gives a quadratic speedup over classical nonlinear ODE algorithms.

Kacewicz' algorithm consists of five steps. As we shall see, only the last step requires a quantum computer. The first step partitions the time interval $0 \leq t \leq T$ into n primary subintervals by introducing $n + 1$ uniformly spaced intermediate times $t_0 = 0, \dots, t_i = ih, \dots, t_n = T$, where $h = T/n$. The i th primary subinterval $[t_i, t_{i+1}]$ is denoted T_i , with $i = 0, \dots, n - 1$. Step two partitions each primary subinterval T_i into $N_k = n^{k-1}$ secondary subintervals by introducing $N_k + 1$ uniformly spaced intermediate times $t_{i,0} = t_i, \dots, t_{i,j} = t_i + j\bar{h}, \dots, t_{i,N_k} = t_{i+1}$, where $\bar{h} = h/N_k = T/n^k$. The j th secondary subinterval $[t_{i,j}, t_{i,j+1}]$ in T_i is denoted $T_{i,j}$. The third step associates with each primary subinterval T_i a parameter y_i . The parameter y_0 is set equal to the ODE initial condition, $y_0 \equiv U_0$, while the remaining parameters y_1, \dots, y_{n-1} approximate the exact solution $U(t)$ at the times t_1, \dots, t_{n-1} , respectively (*viz.* $U(t_i) \approx y_i$ for $i \in \{1, \dots, n - 1\}$). Step five will explain how these $n - 1$ parameters are assigned values. Step four uses Taylor's method (Iserles 2009; Gautschi 2012), to approximate the exact solution $U(t)$ in each of the secondary subintervals $T_{i,j}$ using a truncated Taylor series $\alpha_{i,j}(t)$ expanded about the time $t_{i,j}$. The approximate solution $\alpha_i(t)$ in primary subinterval T_i is then defined in terms of the local approximate solutions $\alpha_{i,j}(t)$ so that when $t \in T_{i,j}$, then $\alpha_i(t) = \alpha_{i,j}(t)$. For example, if $t \in T_{i,0}$, then $\alpha_i(t) = \alpha_{i,0}(t)$, and similarly for all other secondary subintervals $T_{i,j}$. Finally, $\alpha_i(t)$ is

required to be continuous throughout T_i . Thus $\alpha_{i,j}(t)$ and $\alpha_{i,j+1}(t)$ must agree at their common boundary time $t_{i,j+1} : \alpha_{i,j}(t_{i,j+1}) = \alpha_{i,j+1}(t_{i,j+1})$. The final condition placed on $\alpha_i(t)$ is that $\alpha_i(t_i) \equiv y_i$. The fifth step derives a relation that allows $\{y_i\}$ to be determined iteratively

$$y_{i+1} = y_i + \int_{t_i}^{t_{i+1}} df(\alpha_i(t)) \quad (0 \leq i \leq n-2). \quad (2.3)$$

Kacewicz' algorithm requires a quantum integration algorithm to evaluate the integral on the right-hand side of (2.3). It is important to appreciate that this is the only task in this quantum algorithm that requires a quantum computer. All other calculations are done on a classical computer. In the work reported below, we use Novak's quantum integration algorithm (Novak 2001) to evaluate the integrals. The procedure for determining the $\{y_i\}$ begins with $y_0 \equiv U_0$. Knowing y_0 determines the approximate solution $\alpha_0(t)$ (for details, see Kacewicz 2006; Gaitan 2020, 2021). Inserting $\alpha_0(t)$ into the ODE driver function $f(\alpha_i(t))$ determines the integrand in (2.3) for $i = 0$. Novak's quantum integration algorithm is used to approximate the integral and the value returned is (classically) added to y_0 to give y_1 . Knowing y_1 gives $\alpha_1(t)$, which is substituted into $f(\alpha_i(t))$ for $i = 1$ and its integral over T_1 is approximated using Novak's algorithm. The value returned is added to y_1 to give y_2 , etc. At the end of the iteration procedure all approximate solutions $\alpha_0(t), \dots, \alpha_{n-1}(t)$ are determined and the approximate ODE solution is $\alpha(t) = \alpha_i(t)$ for $t \in T_i$.

As mentioned briefly above, Kacewicz (2006) showed that for Hölder class driver functions, the error ε in the approximate solution $\alpha(t)$ returned by his algorithm satisfies

$$\varepsilon = \sup_{0 \leq t \leq T} \|U(t) - \alpha(t)\| = \mathcal{O}(1/n^{\alpha_k}), \quad (2.4)$$

with probability $1 - \delta$. Here: (i) n is the number of primary subintervals; (ii) k is related to the number of secondary subintervals in a primary subinterval: $N_k = n^{k-1}$; (iii) $\alpha_k = k(q+1) - 1$, where $q = r + \rho$, $r + 1$ is the number of terms kept in the truncated Taylor series $\alpha_{i,j}(t)$, and ρ is the Hölder exponent. Kacewicz (2006) showed that to achieve this performance, the upper bound ε_1 on the error δI of the value I of the integral (over a secondary subinterval) must be

$$\varepsilon_1 = 1/n^{k-1}, \quad (2.5)$$

and the probability $1 - \delta_1$ that $\delta I < \varepsilon_1$ is

$$1 - \delta_1 = (1 - \delta)^{1/n^k}. \quad (2.6)$$

Finally, an explanation of how the parameters n and k are determined, given the upper bound ε_1 , is given in the Supplementary Information for Gaitan (2020). The essential parameters for the above construction are then ε_1 which controls the time partitioning, and δ which controls the number of times the QPDE algorithm must be rerun to ensure that the bound (2.4) is satisfied with probability $1 - \delta$. See Gaitan (2024) for a detailed presentation of the above remarks. For the simulations results presented below, we chose $\varepsilon_1 = 0.005$ and $\delta = 0.001$.

Kacewicz (2006) showed that his quantum nonlinear ODE algorithm provides a quadratic quantum speedup over classical nonlinear ODE algorithms, and Gaitan (2021) showed that the QPDE algorithm inherits this quadratic speedup.

3. Non-equilibrium Marshak wave problem: first verification

We consider a radiation source at temperature T_0 in contact (through a planar interface) with a cold, semi-infinite, optically thick slab of matter. The radiation undergoes nonlinear diffusion as it penetrates the matter, manifesting as a propagating thermal front. The radiation and matter exchange energy locally so that the radiation cools down and the matter heats up. After a period of time the radiation and matter come into LTE, and from then on propagate as an equilibrium Marshak wave (Marshak 1958). Our focus here is the approach to LTE, and the emergence of the equilibrium Marshak wave.

Following Marshak we assume the diffusion is one-dimensional, and that hydrodynamic effects can be ignored so that the matter remains at rest and at constant density ρ_0 . We ignore scattering, tracking only absorption, so that the Planck and Rosseland absorption coefficients σ_P and σ_R are equal. The opacity κ is

$$\kappa = \sigma_R / \rho = 1 / \rho \lambda_R, \quad (3.1)$$

where λ_R is the Rosseland mean free path. The opacity κ is assumed to have a power-law dependence on matter density ρ and temperature T

$$\kappa = \mathcal{O}_\rho \mathcal{O}_T \rho^{X_\rho^0} T^{-\gamma}. \quad (3.2)$$

Although we choose values for \mathcal{O}_ρ , \mathcal{O}_T , X_ρ^0 and γ in § 4, for purposes of this section, only γ is needed. The results presented in this section use $\gamma = 3.5$.

3.1. Governing system of nonlinear PDEs

The governing nonlinear PDEs for this problem are the radiation and matter energy balance equations (Pomraning 1979; Graziani 2005)

$$\frac{\partial \varepsilon_R}{\partial t} - \frac{\partial}{\partial x} \left[\frac{c}{3} \lambda_R \frac{\partial \varepsilon_R}{\partial x} \right] = \frac{c}{\lambda_R} [aT^4 - \varepsilon_R], \quad (3.3)$$

$$\frac{\partial}{\partial t} (C_V T) = -\frac{c}{\lambda_R} [aT^4 - \varepsilon_R]. \quad (3.4)$$

Here, $\varepsilon_R = aT_R^4$ is the radiation energy density; T_R is the local radiation temperature; c is the speed of light; T is the local matter temperature; and C_V is the matter heat capacity per volume. From the above remarks we have

$$\lambda_R = \frac{1}{\kappa \rho} = \frac{T^\gamma}{\mathcal{O}_\rho \mathcal{O}_T \rho^{1+X_\rho^0}}. \quad (3.5)$$

We can define a characteristic length scale λ_0 by evaluating (3.5) at the given density $\rho = \rho_0$ and temperature $T = T_0$

$$\lambda_0 = \frac{T_0^\gamma}{\mathcal{O}_\rho \mathcal{O}_T \rho_0^{1+X_\rho^0}}, \quad (3.6)$$

and a characteristic energy density

$$\varepsilon_0 = aT_0^4. \quad (3.7)$$

Our first task is to express (3.3) and (3.4) in dimensionless form. We start with the radiation PDE. To begin, divide (3.3) by ε_0 , and let $\varepsilon \equiv \varepsilon_R / \varepsilon_0 = (T_R / T_0)^4$

$$\frac{\partial \varepsilon}{\partial t} - \frac{\partial}{\partial x} \left[\frac{c}{3} \lambda_R \frac{\partial \varepsilon}{\partial x} \right] = \frac{c}{\lambda_R} \left[\left(\frac{T}{T_0} \right)^4 - \varepsilon \right]. \quad (3.8)$$

Here, $0 < x < \infty$ and $x = 0$ corresponds to the location of the interface separating the radiation source and the matter. Next, we introduce dimensionless coordinates for space and time: $z = x/\lambda_0$ and $\tau = ct/\lambda_0$. Multiplying (3.8) by λ_0/c gives

$$\frac{\partial \varepsilon}{\partial \tau} - \frac{\partial}{\partial z} \left[\frac{1}{3} \left(\frac{\lambda_R}{\lambda_0} \right) \frac{\partial \varepsilon}{\partial z} \right] = \frac{\lambda_0}{\lambda_R} \left[\left(\frac{T}{T_0} \right)^4 - \varepsilon \right]. \tag{3.9}$$

Finally, from (3.5) and (3.6), we have

$$\frac{\lambda_R}{\lambda_0} = \left(\frac{T}{T_0} \right)^\gamma. \tag{3.10}$$

Defining $\mathcal{T} \equiv T/T_0$, and inserting (3.10) in (3.9) gives

$$\frac{\partial \varepsilon}{\partial \tau} - \frac{\partial}{\partial z} \left[D(\mathcal{T}) \frac{\partial \varepsilon}{\partial z} \right] = \frac{[\mathcal{T}^4 - \varepsilon]}{(\mathcal{T})^\gamma}, \tag{3.11}$$

where the diffusion coefficient $D(\mathcal{T}) = (\mathcal{T})^\gamma/3$.

Repeating the same sequence of steps on the matter PDE ((3.4)) gives

$$\frac{\partial}{\partial \tau} \left[\left(\frac{C_V}{aT_0^3} \right) \mathcal{T} \right] = -\frac{\lambda_0}{\lambda_R} [\mathcal{T}^4 - \varepsilon]. \tag{3.12}$$

Since aT_0^3 has the units of heat capacity per volume, we can define a dimensionless matter heat capacity per volume

$$\mathcal{C} = \frac{C_V}{aT_0^3}. \tag{3.13}$$

For the results presented in figures 1 and 2 below we set $\mathcal{C} = 0.01$, while for figure 3 $\mathcal{C} = 0.1$. Using (3.10) and (3.13) in (3.12) gives

$$\frac{\partial \mathcal{T}}{\partial \tau} = -\frac{1}{\mathcal{C}} \frac{[\mathcal{T}^4 - \varepsilon]}{(\mathcal{T})^\gamma}. \tag{3.14}$$

Thus we arrive at our dimensionless pair of governing nonlinear PDEs

$$\frac{\partial \varepsilon}{\partial \tau} - \frac{\partial}{\partial z} \left[D(\mathcal{T}) \frac{\partial \varepsilon}{\partial z} \right] = \frac{[\mathcal{T}^4 - \varepsilon]}{(\mathcal{T})^\gamma}, \tag{3.15}$$

$$\frac{\partial \mathcal{T}}{\partial \tau} = -\frac{1}{\mathcal{C}} \frac{[\mathcal{T}^4 - \varepsilon]}{(\mathcal{T})^\gamma}. \tag{3.16}$$

Finally, we specify the boundary and initial conditions imposed on the PDE solutions. The boundary conditions at the interface, $z = 0$, are

$$\varepsilon(0, \tau) = 1, \tag{3.17}$$

$$\mathcal{T}(0, \tau) = 1. \tag{3.18}$$

We impose floating boundary conditions at the right computational boundary, $z_M = 1$, which allows the Marshak wave to propagate through this boundary (the subscript on z is a grid point label we introduce in § 3.2). Thus, for example, for the radiation energy

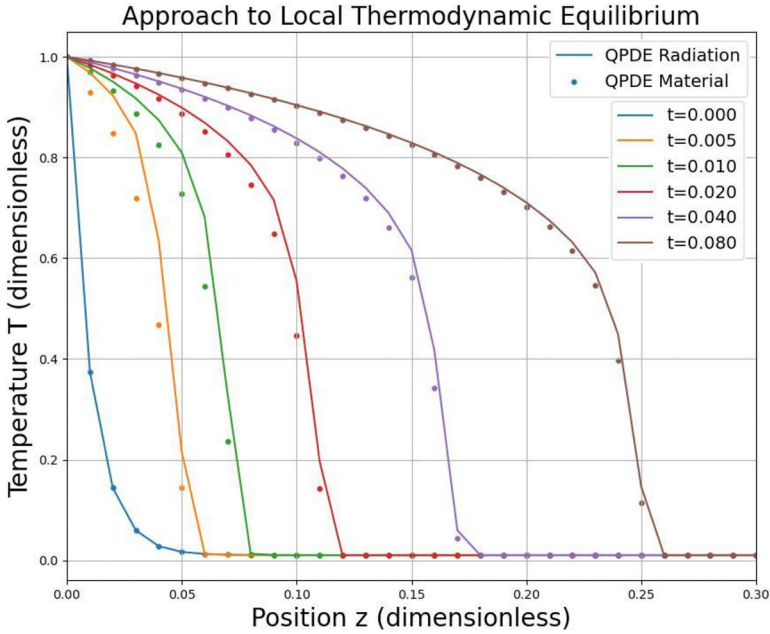


FIGURE 1. Approach to LTE. We plot the results of the approximate solution found through numerical simulation of the QPDE algorithm applied to the non-equilibrium Marshak wave problem. The solid curves show the radiation temperature versus position at six different times, while the dots do the same for the matter temperature. The colours encode the different times shown in the figure. We clearly see the propagating thermal front due to nonlinear radiation diffusion, and the increase of the local matter temperature until LTE is established with the radiation.

density ε , the boundary value $\varepsilon(z_M, \tau)$ is obtained by extrapolating the line going through the interior data points $(z_{M-2}, \varepsilon(z_{M-2}, \tau)), (z_{M-1}, \varepsilon(z_{M-1}, \tau))$. The resulting boundary conditions are

$$\varepsilon(z_M, \tau) = 2\varepsilon(z_{M-1}, \tau) - \varepsilon(z_{M-2}, \tau), \tag{3.19}$$

$$\mathcal{T}(z_M, \tau) = 2\mathcal{T}(z_{M-1}, \tau) - \mathcal{T}(z_{M-2}, \tau), \tag{3.20}$$

for $\tau > 0$. For $\tau = 0$ we set $\mathcal{T}(z_M, 0) = 0.01$ for the simulations results presented in figures 1 and 2, while $\mathcal{T}(z_M, 0) = \sqrt{0.1}$ for the simulation results presented in figure 3.

The matter initial condition is a rapidly decaying exponential passing through the boundary value at $z_1 = 0$ ((3.18)) and $\mathcal{T}(z_M, 0) = 0.01$

$$\mathcal{T}(z, 0) = \mathcal{T}(z_M, 0) + [\mathcal{T}(0, 0) - \mathcal{T}(z_M, 0)] \times \exp[-z/0.01], \tag{3.21}$$

and the initial condition for the radiation is

$$\varepsilon(z, 0) = [\mathcal{T}(z, 0)]^4. \tag{3.22}$$

3.2. Applying the QPDE algorithm

As discussed in § 2, the first step in applying the QPDE algorithm to (3.15) and (3.16) is to discretize space. We thus introduce a spatial grid containing M grid points z_1, \dots, z_M . Here, $z_1 = 0$ and $z_M = 1$ are, respectively, the left and right boundaries of

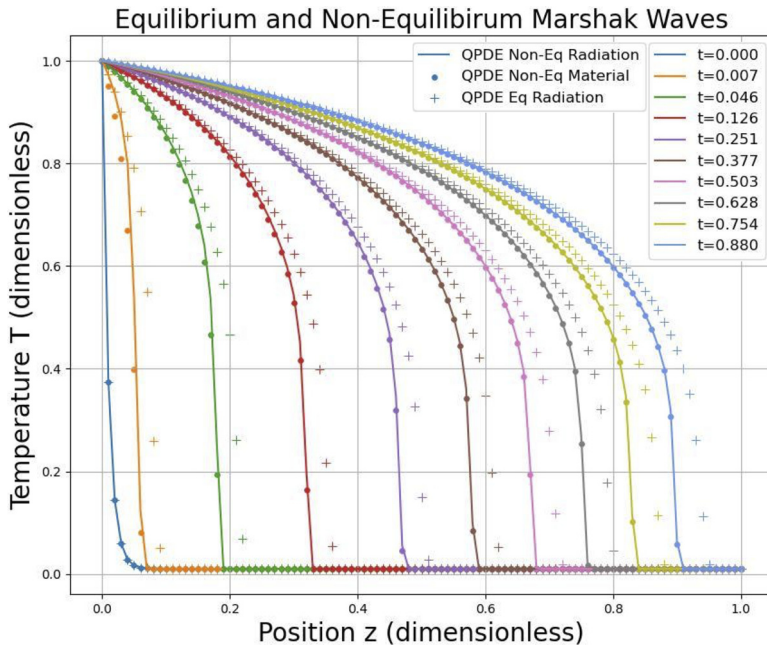


FIGURE 2. Emergence of equilibrium Marshak wave. We plot the simulation results produced by the QPDE algorithm applied to both the NMWP considered in this paper, and the EMWP considered in Gaitan *et al.* (2024). Unlike with the NMWP, the EMWP assumes LTE exists between the radiation and matter. The solid curves and dots correspond to the radiation and matter temperatures for the NMWP, respectively, at ten times. The data shown are for sufficiently large times that the NMWP solution has effectively reached LTE. The crosses correspond to the LTE temperature of the EMWP. We see that the QPDE solution of the NMWP at large times is converging to the EMWP solution, properly capturing the emergence of the equilibrium Marshak wave as the large time limit of the NMWP solution.

our computational domain. The grid points z_2, \dots, z_{M-1} are the interior points. In the simulations discussed in § 3.3, we set $M = 101$, giving a grid spacing of $\Delta z = 0.01$.

The spatial discretization requires approximation of all spatial partial derivatives. We used central-difference FD approximations, and used Python’s symbolic mathematics library SymPy to implement the FD approximations. As explained in § 2, the result of the spatial discretization is to reduce our pair of PDEs to a pair of ODEs (see (2.2)). The resulting ODE solution vector $U_I(t)$ is

$$U_I(t) = \begin{pmatrix} \varepsilon_I(t) \\ \mathcal{T}_I(t) \end{pmatrix}, \tag{3.23}$$

where $\varepsilon_I(t) \equiv \varepsilon(z_I, t)$; $\mathcal{T}_I(t) = \mathcal{T}(z_I, t)$; and $1 \leq I \leq M$. The driver function that results from the spatial discretization is

$$f_I(U) = \begin{pmatrix} \frac{D_j[\varepsilon_{j+1} - 2\varepsilon_j + \varepsilon_{j-1}]}{(\Delta z)^2} + \frac{[D_{j+1} - D_{j-1}][\varepsilon_{j+1} - \varepsilon_{j-1}]}{4(\Delta z)^2} + \frac{[\mathcal{T}_j^4 - \varepsilon_j]}{(\mathcal{T}_j)^\gamma} \\ - \frac{1}{\mathcal{C}} \frac{[\mathcal{T}_j^4 - \varepsilon_j]}{(\mathcal{T}_j)^\gamma} \end{pmatrix}, \tag{3.24}$$

where $D_j = (\mathcal{T}_j)^\gamma/3$.

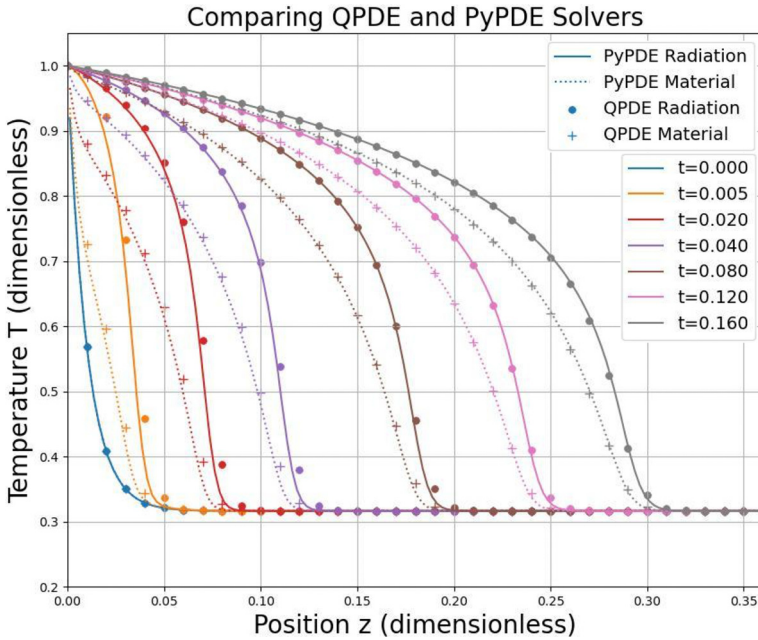


FIGURE 3. Comparing results of the QPDE algorithm and Python’s py-pde solver for the NMWP. We compare the approximate solution found using the QPDE algorithm applied to the NMWP with that found using Python’s py-pde solver. The dots and crosses are the radiation and matter temperatures versus position, respectively, found using the QPDE algorithm, while the solid and dashed curves are the radiation and matter temperatures versus position, respectively, found using the py-pde solver. The simulation results are shown at seven different times, with each time associated with a different colour. We see that there is excellent agreement between the two sets of solutions (crosses with dashes, dots with solid) at intermediate to later times, and good agreement at the earliest times.

The second step in the QPDE algorithm is to use Kacewicz’ quantum ODE algorithm to solve (2.2) with $f_I(U)$ given by (3.24), and the boundary and initial conditions appearing in § 3.1. We numerically simulated determining an approximate solution of our pair of ODEs using Kacewicz’ algorithm. Note that we also used SymPy to analytically determine the first and second time derivatives of $f_I(U)$ which are needed to construct the Taylor polynomials $\alpha_{i,j}(t)$ discussed in § 2. To test the quality of the quantum simulation results, we also used Python’s PDE solver py-pde to obtain an approximate solution of (3.15) and (3.16), subject to the boundary and initial conditions of § 3.1. We compare the py-pde solution with that of the quantum simulation in § 3.3.

3.3. Simulation results

In this section we present our simulation results for the non-equilibrium Marshak wave problem (NMWP). These simulations used the following parameter values: (i) number of spatial grid points $M = 101$; (ii) grid point spacing $\Delta z = 0.01$; (iii) $\gamma = 3.5$; and (iv) for figures 1 and 2, $\mathcal{Q} = 0.01$, while for figure 3, $\mathcal{Q} = 0.1$. The boundary and initial conditions were specified in § 3.1.

Figure 1 shows the simulation results obtained by applying the QPDE algorithm to the NMWP. The governing PDEs are (3.15) and (3.16). The figure plots the radiation (solid curve) and matter (dots) temperatures versus position at six times. The propagating

radiation and matter thermal fronts associated with NRD are clearly visible. If we focus on a particular position z in the figure, we see that the local matter temperature increases with time as the matter absorbs energy from the radiation. The local matter temperature continues to increase until it equals the local radiation temperature, signalling the establishment of LTE. We see that by $t = 0.080$, the QPDE simulation results indicate that LTE is effectively established.

In this paper we have considered the NMWP in which the radiation and matter are not assumed to be in LTE. On the other hand, Gaitan *et al.* (2024) considered the equilibrium Marshak wave problem (EMWP) where radiation and matter are assumed to be in LTE. As was just seen in figure 1, the QPDE algorithm simulation results show the approach of the radiation and matter to LTE. In figure 2 we plot the QPDE algorithm simulation results for the NMWP at times that are sufficiently large that the radiation and matter are effectively in LTE, with the two systems getting closer and closer to LTE as time progresses. We also plot the QPDE algorithm solution of the EMWP. The solid curve and dots are the radiation and matter temperatures versus position, respectively, for the NMWP, while the crosses are the LTE temperature versus position for the EMWP found by the QPDE algorithm. The results are plotted for ten times, with a different colour associated with each time. We see that as time progresses the NMWP solution is converging to the EMWP solution. The QPDE algorithm thus properly captures the emergence of the equilibrium Marshak wave as the large time limit of the NMWP solution.

In figure 3 we compare the approximate solutions found for the NMWP using the QPDE algorithm and Python's `py-pde` solver. The dots and crosses represent the radiation and matter temperatures versus position, respectively, found using the QPDE algorithm, while the solid and dashed curves represent the radiation and matter temperatures versus position, respectively, found using the `py-pde` solver. We show results for seven times, each time encoded with a different colour. Inspection of figure 3 shows that the dots and solid curves at corresponding times, representing the radiation temperature versus position for the two simulations, are in excellent agreement at intermediate to later times, and in good agreement at the earliest times. Similarly, the crosses and dashed curves at corresponding times, representing the matter temperature versus position for the two simulations, are also in excellent agreement at intermediate to later times, and in good agreement at the earliest times. Figures 1–3 provide strong evidence that the QPDE algorithm correctly solves the NMWP. The QPDE algorithm was similarly verified on the EMWP in Gaitan *et al.* (2024).

4. Non-equilibrium Marshak wave problem: second verification

In this section we compare the simulation results for the NMWP generated by the QPDE algorithm and the KULL ICF simulation code. The KULL multiphysics code (Rathkopf *et al.* 2000) was developed at LLNL primarily for simulating inertial confinement fusion processes at the National Ignition Facility. The code is multidimensional, mesh based, and able to accommodate many types of physics, including multiple radiation transport options: diffusion, discrete ordinates (also known as SN), and implicit Monte Carlo. In this section we are comparing with KULL's radiation diffusion package, which uses a node-centred finite element discretization.

As the KULL software is designed to predict experimental outcomes, it utilizes dimensionful parameters in the radiation and matter energy balance PDEs ((3.3) and (3.4)). The basic units are: [length] = cm; [time] = shakes (sh) = 10^{-8} s; [energy] = jerks (jrk) = 10^{16} ergs; and [temperature] = keV. In these units the speed of light is $c = 299.7925$ cm sh $^{-1}$ and $a = 1.372017 \times 10^{-2}$ jrjrk cm $^{-3}$ keV 4 . The parameter values used in the simulations were chosen so that simulation runtimes would not be excessively long. We discuss this further in § 5. The Rosseland mean free path λ_R is given by (3.5),

with $\mathcal{O}_\rho = 0.9053885 \text{ cm}^{4.25} \text{ g}^{-1.75}$; $\mathcal{O}_T = 3.162278 \text{ keV}^2$; $X_\rho^0 = 0.75$; and $\gamma = 2.0$. The constant matter density is $\rho = 0.6729501 \text{ g cm}^{-3}$. The matter heat capacity per volume $C_V = de/dT$ is obtained from the equation of state

$$e = C_{eos} \rho^{X_\rho^{eos}} T^{X_T^{eos}}, \quad (4.1)$$

where T and ρ are the matter temperature and density, respectively; $C_{eos} = 4.90693 \times 10^{-4} \text{ jrk cm}^{-3} \text{ g}^{-0.91} \text{ keV}^{-1}$; $X_\rho^{eos} = 0.91$; and $X_T^{eos} = 1.0$. Finally, we define $\varepsilon \equiv \varepsilon_R/a = T_R^4$. Inserting these parameter values into (3.3) and (3.4) we obtain the following governing PDEs:

$$\frac{\partial \varepsilon}{\partial t} - \frac{\partial}{\partial x} \left[69.8063 T^\gamma \frac{\partial \varepsilon}{\partial x} \right] = 429.167 \left[\frac{T^4 - \varepsilon}{T^\gamma} \right], \quad (4.2)$$

$$\frac{\partial T}{\partial t} = -1.72073 \times 10^4 \left[\frac{T^4 - \varepsilon}{T^\gamma} \right]. \quad (4.3)$$

The spatial computational domain is $0 \text{ cm} \leq x \leq 8.6258 \times 10^{-2} \text{ cm}$.

The boundary conditions at the left boundary $x = 0$ are $T_R(0, t) = T(0, t) = 0.15 \text{ keV}$ so that $\varepsilon(0, t) = 5.0625 \times 10^{-4} \text{ keV}^4$. We again impose floating boundary conditions at the right boundary $x_M = 8.6258 \times 10^{-2} \text{ cm}$ so that for $t > 0$,

$$\varepsilon(x_M, t) = 2\varepsilon(x_{M-1}, t) - \varepsilon(x_{M-2}, t), \quad (4.4)$$

$$T(x_M, t) = 2T(x_{M-1}, t) - T(x_{M-2}, t), \quad (4.5)$$

while for $t = 0$ we set $T_R(x_M, 0) = T(x_M, 0) = 0.015 \text{ keV}$, and so $\varepsilon(x_M, 0) = 5.0625 \times 10^{-8} \text{ keV}^4$. The initial condition for the matter temperature $T(x, 0)$ is

$$T(x, 0) = T(x_M, 0) + [T(0, 0) - T(x_M, 0)] \exp(-x/\Delta), \quad (4.6)$$

where $\Delta = 8.6258 \times 10^{-4} \text{ cm}$. The initial condition for the radiation is $\varepsilon(x, 0) = [T(x, 0)]^4$. The simulations used $M = 101, 201, 401$ grid points, giving a grid spacing $\Delta x = x_M/(M - 1) = \{8.6258, 4.3129, 2.1565\} \times 10^{-4} \text{ cm}$.

Application of the QPDE algorithm to the dimensionful energy balance PDEs is the same as described in § 3.2 and so need not be repeated here.

Figure 4 shows the simulation results for the QPDE algorithm and the LLNL/KULL ICF software for spatial grids using $M = 101, 201, 401$ grid points. The dots and crosses are the QPDE radiation and matter temperatures, respectively, while the solid and dashed curves are the KULL radiation and matter temperatures. We see that for each spatial grid and each of the times plotted, the agreement is quite good, although the KULL solution is seen to produce a slightly faster propagating thermal front than the QPDE solution. Consequently, the agreement between the two simulations slowly deteriorates with increasing time since the KULL thermal fronts get further and further ahead of the QPDE thermal fronts with time. Note that the speed difference appears to be an artefact of the grid size. For the 3 spatial grids considered, the speed difference is largest for the grid with $M = 101$ grid points, is smaller for $M = 201$ and even smaller for $M = 401$. This suggests that this speed difference, and hence the deteriorating agreement between the simulations with time, will go to zero in the continuum limit. To examine this further, in figure 5 we plot the temperature difference $\Delta T = T_{KULL} - T_{QPDE}$ versus grid position z for the 3 spatial grids: $M = 101, 201, 401$. This temperature difference provides a direct measure of the disagreement between the QPDE and KULL simulation results. The solid

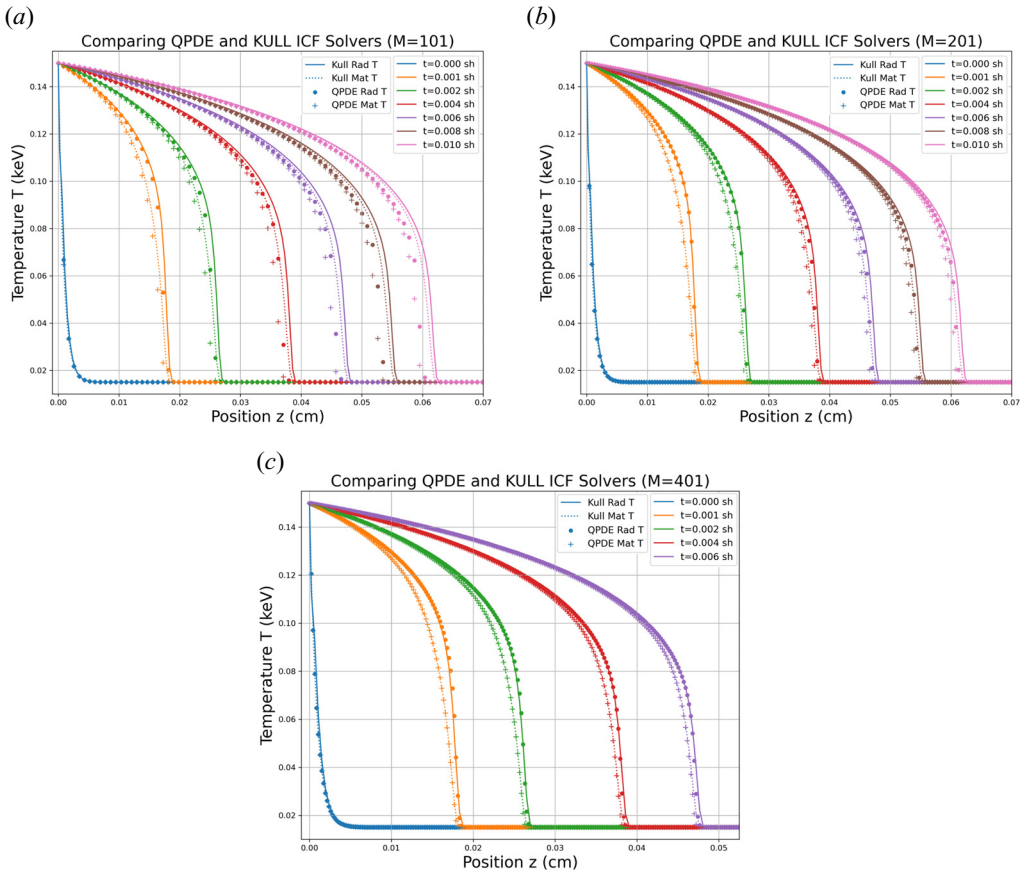


FIGURE 4. Comparing results of the QPDE and KULL ICF simulations for the NMWP. We compare the approximate solution found using the QPDE algorithm applied to the NMWP with that found using LLNL’s KULL ICF radiation diffusion simulation software. We consider 3 spatial grids with (a) $M = 101$ grid points; (b) $M = 201$ grid points; and (c) $M = 401$ grid points. The dots and plus signs are the QPDE radiation and matter temperatures, respectively, while the solid and dashed curves are the KULL radiation and matter temperatures. In (a) we see that both the QPDE and KULL simulation results each show the radiation and matter approaching local thermal equilibrium as time increases, however, the agreement between the two simulations which is quite good initially, is seen to slowly deteriorate with time as the KULL radiation and matter thermal fronts appear to be moving slightly faster than the QPDE radiation and matter thermal fronts. The same behaviour is seen in (b,c), only the difference in thermal front speeds is smaller for $M = 201$, and even smaller for $M = 401$. This suggests that this speed difference, and the associated slow deterioration of the agreement in the simulation results with time, is an artefact of the grid-size that will go to zero in the continuum limit.

curve gives the temperature difference ΔT for the matter field, and the dashed curve gives ΔT for the radiation. We see that the difference in the thermal front speeds for the two simulations produces a systematic increase in ΔT with time, and is largest for $M = 101$, is much less for $M = 201$ and much, much less for $M = 401$. As noted above, this suggests that the difference in thermal front speeds between the two simulations, and consequently, their discrepancy measure ΔT , goes to zero in the continuum limit. Two other suspected causes for the small difference in thermal front speeds are: (i) the finite time scales

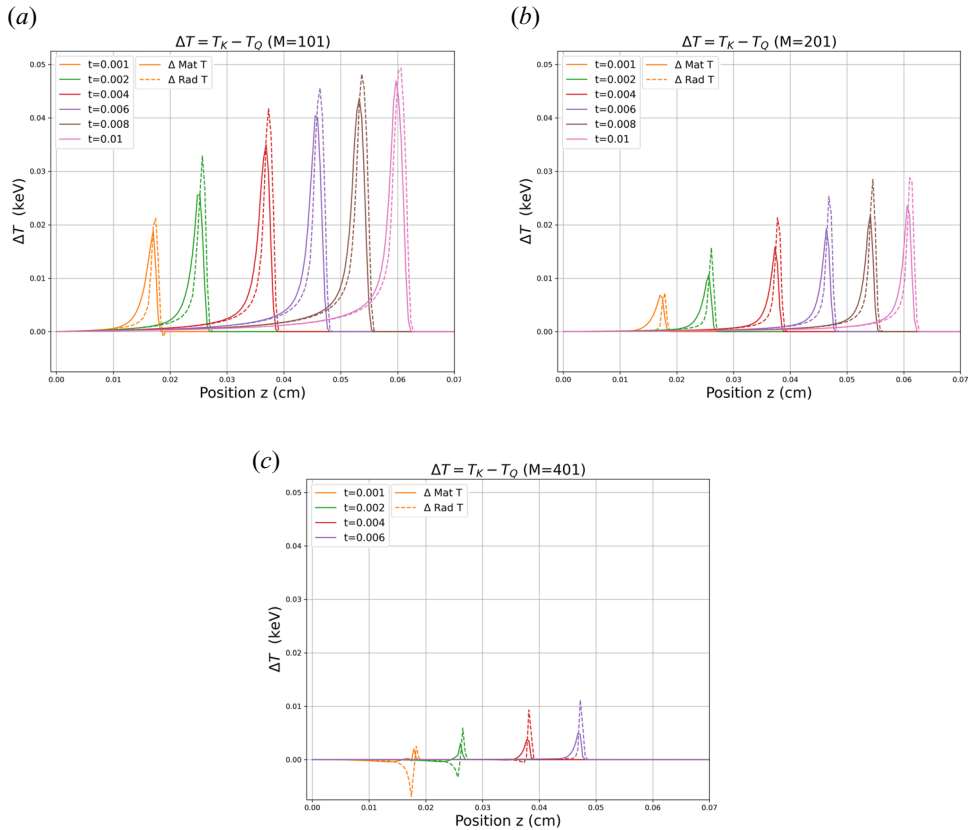


FIGURE 5. Comparing the temperature difference $\Delta T = T_{KULL} - T_{QPDE}$ versus position z for the QPDE and KULL ICF simulations for the NMWP. We use the temperature difference $\Delta T = T_{KULL} - T_{QPDE}$ versus position z as a direct measure of the disagreement between the QPDE and KULL ICF simulation results. The solid curve gives the temperature difference ΔT for the matter field, and the dashed curve gives ΔT for the radiation. (a) Plots ΔT for a grid with $M = 101$ grid points; (b) for $M = 201$; and (c) for $M = 401$. We see that the difference in the radiation and matter thermal front speeds noted in figure 4 causes a systematic increase in ΔT with time as the KULL thermal fronts move slightly faster than the QPDE thermal fronts, thus getting further and further ahead of the QPDE thermal fronts with time. The effect is seen to be largest for the grid with $M = 101$, to be much less for the grid with $M = 201$ and much, much less for the grid with $M = 401$. This suggests that the difference in thermal front speeds between the two simulations, and consequently, their discrepancy ΔT , is an artefact of the grid size that would appear to go to zero in the continuum limit.

used to propagate the two simulations forward in time; and (ii) the use of implicit time stepping versus explicit time stepping in the KULL and QPDE simulations, respectively. In future work we plan to do a more complete convergence study of the KULL and QPDE simulations and we anticipate the radiation and matter thermal front speeds for the two simulations will converge to common speeds. As will be discussed further in § 5, work is currently underway to implement the implicit version of the QPDE algorithm presented in Gaitan (2021). With that said, the simulation results presented here provide further evidence that the QPDE algorithm correctly solves the NMWP, this time when compared

with the state-of-the-art radiation diffusion package of LLNL's KULL ICF simulation software.

5. Discussion

In this paper we applied a QPDE algorithm (Gaitan 2020, 2021), which provides up to a quadratic quantum speedup, to the NMWP (Pomraning 1979). We showed that the QPDE algorithm correctly captured the: (i) approach to LTE of the coupled radiation/matter system; and (ii) emergence of the equilibrium Marshak wave (Marshak 1958) as the large time limit of the NMWP solution. The QPDE algorithm solution was compared with that found using Python's `py-pde` solver (Zwicker 2020) and the radiation diffusion package of LLNL's KULL ICF simulation software (Rathkopf *et al.* 2000). Both comparisons provided clear evidence that the QPDE algorithm correctly solves the NMWP.

The QPDE algorithm considered in this paper uses an explicit time development scheme (see (2.3)). Because the radiation transport time scale is much shorter than the matter time scale, numerical stability required us to use a secondary subinterval duration time \bar{h} that is less than the radiation time scale. This, unfortunately, leads to prohibitively long runtimes when physical parameters are assigned values typical of ICF experiments. As noted in § 4, to mitigate this situation, it was necessary to restrict ourselves to parameter values that allowed practical simulation runtimes. In an attempt to resolve this issue, work is currently underway to implement the implicit version of the QPDE algorithm introduced in Gaitan (2021). Implicit algorithms are typically more stable than explicit algorithms, and can often be shown to be unconditionally stable. This will allow us to use larger values of \bar{h} than is possible with our explicit algorithm, thus reducing runtimes. We anticipate that this will allow us to simulate the QPDE algorithm with reasonable runtimes, even when using experimentally relevant ICF parameter values. In fact, present-day RH codes simulate radiation transport using implicit codes for this reason.

Acknowledgements

F.G. thanks T. Howell III for continued support. Part of this work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC.

Editor Nuno Loureiro thanks the referees for their advice in evaluating this article.

Declaration of interest

The authors report no conflict of interest.

Code availability

The computer code used to implement the numerical simulations presented in this Paper is available at: <https://github.com/myerscon/PythonQPDE.git>.

REFERENCES

- CASTOR, J.I. 2004 *Radiation Hydrodynamics*. Cambridge University Press.
- GAITAN, F. 2020 Finding flows of a Navier–Stokes fluid through quantum computing. *NPJ Quant. Inf.* **6**, 61.
- GAITAN, F. 2021 Finding solutions of the Navier–Stokes equations through quantum computing—recent progress, a generalization, and next steps forward. *Adv. Quant. Tech.* **4**, 2100055.
- GAITAN, F. 2024 Circuit implementation of oracles used in a quantum algorithm for solving nonlinear partial differential equations. *Phys. Rev. A* **109**, 032604.

- GAITAN, F., GRAZIANI, F. & PORTER, M. 2024 Simulating nonlinear radiation diffusion through quantum computing. *Int. J. Theor. Phys.* **63**, 260.
- GAUTSCHI, W. 2012 *Numerical Analysis*. Birkhäuser.
- GRAZIANI, F. 2005 Radiation diffusion: an overview of physical and numerical concepts. *Tech. Rep.* UCRL-PROC-209053. Lawrence-Livermore National Laboratory. Available at: <https://www.osti.gov/servlets/purl/917504>.
- ISERLES, A. 2009 *A First Course in the Numerical Analysis of Differential Equations*. Cambridge University Press.
- KACEWICZ, B. 2006 Almost optimal solution of initial-value problems by randomized and quantum algorithms. *J. Complex.* **22**, 676.
- MARSHAK, R.E. 1958 Effect of radiation on shock behavior. *Phys. Fluids* **1**, 24.
- NOVAK, E. 2001 Quantum complexity of integration. *J. Complex.* **17**, 2.
- PLETCHER, R.H., TANNEHILL, J.C. & ANDERSON, D.A. 2013 *Computational Fluid Mechanics and Heat Transfer*, 3rd edn. CRC Press.
- POMRANING, G.C. 1979 The non-equilibrium Marshak wave problem. *J. Quant. Spectrosc. Radiat. Transfer* **21**, 249.
- RATHKOPF, J.A., MILLER, D.S., OWEN, J.M., STUART, L.M., ZIKA, M.R., ELTGROTH, P.G., MADSEN, N.K., MCCANDLESS, K.P., NOWAK, P.F., NEMANIC, M.K., GENTILE, N.A., KEEN, N.D. & PALMER, T.S. 2000 KULL: LLNL's ASCI inertial confinement fusion simulation code. *Tech. Rep.* UCRL-JC-137053. Lawrence-Livermore National Laboratory. Available at: <https://www.osti.gov/servlets/purl/791024>.
- ZWICKER, D. 2020 py-pde: a python package for solving partial differential equations. *J. Open Source Softw.* **5**, 2158.