Dimensional Scaling and Numerical Similarity in Hyperbolic Method for Diffusion

Hiroaki Nishikawa*

National Institute of Aerospace, Hampton, VA 23666, USA

Yoshitaka Nakashima[†] Software Cradle Co., Ltd, Osaka, JAPAN

November 14, 2017

Abstract

This paper discusses issues encountered by the hyperbolic method for diffusion [J. Comput. 26 Phys. 227 (2007) 315-352] in dimensional heat conduction problems, and proposes a practical resolution. It is shown that the relaxation length must be scaled by a reference length of a domain of interest for solving dimensional equations, and the corresponding non-dimensionalized length should be given an optimal value for fast iterative convergence. To achieve both, a practical formula is proposed for computing a reference length for a given computational grid, such that $(2\pi)^{-1}$ gives an optimal value for rectangular domains and also serves as an effective approximation for general domains. Numerical results confirm that the proposed scaling is critically important for rendering hyperbolic diffusion schemes independent of the grid unit and for achieving optimal performance of a hyperbolic diffusion solver.

1 Introduction

The hyperbolic method is a method of deriving a spatial discretization of the diffusion equation:

$$\partial_t u = \nu (\partial_{xx} u + \partial_{yy} u), \tag{1}$$

where ν is a diffusion coefficient and u is a scalar solution variable, by discretizing a first-order hyperbolic system [1]:

$$\partial_{\tau}u = \nu(\partial_x p + \partial_y q) - \partial_t u, \quad \partial_{\tau}p = \frac{1}{T_r}(\partial_x u - p), \quad \partial_{\tau}q = \frac{1}{T_r}(\partial_y u - q), \tag{2}$$

where τ is a pseudo time, and T_r is called the relaxation time. The physical time derivative is discretized in time and treated as a source term, thus setting up a problem of solving the hyperbolic diffusion system towards the pseudo steady state to advance the solution by one physical time step [2, 3, 4]. Although it resembles the classical hyperbolic heat equations of Cattaneo [5] and Vernotte [6], the above formulation allows us to take T_r as a *free parameter* and it can be defined so as to accelerate convergence towards the pseudo steady state. This is the core idea of the hyperbolic method. Dimensional consistency suggests that we set

$$T_r = \frac{L_r^2}{\nu},\tag{3}$$

where L_r is a relaxation length, and an optimal formula for L_r has been derived [1, 7] as

$$L_r = \frac{1}{2\pi},\tag{4}$$

^{*(}hiro@nianet.org), 100 Exploration Way, Hampton, VA 23666 USA,

[†](nakashima@cradle.co.jp), Software Cradle Co., Ltd, Umeda Kita-ku, Osaka, Japan

which is the leading term of the one derived specifically for a residual-distribution scheme in Ref.[1], and also of the one derived for a first-order finite-volume scheme to ensure that Fourier modes are propagated, instead of purely damped, to maximize the effect of error propagation [7]. The system is hyperbolic in the pseudo time τ , and therefore can be discretized by methods for hyperbolic systems (e.g., upwind methods). Once the system is discretized in space, the pseudo time derivatives may be dropped. Then, the resulting discretization will be a consistent discrete approximation to the original diffusion equation written as a first-order system. The discretization derived this way has been found to offer various advantages: higher-order accuracy and higher quality of the gradients than reconstructed ones on unstructured grids, higher-order advective schemes if applied to advection-diffusion equations, convergence acceleration by the elimination of second derivatives, high-order inviscid schemes directly applicable to viscous terms [8, 9]. The method has been extended to a tensor-coefficient diffusion equation [10, 11], incompressible/compressible Navier-Stokes equations [12, 13, 14], an incompressible magnetohydrodynamics model [15], quasi-neutral plasma models [10, 16], and the dispersion equation [17]. Similar hyperbolic approaches exist in the literature [18, 19], but the hyperbolic method considered here is different in that the relaxation time is an O(1) free parameter and carries no physical significance. More specifically, unlike other approaches, the hyperbolic method does not propose a new physical model for diffusion; it is merely a way of constructing a superior spatial discretization for the second-derivative diffusion operator expressed in a consistent first-order system form.

A recent attempt to solve a dimensional heat conduction equation reveals that the hyperbolic method encounters serious accuracy and convergence problems, exhibiting strong dependence on the scale used to define a computational grid. This is a non-typical behavior for a diffusion scheme since there are no similarity parameters in the diffusion equation. Any numerical scheme should yield the same numerical solution, with a given set of boundary and initial conditions, for any grid unit. The objective of this paper is to clarify the issue and present a practical resolution, so that the hyperbolic method can be applied to a wide range of applications, including dimensional equations. As will be discussed below, the erratic behavior of the hyperbolic diffusion scheme is related to the scaling of L_r . The subjects of discussion are threefold:

- 1. Numerical dependence on L_r .
- 2. Dimensional scaling of L_r to achieve scale-invariant computations.
- 3. Estimate of a reference length to achieve optimal performance.

The first subject concerns the fact that the pseudo-steady solution to the hyperbolic diffusion system is independent of L_r analytically, but numerical solutions are not. It will be shown that L_r influences the dissipation of a numerical scheme, and so accuracy and iterative convergence will be affected unless L_r is defined properly for varying units. The second subject deals with a proper scaling of L_r to remove scale-dependence from numerical solutions and achieve numerical similarity. The third, perhaps the most important, subject is on how to determine the reference length for properly scaling L_r for a given problem in a general, practical, and optimal manner. For discussions that follow, the physical time derivative is irrelevant, and therefore we ignore the physical time derivative and focus on the spatial discretization. However, numerical results are presented for an unsteady heat conduction problem to demonstrate that improvements carry over directly to unsteady problems. It is emphasized that the dimensional issues discussed in this paper are relevant to other hyperbolic approaches such as Ref.[18], but the proposed resolution is specific to the hyperbolic method considered here and may not be directly applicable to other hyperbolic methods.

The paper is organized as follows. In Section 2, it is shown that a hyperbolic diffusion scheme is independent of ν for steady problems, but strongly depends on the relaxation length L_r . In Section 3, L_r is shown to be a dimensional parameter and needs to be scaled by a reference length when the dimensional system is solved. A question is raised: What reference length would make $\frac{1}{2\pi}$ be optimal for non-square domains? In Section 4, the question is answered for a rectangular domain by deriving an optimal formula for the dimensional L_r . In Section 5, the reference length for a rectangular domain is extended to arbitrary shaped domains. In Section 6, the discretization and the implicit solver used to perform numerical experiments are briefly described. In Section 7, numerical results are presented for both steady and unsteady heat conduction problems, including a steady nonlinear problem. Finally, the paper concludes with remarks.

2 Dependence on L_r

Consider the hyperbolic system with the physical time derivative ignored:

$$\partial_{\tau}u = \nu(\partial_x p + \partial_y q), \quad \partial_{\tau}p = \frac{\nu}{L_r^2}(\partial_x u - p), \quad \partial_{\tau}q = \frac{\nu}{L_r^2}(\partial_y u - q).$$
 (5)

In the pseudo steady state or as soon as we drop the pseudo time terms, we have

$$0 = \nu(\partial_x p + \partial_y q), \quad 0 = \frac{\nu}{L_r^2} (\partial_x u - p), \quad 0 = \frac{\nu}{L_r^2} (\partial_y u - q), \tag{6}$$

which reduce to, for any nonzero ν and L_r ,

$$0 = \partial_x p + \partial_y q, \quad 0 = \partial_x u - p, \quad 0 = \partial_y u - q, \tag{7}$$

and thus

$$\partial_{xx}u + \partial_{yy}u = 0. \tag{8}$$

The hyperbolic diffusion system is, therefore, equivalent to the steady diffusion equation. Steady solutions have no dependence on ν nor L_r in the differential-equation level.

In the discrete level, the relaxation length L_r plays a key role in characterizing the spatial discretization of the hyperbolic diffusion system. Upwind discretizations, as often employed in the hyperbolic method, are constructed based on the eigenvalues of a hyperbolic system. For the hyperbolic diffusion system, the eigenvalues are given by [1].

$$\lambda = \pm \nu / L_r, 0. \tag{9}$$

The corresponding eigenvectors are linearly independent, and therefore the upwind flux typical in a finite-volume scheme can be constructed (see Ref.[7]) as

$$\mathbf{\Phi}_{jk} = \frac{\nu}{2} \begin{bmatrix} -(p_L, q_L) \cdot \mathbf{n}_{jk} - (p_R, q_R) \cdot \mathbf{n}_{jk} \\ -\frac{1}{L_r^2} (u_L + u_R) n_x \\ -\frac{1}{L_r^2} (u_L + u_R) n_y \end{bmatrix} - \frac{\nu}{2L_r} \begin{bmatrix} 1 & 0 & 0 \\ 0 & n_x^2 & n_x n_y \\ 0 & n_y n_x & n_y^2 \end{bmatrix} \begin{bmatrix} u_R - u_L \\ p_R - p_L \\ q_R - q_L \end{bmatrix}, \quad (10)$$

where $\mathbf{n}_{jk} = (n_x, n_y)$ is a unit vector normal to an interface between two adjacent cells j and k, the subscripts L and R indicate face values linearly extrapolated from the centroids of the cells j and k, respectively, for second-order accuracy. Observe that the whole expression is proportional to ν . And so is the source term: $\mathbf{s} = (0, -\nu p/L_r^2, -\nu q/L_r^2)$. Hence, for steady diffusion problems, the discrete residual equation defined as an integral of the system over a cell j,

$$\operatorname{Res}_{j} = \sum_{k \in \{k_{j}\}} \Phi_{jk} A_{jk} - \mathbf{s}_{j} V_{j} = 0, \qquad (11)$$

is independent of ν , where $\{k_j\}$ is a set of neighbor cells, A_{jk} is the face area (length in two dimensions), and V_j is the measure of the cell j. Dividing the residual Res_j by ν and multiplying the last two equations by L_r^2 , we obtain

$$-\sum_{k\in\{k_j\}} \left[\{ (p_L, q_L) + (p_R, q_R) \} \cdot \mathbf{n}_{jk} + \frac{1}{L_r} (u_R - u_L) \right] A_{jk} = 0,$$
(12)

$$-\frac{1}{2}\sum_{k\in\{k_j\}} \left[(u_L+u_R)n_x + L_r \left\{ (p_R-p_L)n_x + (q_R-q_L)n_y \right\} n_x \right] A_{jk} = -p_j V_j,$$
(13)

$$-\frac{1}{2}\sum_{k\in\{k_j\}} \left[(u_L+u_R)n_y + L_r \left\{ (p_R-p_L)n_x + (q_R-q_L)n_y \right\} n_y \right] A_{jk} = -q_j V_j,$$
(14)



Figure 1: Non-dimensionalization of a square domain.

which shows that numerical solutions have no dependence on ν , but the relaxation length L_r remains and influences the amount of dissipation. Therefore, numerical solutions and solvers are affected by L_r . But it also shows that the residual is a consistent approximation to the system (7), and therefore L_r does not affect the consistency and order of accuracy of the numerical approximation. Note further that the quantities, A_{jk} , V_j , and the gradient variables (p,q) will change their numerical values for a change in the grid unit. To eliminate the dimensional dependence from the discrete equations, L_r must also change its value in a consistent manner with other quantities.

3 Numerical Similarity for Hyperbolic Scheme

3.1 Numerical Similarity

To ensure that the same numerical solution is obtained for a given problem with varying coordinate units or scaling, the relaxation length L_r needs to be defined properly. To illustrate the point, we non-dimensionalize the system (5) with the following non-dimensional variables indicated by the overbar:

$$\bar{x} = x/L, \quad \bar{y} = y/L, \quad \bar{u} = u/U, \quad \bar{p} = p/(U/L), \quad \bar{q} = q/(U/L), \quad \hat{\nu} = \nu/(L^2/T), \quad \bar{\tau} = \tau/T,$$
 (15)

where T, L, U are reference time, length, and solution, respectively. The reference length L is assumed to be chosen to scale x and y to be O(1). See Figure 1 for an example of a square domain. Substituting them into the system (5), we obtain a non-dimensionalized hyperbolic diffusion system:

$$\partial_{\bar{\tau}}\bar{u} = \hat{\nu}(\partial_{\bar{x}}\bar{p} + \partial_{\bar{y}}\bar{q}), \quad \partial_{\bar{\tau}}\bar{p} = \frac{\hat{\nu}}{\bar{L}_r^2}(\partial_{\bar{x}}\bar{u} - \bar{p}), \quad \partial_{\bar{\tau}}\bar{q} = \frac{\hat{\nu}}{\bar{L}_r^2}(\partial_{\bar{y}}\bar{u} - \bar{q}), \tag{16}$$

where

$$\bar{L}_r = \frac{L_r}{L}.\tag{17}$$

Notice that the form of the non-dimensionalized system is identical to that of the dimensional system with the non-dimensional relaxation length \bar{L}_r . As discussed earlier, steady numerical solutions are independent of $\hat{\nu}$, but depend on \bar{L}_r . Hence, a numerical scheme for the non-dimensionalized system will yield the same solution for geometrically similar objects if \bar{L}_r based on a common reference length matches. This is a numerical similarity condition for hyperbolic schemes. Translating it back to the dimensional system, we find that L_r must be defined as

$$L_r = L_r L,\tag{18}$$

which indicates that numerical schemes for the dimensional system will yield the same solution if L is defined uniquely for geometrically similar problems so as to match \bar{L}_r .

To solve the dimensional system, it is necessary to determine L and \bar{L}_r for a given problem. It is, however, not straightforward as we will discuss in the next section.

3.2 Choosing L and \overline{L}_r

Definitions of L and \overline{L}_r are rather obvious for a problem in a square domain. Consider a steady heat conduction problem in a square domain with the side length L_s : $(x, y) \in [0, L_s] \times [0, L_s]$. For a given boundary



Figure 2: Rectangular domain and two examples of non-dimensionalizations.

condition, the temperature field must be independent of the coordinate unit. A numerical scheme for the dimensional hyperbolic diffusion system will yield the same solution if the relaxation length is set by $L_r = \bar{L}_r L$. A natural choice for the reference length is the side length: $L = L_s$. Then, a proper choice for \bar{L}_r would be $\bar{L}_r = \frac{1}{2\pi}$ because it is known to be optimal for a unit square [1, 7]. As one can see, this is equivalent to non-dimensionalizing the coordinates by L_s and then solving the non-dimensionalized system with $\bar{L}_r = \frac{1}{2\pi}$.

Ambiguity arises for non-square domains. Consider a rectangular domain as illustrated in Figure 2. Apparently, there can be, at least, two possible reference lengths: the length L_x or the width L_y . In solving the dimensional system, one may set $L_r = \frac{L_x}{2\pi}$ or $L_r = \frac{L_y}{2\pi}$. Then, it leads to two different numerical solutions (although both are consistent and accurate as designed, and thus converge to the exact solution in the grid refinement). To see this, consider \bar{L}_r with a common reference length L' (say, the diagonal): $\bar{L}_r = L_r/L'$. Then, we have $\bar{L}_r = \frac{L_x/L'}{2\pi}$ in the former and $\bar{L}_r = \frac{L_y/L'}{2\pi}$ in the latter. Clearly, \bar{L}_r does not match, and therefore, numerical solutions will differ by the amount of dissipation. Non-dimensionalization does not resolve the ambiguity because it can also result in a mismatch. For example, we may non-dimensionalize the system with $L = L_x$ or $L = L_y$. The corresponding non-dimensionalized domains are illustrated in Figure 2. If we look at \bar{L}_r based on a common length, say L_x , we have $\bar{L}_r = \frac{1}{2\pi}$ in the former and $\bar{L}_r = \frac{L_y}{2\pi L_x}$ in the latter. Again, numerical solutions are expected to differ by the amount of dissipation. Moreover, a hyperbolic solver may encounter various problems if the dissipation is too low (or too large), which can be the case for $\bar{L}_r = \frac{L_y}{2\pi L_x}$ with $L_y/L_x \to \infty$, for instance. This rectangular case raises an important question: Would the value of $\frac{1}{2\pi}$ be optimal for a rectangle with a unit length or width, or something else? This is a very important question. For each reference length chosen, we would need to define a value of \bar{L}_r . For a square domain, $\frac{1}{2\pi}$ is known to be optimal, but it may not be the same for non-square domains.

It follows that there are two important points to consider. First, in order to remove the ambiguity and achieve the numerical similarity, we need to define a unique reference length L for a given problem. Second, for optimal performance of a numerical scheme, an optimal value of \bar{L}_r must be identified for a chosen reference length. Both taken into consideration, a practical approach would be to identify a reference length L that will make $L_r/L = \frac{1}{2\pi}$ optimal for a given problem; we call it an optimal reference length and denote by L_{opt} . This approach is further discussed in the next section.

4 Optimal Reference Length L for Rectangular Domain

We begin by deriving the optimal formula (4) for an arbitrary square domain. Following Ref.[7], consider a discrete Fourier mode defined on a Cartesian grid of spacing h,

$$\mathbf{U}_0 \exp\{i(\beta_x x/h + \beta_y y/h)\},\tag{19}$$

where $\mathbf{U}_0 = [u_0, p_0, q_0]$ is a vector of amplitudes, β_x and β_y are the frequencies in x and y directions, respectively, and $i = \sqrt{-1}$. Here, we consider a grid in a square domain of size L: $(x, y) \in [0, L] \times [0, L]$. The grid has a spacing h = L/N, where N is the number of grid spacings (or cells) in each coordinate direction. See Figure 3 for an example of N = 8. The smoothest mode is given by the frequency $\beta_x = \beta_y = \pi h/L = \pi/N$. Substitute the Fourier mode into a first-order accurate version of a hyperbolic finite-volume discretization, and require that a pair of the eigenvalues be complex conjugates, so that Fourier modes propagate, to find the condition:

$$L_r \ge \frac{2h}{\beta(\beta+4)},\tag{20}$$

where $\beta = \sqrt{\beta_x^2 + \beta_y^2}$. See Appendix A for the derivation. Note that there is a typo in the direction of the inequality in Ref.[7]; it is corrected in the above. To ensure that all possible discrete modes will propagate, we



Figure 3: Regular grid in a square domain.

enforce the equality for the smoothest mode:

$$L_r = \frac{2L}{\pi(\pi h/L + 4)} = \frac{2L}{\pi(\pi/N + 4)} \approx \frac{L}{2\pi},$$
(21)

for a large N. It implies that the optimal formula (4) must be scaled by L for the dimensional L_r , or equivalently, it is the non-dimensional formula that must be given a value of $\frac{1}{2\pi}$:

$$\bar{L}_r = \frac{L_r}{L} = \frac{1}{2\pi}.$$
(22)

Furthermore, it can also be interpreted that the side length L is the optimal reference length for a square domain since it scales L_r to make $L_r/L = \frac{1}{2\pi}$ optimal.

It is noted, before proceeding, that the requirement for error propagation ensures that the discrete operator (and the differential operator as $h \to 0$) remains hyperbolic. Error propagation is typically much faster than error damping as is well known for advection schemes, especially for smooth components, which are the main cause of slow asymptotic convergence of conventional diffusion schemes [20]; the hyperbolic diffusion method eliminates this slow convergence completely.

Next, consider a rectangular domain: $(x, y) \in [0, L_x] \times [0, L_y]$. For simplicity, we assume a Cartesian grid with a uniform spacing $h = L_x/N_x = L_y/N_y$, where N_x and N_y are the numbers of cells in x- and y-coordinate directions, respectively. Following a similar derivation as in the square domain case, we arrive at the same condition as in Equation (20). In the square domain case, we enforced the condition for the smoothest mode in each coordinate direction, which is the same: $\beta_x = \beta_y = \pi/N$. We now have different smoothest frequencies: $\beta_x = \pi/N_x$ and $\beta_y = \pi/N_y$. Then, it would be reasonable to take the smooth-smooth combination $(\beta_x, \beta_y) = (\pi/N_x, \pi/N_y)$:

$$\beta = \sqrt{\beta_x^2 + \beta_y^2} = \frac{\pi}{\tilde{N}}, \quad \tilde{N} = \frac{1}{\sqrt{1/N_x^2 + 1/N_y^2}} = \frac{1}{h\sqrt{1/L_x^2 + 1/L_y^2}},$$
(23)

to enforce the condition (20):

$$L_r = \frac{2h}{\pi/\tilde{N}(\pi/\tilde{N}+4)}.$$
(24)

Assuming $1/\tilde{N} \ll 1$, we define

$$L_r = \frac{1}{2\pi} h \tilde{N} = \frac{1}{2\pi} \frac{1}{\sqrt{1/L_x^2 + 1/L_y^2}},$$
(25)

and express it in terms of the volume $V = L_x L_y$ and the diagonal distance of the rectangle $Diag = \sqrt{L_x^2 + L_y^2}$

$$L_r = \frac{1}{2\pi} \left(\frac{V}{Diag} \right), \tag{26}$$

or

$$\bar{L}_r = \frac{L_r}{V/Diag} = \frac{1}{2\pi},\tag{27}$$

which suggests that the optimal reference length for a rectangular domain is given by

$$L_{opt} = \frac{V}{Diag}.$$
(28)

That is, L_{opt} is the reference length that scales L_r to make $L_r/L_{opt} = \frac{1}{2\pi}$ optimal. In the case of a square domain $L_x = L_y = L$, this reduces to $L_r/L_{opt} = \frac{1}{2\sqrt{2\pi}}$, which differs from Equation (22) by the factor $1/\sqrt{2}$. This factor arises from the fact that we have taken the smooth-smooth components instead of the smooth component in each coordinate direction to define L_r . As we will show later numerically, the difference is minor and does not have a significant impact on numerical results. Finally, it is interesting to note that Equation (28) can be expressed also in terms of the aspect ratio of the domain:

$$L_{opt} = \sqrt{\frac{V}{\frac{1}{\mathcal{AR}} + \mathcal{AR}}},\tag{29}$$

where $\mathcal{AR} = L_x/L_y$ or $\mathcal{AR} = L_y/L_x$.

Extending the analysis to arbitrary domains does not seem straightforward, and may not necessarily lead to a practical formula. In the next section, we propose a simple estimate of L_{opt} for an arbitrary domain by rearranging the formula (28) into a form that can be easily evaluated for a given computational grid.

Estimate of L_{out} for Arbitrary Domain $\mathbf{5}$

A possible approach to defining L_{opt} for an arbitrary domain would be to directly apply the rectangular formula (28). However, the diagonal distance, Diag, cannot be clearly defined for an arbitrary domain. To devise a practical formula, we express Equation (28) in terms of the volume V and the perimeter of the domain ℓ : $Diag^2 = \ell^2/4 - 2V > 0$, which can be easily computed for a given grid. The resulting formula is

$$L_{opt} = \frac{V}{\sqrt{\ell^2/4 - 2V}}.$$
(30)

This estimate will, at least, guarantee that $\bar{L}_r = \frac{1}{2\pi}$ is optimal for a rectangular domain in two dimensions, and account for the aspect ratio of the domain. In this work, we focus on this simple estimate, and demonstrate that it serves our purpose rather well although the optimality of $\bar{L}_r = \frac{1}{2\pi}$ is not guaranteed exactly for domains other than rectangles. Moreover, it has a very convenient feature that it can be computed inside a code for a given grid without any detailed information about the geometry of the domain of interest as well as the coordinate unit used.

It should be noted that this reference-length estimate is important even when the problem is already nondimensionalized by a user-defined reference length L_{user} . In this case, one would need to set the value of

$$\bar{L}_r = \frac{L_r}{L_{user}},\tag{31}$$





Figure 5: Boundary face, where a boundary condition is imposed through the right state \mathbf{U}_R . A_B denotes the magnitude of the boundary face. The interior state \mathbf{U}_L is computed by a linear extrapolation from the centroid j.

Figure 4: Triangular grid with a neighbor k, the face midpoint t m, and the face normal vector. The LSQ gradient is computed is with the neighbors and their face-neighbors for robustness.

but this should not be be given a value of $\frac{1}{2\pi}$ unless the domain is a unit square. It should be defined as

$$\bar{L}_r = \frac{\bar{L}_{opt}}{2\pi},\tag{32}$$

where \bar{L}_{opt} is computed for a given grid in the grid unit, i.e., $\bar{L}_{opt} = L_{opt}/L_{user}$, so that we have

$$\frac{L_r}{L_{opt}} = \frac{1}{2\pi}.$$
(33)

The scaling in Equation (32) will be very important for the hyperbolic method applied to a system already non-dimensionalized by a certain reference length; the performance of a hyperbolic solver may not be optimal and can cause accuracy and convergence problems unless the relaxation length is scaled by \bar{L}_{opt} as in Equation (32).

Finally, it is remarked that the proposed estimate for L_{opt} is specific to diffusion problems in a bounded domain. It remains to be investigated for other systems such as the hyperbolic Navier-Stokes systems [12, 13, 14], which is left as future work.

6 Discretization and Solver

6.1 Discretization

In this work, we solve the dimensional hyperbolic diffusion system (2) by a second-order cell-centered finitevolume scheme with $L_r = \frac{L}{2\pi}$ for various choices for L. The spatial residual (11) over each cell, with a physical time derivative included for unsteady problems, is defined by the flux integral evaluated by the mid-point rule with the upwind flux (10) based on the left and right solutions linearly extrapolated from the two cells sharing the face-midpoint with the gradients computed by an unweighted linear least-squares (LSQ) fit over the face-neighbors and their face neighbors:

$$\mathbf{U}_{L} = \mathbf{U}_{j} + \frac{1}{2} \nabla \mathbf{U}_{j} \cdot \Delta \mathbf{x}_{jm}, \qquad \mathbf{U}_{R} = \mathbf{U}_{k} + \frac{1}{2} \nabla \mathbf{U}_{k} \cdot \Delta \mathbf{x}_{km},$$
(34)

where $\mathbf{U} = (u, p, q)$. See Figure 4. In the hyperbolic method, the LSQ gradient computation can be avoided for the variable u by using the gradient variables (p, q) as ∇u :

$$u_L = u_j + \frac{1}{2} \left(p_j, q_j \right) \cdot \Delta \mathbf{x}_{jm}, \qquad u_R = u_k + \frac{1}{2} \left(p_k, q_k \right) \cdot \Delta \mathbf{x}_{km}, \tag{35}$$

This construction is called Scheme-II [7]. Here, it provides more accurate gradients for the linear reconstruction of u, and also contributes to a computational saving since we do not need to compute LSQ gradients for u. In this study, Scheme-II is used for all problems. Both Dirichlet and Neumann boundary conditions are considered, and implemented weakly by the upwind flux with the right (boundary) state specified by the boundary condition as illustrated in Figure 5. In the Dirichlet case, we define the boundary state as

$$\mathbf{U}_R = (2u_B - u_L, p_L, q_L),\tag{36}$$

where u_B is the solution given at the boundary by the Dirichlet condition, so that the average of the boundary state and the interior state u_L gives u_B . In this case, the numerical flux (10) becomes

$$\Phi_{jk}(\mathbf{U}_{L},\mathbf{U}_{R}) = \nu \begin{bmatrix} -(p_{n})_{L} - \frac{1}{L_{r}}(u_{B} - u_{L}) \\ -\frac{1}{L_{r}^{2}}u_{B}n_{x} \\ -\frac{1}{L_{r}^{2}}u_{B}n_{y} \end{bmatrix}, \qquad (37)$$

where $(p_n)_L = (p_L, q_L) \cdot \mathbf{n}_B$ and $\mathbf{n}_B = (n_x, n_y)$ is the unit vector normal to the boundary face. In the Neumann case, we define the boundary state as

$$\mathbf{U}_{R} = (u_{L}, [2(p_{n})_{B} - (p_{n})_{L}]n_{x}, [2(p_{n})_{B} - (p_{n})_{L}]n_{y}),$$
(38)

where $(p_n)_B$ is the normal gradient given at the boundary by the Neumann condition. In this way, the normal gradient condition is satisfied by the averaged state $(\mathbf{U}_L + \mathbf{U}_R)/2$ projected along \mathbf{n}_B . The numerical flux (10) then becomes

$$\Phi_{jk}(\mathbf{U}_L, \mathbf{U}_R) = \nu \begin{bmatrix} -(p_n)_B \\ -\frac{1}{L_r^2} u_L n_x - \frac{1}{L_r} [(p_n)_B - (p_n)_L] n_x \\ -\frac{1}{L_r^2} u_L n_y - \frac{1}{L_r} [(p_n)_B - (p_n)_L] n_y \end{bmatrix}.$$
(39)

See Refs.[11, 21] for further details of cell-centered finite-volume schemes for the hyperbolic diffusion system, including higher-order variants.

6.2 Solver

The resulting global system of residual equations,

$$0 = \mathbf{Res}(\mathbf{U}_h),\tag{40}$$

where \mathbf{U}_h denotes the global solution vector, is solved by an implicit defect-correction solver:

$$\mathbf{U}_{h}^{k+1} = \mathbf{U}_{h}^{k} + \Delta \mathbf{U}_{h},\tag{41}$$

$$\frac{\partial \overline{\mathbf{Res}}}{\partial \mathbf{U}} \Delta \mathbf{U}_h = -\mathbf{Res}(\mathbf{U}_h^k),\tag{42}$$

where k is the iteration counter, and the Jacobian $\partial \overline{\text{Res}} / \partial \mathbf{U}$ is the exact differentiation of the first-order version of the residual $\overline{\text{Res}}$ (zero LSQ gradients), which is defined by the following left and right values at each face:

$$p_{L} = p_{j}, \qquad p_{R} = q_{k},$$

$$p_{L} = p_{j}, \qquad q_{R} = q_{k},$$

$$u_{L} = u_{j} + \frac{1}{2} (p_{j}, q_{j}) \cdot \Delta \mathbf{x}_{jm}, \qquad u_{R} = u_{k} + \frac{1}{2} (p_{k}, q_{k}) \cdot \Delta \mathbf{x}_{km}.$$

$$(43)$$

Below, for simplicity, we present the Jacobian obtained with $\mathbf{U}_L = \mathbf{U}_j$ and $\mathbf{U}_R = \mathbf{U}_k$, which should work equally well in the implicit solver (in our implementation, the Jacobian matrix contains contributions from the gradient terms in u_L and u_R). The diagonal block at a cell j is given by

$$\left(\frac{\partial \overline{\mathbf{Res}}}{\partial \mathbf{U}}\right)_{jj} = \frac{\partial \overline{\mathbf{Res}}_j}{\partial \mathbf{U}_j} = \sum_{k \in \{k_j\}} \frac{\partial \Phi_{jk}}{\partial \mathbf{U}_j} A_{jk} - \frac{\partial \mathbf{s}_j}{\partial \mathbf{U}_j} V_j, \tag{44}$$

where

$$\frac{\partial \mathbf{\Phi}_{jk}}{\partial \mathbf{U}_{j}} = \frac{\nu}{2} \begin{bmatrix} 1/L_{r} & -n_{x} & -n_{y} \\ -n_{x}/L_{r}^{2} & n_{x}^{2}/L_{r} & n_{x}n_{y}/L_{r} \\ -n_{y}/L_{r}^{2} & n_{y}n_{x}/L_{r} & n_{y}^{2}/L_{r} \end{bmatrix}, \quad \frac{\partial \mathbf{s}_{j}}{\partial \mathbf{U}_{j}} = \begin{bmatrix} -(\partial_{t}u)_{j}/\partial u_{j} & 0 & 0 \\ 0 & -1/T_{r} & 0 \\ 0 & 0 & -1/T_{r} \end{bmatrix}, \quad (45)$$

where $(\partial_t u)_j$ denotes the physical time derivative at j (see Section 7.2), and the off-diagonal block is given by

$$\left(\frac{\partial \overline{\mathbf{Res}}}{\partial \mathbf{U}}\right)_{jk} = \frac{\partial \overline{\mathbf{Res}}_j}{\partial \mathbf{U}_k} = \frac{\partial \Phi_{jk}}{\partial \mathbf{U}_k} A_{jk},\tag{46}$$

where

$$\frac{\partial \Phi_{jk}}{\partial \mathbf{U}_k} = \frac{\nu}{2} \begin{bmatrix} -1/L_r & -n_x & -n_y \\ -n_x/L_r^2 & -n_x^2/L_r & -n_x n_y/L_r \\ -n_y/L_r^2 & -n_y n_x/L_r & -n_y^2/L_r \end{bmatrix}.$$
(47)

For boundary fluxes, the flux Jacobian contributes to the diagonal block: in the Dirichlet case,

$$\frac{\partial \mathbf{\Phi}_{jk}}{\partial \mathbf{U}_j} = \nu \begin{bmatrix} 1/L_r & -n_x & -n_y \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$
(48)

and in the Neumann case

$$\frac{\partial \mathbf{\Phi}_{jk}}{\partial \mathbf{U}_{j}} = \frac{\nu}{L_{r}} \begin{bmatrix} 0 & 0 & 0\\ -n_{x}/L_{r} & n_{x}^{2} & n_{x}n_{y}\\ -n_{y}/L_{r} & n_{y}n_{x} & n_{y}^{2} \end{bmatrix}.$$
(49)

In the linearized system (42), the pseudo-time derivatives have been dropped, but the physical time derivative is included as a source term for unsteady problems. The linear system is relaxed by the sequential Gauss-Seidel scheme until the linear residual is reduced by half or the specified maximum of 500 is reached. The implicit solver is taken as converged when the maximum of the L_1 norms of the residual components is reduced by ten orders of magnitude for steady problems, and three orders of magnitude for unsteady problems. All residual convergence histories shown later are normalized by an initial residual norm.

7 Numerical Results

This section presents numerical results for demonstrating scale-invariance of the hyperbolic diffusion scheme, and the optimal performance by the proposed optimal reference length, L_{opt} . We focus on solving the dimensional heat equation in various grid units, but it is equivalent to solving a non-dimensional equation for various user-defined reference lengths. As discussed in Section 5, the non-dimensional relaxation length should still be scaled by \bar{L}_{opt} , which is evaluated for a given non-dimensionalized grid, for optimal performance. Below, we compare various choices for L in $L_r = \frac{L}{2\pi}$, but it is equivalent to comparing various choices for L/L_{user} in $L_r/L_{user} = \frac{L/L_{user}}{2\pi}$, where L_{user} is a dimensional length corresponding to 1.0 in the grid unit.

In this paper, we mainly consider linear problems, where ν is a constant. But the method can be applied to nonlinear problems by the non-linear hyperbolic formulation as described in Refs.[22, 2]. It can be applied also to a tensor coefficient diffusion equation [11]. To demonstrate the effect of the proposed reference length on nonlinear problems, we consider a nonlinear diffusion equation in Section 7.1.4.

7.1 Steady Heat Conduction Problem

We consider a steady heat conduction problem in a bounded domain in the coordinates defined in meter for the exact temperature distribution measured in Kelvin:

$$u(x,y) = \frac{\sinh(\pi x)\sin(\pi y) + \sinh(\pi y)\sin(\pi x)}{\sinh(\pi)},\tag{50}$$

which satisfies the steady diffusion equation (8). It is emphasized that the exact solution is defined in the coordinates measured in meter, and its distribution over the domain will not change when the coordinate unit is changed. The domain defined in meter is taken as a baseline, and we will investigate the behavior of the hyperbolic scheme over the same domain expressed in other units. The problem is taken as a Dirichlet problem with the temperature at the boundary is specified by the exact solution, or as a Neumann problem with the temperature gradient normal to the boundary is specified. The diffusion coefficient ν has no impact on the steady problem and also on numerical solutions; therefore, we simply set $\nu = 1$.

For all steady problems, initial solutions are set to be randomly-perturbed exact solutions. At each cell, exact solution values are multiplied by a factor (1 + 0.25rdn), where rdn is a random number in the range [-1, 1]. This way of adding perturbation is important to investigate scale-independent iterative convergence. The variable u is scale-invariant, but p and q are not. Simply adding random perturbation to p and q would lead to completely different amounts of perturbation for different scalings. Multiplying exact values by (1+0.25rdn) will ensure that the perturbation is consistently scaled and generate similar settings for different scalings.

In the next section, we consider a square domain to illustrate the issue of scaling and the effect of the optimal relaxation length. In the second section, we investigate the optimal nature of formulas for a rectangular domain. In the third section, we consider a non-square domain to investigate the choice of the reference length and its impact on accuracy and iterative convergence. Finally, we consider a nonlinear problem.

7.1.1 Square Domain

A unit square domain in meter is triangulated randomly to yield three levels of grids: Grid1, Grid2, Grid3, with 2048, 8192, and 32768 cells, respectively. Grid1 is shown in Figure 6. To investigate the effect of coordinate units, we scale them by a factor D: D = 0.001 and D = 1000, corresponding to the units [km] and [mm], respectively. The resulting grids are referred to as Grid1[m], Grid1[km], Grid1[mm], and similarly for Grid2 and Grid3. The temperature should have the same distribution for all the grid units. See Figure 7. For this problem, we compare the following choices for L in $L_r = \frac{L}{2\pi}$:

$$L = 1, \quad L_{opt} \left(=\frac{D}{\sqrt{2}}\right), \quad \frac{D}{2}, \quad 2D.$$
 (51)

The choice L = 1 corresponds to a problematic case, and others should yield scale-invariant solutions. In terms of the non-dimensionalzed length defined by the side length of the square: $\bar{L}_r = L_r/D$, we are comparing, respectively,

$$\bar{L}_r = \frac{1}{2\pi D}, \quad \frac{1}{2\sqrt{2}\pi}, \quad \frac{1}{4\pi} \quad \frac{1}{\pi}.$$
 (52)

Therefore, it varies across the grid units for L = 1 with the optimal value $\bar{L}_r = \frac{1}{2\pi}$ achieved only for D = 1. $L = L_{opt}$ corresponds to $1/\sqrt{2}$ times the optimal value; $L = \frac{D}{2}$ and L = 2D correspond to half and twice the optimal value, respectively.

Convergence histories for Grid3 are shown in Figure 8. Results for Grid1 and Grid2 are very similar to these results, and therefore not shown. As expected, the convergence histories are perfectly identical (within round-off) in each of the cases: $L = L_{opt}$, L = D/2, and L = 2D. For Grid3[m], comparing L = 1 (optimal) and $L = L_{opt}$, we find that convergence histories are almost identical, thus indicating that the factor $1/\sqrt{2}$ has a very minor impact on the convergence. For other grids, Grid3[km] and Grid3[mm], the choice L = 1 yields convergence histories widely varying among the grid units. Figure 8(b) shows that the linear relaxation scheme cannot reduce the linear residual by half within the maximum of 500 on Grid3[km]. The solver converges very slowly; the residual is reduced by six orders of magnitude in 2000 iterations. Also, for Grid3[mm], it takes more and more linear relaxations as the iteration continues and eventually diverges. Another important observation is that $L = L_{opt}$ takes the least number of linear relaxations among all. This indicates that the formulas $\bar{L}_r = \frac{1}{2\sqrt{2}\pi}$ as well as $\bar{L}_r = \frac{1}{2\sqrt{2}\pi}$ can be considered as optimal for a square-domain problem, as expected from the analysis.

Error convergence results are shown in Figure 9. Results are shown for the temperature u[K] and the gradient variable $p(=\partial_x u)[K/m]$; results for the other gradient variable q are very similar to those for p, and therefore not shown. To make the comparison easier, we have scaled the gradient errors by D; therefore, the unit is [K/m] for all. It can be seen that L = 1 exhibits strong dependence on the coordinate unit (the solver diverged on Grid3[mm], and thus no data shown). However, as expected, the order of error convergence is second-order on all grid units as long as the solver converges. All other reference lengths do not show any dependence on the grid unit. These results show that the error levels are not strongly affected by the reference length in this problem.

Overall, these results confirm the importance of the relaxation length scaling to eliminate the scale-dependence from the numerical solution algorithm. It is also confirmed that the proposed reference length $L = L_{opt}$ yields nearly optimal results among others tested for a square domain.

7.1.2 Rectangular Domain

To test the formula (26) for a rectangular domain, we consider the same problem with a restricted domain. A rectangular domain is taken as $(x, y) \in [0, 0.1] \times [0, 1]$. The Dirichlet condition is applied at the boundary. The irregular triangular grids used in the previous section are mapped onto the rectangular domain by re-scaling the x-coordinate. See Figure 10. Note that the domain has the aspect ratio of 10, and also triangular cells have a similar aspect ratio. We compare the following choices for L:

$$L = 1, \quad L_{opt}, \tag{53}$$

for grids in meter only, in order to investigate the optimality for a rectangular domain.

Iterative convergence results are shown in Figure 11(a). It is observed that the choice $L = L_{opt}$ gives faster convergence. The choice L = 1 requires much more linear relaxations and slightly more implicit iterations to converge. Despite such variation in iterative convergence, no significant differences are observed in the error convergence for the solution variable. The error in the gradient variable is slightly larger for $L = L_{opt}$. See Figure 11(b). As a check, we have performed the same computations with an alternative rectangular domain: $(x, y) \in [0, 1] \times [0, 0.1]$, and obtained very similar results (not shown).

To confirm that the cell aspect ratio has no impact on the iterative convergence, we consider a very thin rectangular domain $(x, y) \in [0, 1] \times [0, 0.01]$ with uniform Cartesian grids. Three levels of Cartesian gris have been generated: Grid1, Grid2, Grid3, with 400×4 , 800×8 , 1600×16 cells, respectively. Grid1 is shown in Figure 12. Iterative convergence results are similar for all grids, and thus the result is shown only for Grid3 in Figure 13(a). As can be seen, the solver slows down significantly for L = 1, but converges rapidly for $L = L_{opt}$. Error levels are almost the same as shown in Figure 13(b). These results demonstrate that it is the domain aspect ratio, not the cell aspect ratio, that affects the iterative convergence.

7.1.3 Non-Square Domain

To explore non-regular domains, we consider a tube-like domain bounded by the curves $y = 0.25 \sin^4(2.5\pi x)$ and $y = 0.25 \sin^4(2.5\pi x) + 0.1$, the y-axis, and the vertical line at x = 1. Three levels of irregular triangular grids have been generated: Grid1, Grid2, Grid3 with 768, 3584, and 15360 cells, respectively. Grid1 is shown in Figure 14. These grids are scaled by a factor D: D = 0.001 and D = 1000, corresponding to the units [km]and [mm], respectively. The resulting grids are, again, referred to as Grid1[m], Grid1[km], Grid1[mm], and similarly for other levels. The same exact temperature distribution (50) is used, as shown in Figure 15, and the Dirichlet condition is applied at the boundary. For this problem, we compare the following choices for L in $L_r = \frac{L}{2\pi}$:

$$L = 1, \quad L_{opt} (= 0.05643412996384048D), \quad d(= 0.1D), \quad \ell(= 1.72758472307282D), \tag{54}$$

where d is the thickness of the tube domain at x = 0, and ℓ is the length of the tube. The latter has been computed numerically for a given grid. Also, L_{opt} is computed numerically inside a code from the volume V and the perimeter ℓ computed by summing up the element volumes and the length of boundary edges. The thickness and the length of the tube are very reasonable reference lengths that one would choose. However, the question being asked here is whether the formula $\bar{L}_r = L_r/L = \frac{1}{2\pi}$ is optimal (or yields the best results) for a tube of unit thickness, length, or L_{opt} . Recall that the optimal formula is not known for domains other than rectangles. Figure 16 shows iterative convergence histories for Grid3. Again, results for Grid1 and Grid2 are very similar to these results, and therefore not shown. As before, L = 1 shows significantly slow convergence for Grid3[km] and divergence for Grid3[mm]. Other reference lengths yield scale-invariant convergence as expected. Both $L = L_{opt}$ and L = d yield very fast convergence; L_{opt} requires slightly more linear relaxations but less implicit iterations. Figure 17 shows error convergence results. L = 1 exhibits wildly-varying error levels for grids with different units; it diverged for Grid3[mm], and thus no data are shown. Others do not show any scale-dependence, and produce almost the same results except that $L = \ell$ yields smaller errors in the gradient variable. These results indicate that $L = L_{opt}$ can serve as a good and practical approximation to an optimal value that is not known for this problem.

7.1.4 Nonlinear Example in Non-Square Domain

To demonstrate the impact of $L = L_{opt}$ on nonlinear problems, we consider the following problem:

$$0 = \partial_x (\nu \partial_x u) + \partial_y (\nu \partial_y u) - g, \tag{55}$$

where

$$\nu = 1 + u^2, \quad g = (1 + b^2) \left(\partial_{xx} b + \partial_{yy} b \right) + 2b \left[(\partial_x b)^2 + (\partial_y b)^2 \right], \quad b = \sin(\pi x) \sin(\pi y), \tag{56}$$

which has the exact solution:

$$u = \sin(\pi x)\sin(\pi y). \tag{57}$$

The problem is solved in the same non-square domain considered in the previous section. The same grids are used with D: D = 0.001, D = 1, and D = 1000. See Figure 14 for Grid 1. To investigate the scale invariance, the exact solution and the forcing term g are always evaluated in the coordinates measured in meter (D = 1), and the first and second derivatives of b are, then, multiplied by 1/D and $1/D^2$, respectively.

The second-order cell-centered finite-volume discretization is constructed based on a nonlinear hyperbolic formulation [13, 22, 23]:

$$\mathbf{P}^{-1}\partial_{\tau}\mathbf{u} + \partial_{x}\mathbf{f} + \partial_{y}\mathbf{g} = \mathbf{s},\tag{58}$$

where

$$\mathbf{P}^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & T_r & 0 \\ 0 & 0 & T_r \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u \\ p \\ q \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} -\nu p \\ -u \\ 0 \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} -\nu q \\ 0 \\ -u \end{bmatrix}, \quad \mathbf{s} = \begin{bmatrix} -g \\ -p \\ -q \end{bmatrix}.$$
(59)

The relaxation time is still given by $T_r = L_r^2/\nu$ with $L_r = \frac{L}{2\pi}$, but it is now a function of the solution because $\nu = \nu(u)$. The residual equation at a cell j is given by

$$\mathbf{Res}_j = \sum_{k \in \{k_j\}} \mathbf{P}_j \left[\mathbf{\Phi}_{jk} A_{jk} - \mathbf{s}_j V_j \right] = 0, \tag{60}$$

where the pseudo time term has been dropped, and \mathbf{P}_{i} is evaluated by u_{i} . The numerical flux is given by

$$\Phi = \frac{1}{2} \left[(\mathbf{f}_L, \mathbf{g}_L) + (\mathbf{f}_R, \mathbf{g}_R) \right] \cdot \mathbf{n}_{jk} - \frac{1}{2} \hat{\mathbf{P}}^{-1} \left| \hat{\mathbf{P}} \hat{\mathbf{A}}_n \right| (\mathbf{u}_R - \mathbf{u}_L)$$

$$= \frac{1}{2} \begin{bmatrix} -\nu_L(p_L, q_L) \cdot \mathbf{n}_{jk} - \nu_R(p_R, q_R) \cdot \mathbf{n}_{jk} \\ -(u_L + u_R)n_x \\ -(u_L + u_R)n_y \end{bmatrix} - \frac{1}{2} \sqrt{\frac{\hat{\nu}}{\hat{T}_r}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \hat{T}_r n_x^2 & \hat{T}_r n_x n_y \\ 0 & \hat{T}_r n_y n_x & \hat{T}_r n_y^2 \end{bmatrix} \begin{bmatrix} u_R - u_L \\ p_R - p_L \\ q_R - q_L \end{bmatrix}, (61)$$

where $\mathbf{f}_{L/R} = \mathbf{f}(\mathbf{u}_{L/R}), \ \mathbf{g}_{L/R} = \mathbf{g}(\mathbf{u}_{L/R}), \ \hat{\mathbf{A}}_n = (\partial \hat{\mathbf{f}} / \partial \mathbf{u}) n_x + (\partial \hat{\mathbf{g}} / \partial \mathbf{u}) n_y, \ \hat{\mathbf{f}} = (-\hat{\nu}p, -u, 0), \ \hat{\mathbf{g}} = (-\hat{\nu}q, 0, -u),$ $\nu_{L/R} = \nu(u_{L/R}), \ \hat{\nu} = \nu(\hat{u}), \ \hat{u} = (u_L + u_R)/2, \ \hat{T}_r = L_r^2/\hat{\nu}.$ The left and right states, \mathbf{u}_L and \mathbf{u}_R , are computed by the linear extrapolation as described in Section 6.1. Further details on the discretization of the nonlinear formulation can be found in Refs.[13, 22, 23]. The resulting global system of the residual equations is solved by the implicit solver as described in Section 6.2, but the Jacobian is different from those in Section 6.2: it is derived by exactly differentiating the first-order accurate version of the nonlinear diffusion residual (60) with respect to the discrete unknowns. Also, the residual Jacobian is re-computed at every iteration in the nonlinear case, whereas it is computed only once at the beginning of the implicit iteration in the linear case.

Results are shown in Figure 18. Only the results for $L = L_{opt}$ are shown because the solver diverged for L = 1 and $L = \ell$, and the results for L = d are very similar to those for $L = L_{opt}$. As can be seen from Figure 18(a) and Figure 18(b), all three curves, corresponding to three different units, overlap one another, indicating that the iterative convergence and error convergence are independent of the grid unit. The failure with L = 1 and $L = \ell$ indicates that a proper choice of the reference length is critically important for nonlinear diffusion problems.

7.2 Unsteady Heat Conduction Problem in Non-Square Domain

To demonstrate the scale-invariance of the hyperbolic method for unsteady problems, we consider the unsteady diffusion equation (1). The problem is directly relevant to heat conduction problems, which are typically given as

$$\partial_t (\rho C_p u) = \kappa (\partial_{xx} u + \partial_{yy} u), \tag{62}$$

where $\rho[kg/m^3]$ is the density of the material of interest, $C_p = C_p(u)[J/(kg \cdot K)]$ is the specific heat capacity, and $\kappa[W/(m \cdot K)]$ is the heat conductivity. Assuming ρ is constant, we discretize the physical time derivative by the variable-time-step backward Euler scheme:

$$0 = \frac{\kappa}{\rho} (\partial_{xx}u + \partial_{yy}u) - \frac{\alpha u C_p(u) + \alpha_n u^n C_p(u^n) - \alpha_{n-1}u^{n-1}C_p(u^{n-1})}{\Delta t},$$
(63)

where

$$\alpha = \frac{1}{\Delta t_n}, \quad \alpha_n = -\frac{1}{\Delta t_n}, \quad \alpha_{n-1} = 0, \quad \Delta t_n = t^{n+1} - t^n, \tag{64}$$

for first-order accuracy, and

$$\alpha = \frac{1}{\Delta t_n} \left(1 + \frac{\Delta t_n}{\Delta t_n + \Delta t_{n-1}} \right), \quad \alpha_n = -(\alpha + \alpha_{n-1}), \quad \alpha_{n-1} = \frac{\Delta t_n}{\Delta t_{n-1} \left(\Delta t_n + \Delta t_{n-1} \right)}, \quad \Delta t_{n-1} = t^n - t^{n-1}, \quad (65)$$

for second-order accuracy. The former is used in the first time step to start up the unsteady simulation, with one-tenth Δt to reduce the effect of first-order accuracy. Variable time steps are used only at the beginning and the end of each calculation (to finish the computation precisely at the final time if necessary). Both schemes are unconditionally stable, and therefore the time step can be determined solely based on the resolution requirement, not limited by numerical stability. The unsteady heat equation can be formulated as a pseudo-time problem:

$$\overline{C}_p \partial_\tau u = \frac{\kappa}{\rho} (\partial_{xx} u + \partial_{yy} u) - \frac{\alpha u C_p(u) + \alpha_n u^n C_p(u^n) - \alpha_{n-1} u^{n-1} C_p(u^{n-1})}{\Delta t}, \tag{66}$$

where \overline{C}_p is a reference capacity (e.g., at a standard temperature); or by using the thermal diffusivity $\nu = \kappa/(\rho \overline{C}_p)[m^2/s]$, as

$$\partial_{\tau} u = \nu (\partial_{xx} u + \partial_{yy} u) - s(u), \tag{67}$$

where s(u) represents the discretized physical time derivative normalized by the reference capacity. Then, the steady solution to the pseudo-time problem corresponds to the temperature at the next time level, u^{n+1} . In this form, the hyperbolic method is directly applied as discussed earlier. As in steady problems, we drop the pseudo time derivatives and solve the resulting globally coupled system of discrete equations by the implicit solver until the maximum of the L_1 norm of the residual components is reduced by three orders of magnitude. It is important to note that the thermal diffusivity ν depends on the coordinate unit, and therefore it has to be scaled for a chosen coordinate unit. Note also that we solve the hyperbolic system (2) with the pseudo time terms dropped, and thus the target system is equivalent to Equation (62) and shares the well-known minimum/maximum principle of the heat equation, at least, in the differential-equation level. The problem is taken over a domain bounded by two co-centric circles of radii $R_i = 0.75[m]$ and $R_o = 1.0[m]$. In order to be able to investigate discretization errors, we consider an idealized problem $C_p = 1.0[J/(kg \cdot K)]$ and $\nu = 1.0[m^2/s]$. The temperature at the inner and outer circle are specified as

$$u_i(t) = \frac{1}{t+0.01} \exp\left(-\frac{R_i^2}{4\nu(t+0.01)}\right), \quad u_o(t) = \frac{1}{t+0.01} \exp\left(-\frac{R_o^2}{4\nu(t+0.01)}\right), \tag{68}$$

so that the exact solution is given by

$$u(x,t) = \frac{1}{t+0.01} \exp\left(-\frac{x^2+y^2}{4\nu(t+0.01)}\right),\tag{69}$$

with the initial condition set at $t_0 = 0$ and the final time at $t_f = 0.15$. Dirichlet condition is applied at the inner and outer circle, and Neumann condition is applied at y = 0 as $\partial u/\partial y = 0$.

Four levels of irregular triangular grids have been generated: Grid1, Grid2, Grid3, Grid4 with 896, 3840, 15872, and 64512 cells, respectively. Grid1 is shown in Figure 19. As before, these grids are scaled by a factor D for D = 0.001 and D = 1000, corresponding to the units [km] and [mm], respectively. The resulting grids are, again, referred to as Grid1[m], Grid1[km], Grid1[mm], and similarly for other levels. The thermal diffusivity ν is also scaled as $\nu = 10^{-6} [km^2/s]$ and $\nu = 10^6 [mm^2/s]$. But the temperature field must be identical for all grid units. See Figure 20. The time step is chosen as $\Delta t = 0.001$, requiring 151 time steps to reach the final time. Note that the time step is small enough to be able to observe error convergence in space, but much larger than those required by explicit time-stepping schemes, which would be restricted by the well-known condition: $\Delta t = O(h_{min}^2)$, corresponding to $O(10^{-5})$ even on the coarsest grid.

For this problem, we compare the following choices:

$$L = L_{opt} (= 0.248970545904566D), \quad d(=D), \quad \ell(=\pi D), \tag{70}$$

where d is the width of the domain, and ℓ is the length of the outer semi-circle boundary. The results are compared also with a conventional diffusion scheme [24], which is denoted here by Alpha4/3. The residual of Alpha4/3 at the cell j, which is the discrete approximation to the diffusion equation (67) with the pseudo time term dropped, is given by

$$Res_j = \sum_{k \in \{k_j\}} \phi_{jk} A_{jk} - s(u_j) V_j = 0,$$
(71)

where $s(u_j)$ is the physical time derivative discretized in time by the backward Euler schemes as described earlier. The numerical flux ϕ_{jk} is given by

$$\phi_{jk} = \sum_{k \in \{k_j\}} \nu \left[\frac{\nabla u_j + \nabla u_k}{2} \cdot \mathbf{n}_{jk} + \frac{\alpha}{|\mathbf{e}_{jk} \cdot \mathbf{n}_{jk}|} (u_R - u_L) \right],\tag{72}$$

where ∇u_j and ∇u_k are the LSQ gradients at the cells j and k, respectively, u_L and u_R are linearly extrapolated solutions as in Equation (34), $\mathbf{e}_{jk} = (x_k - x_j, y_k - y_j)$, and α is a parameter that controls the amount of highfrequency damping and the optimal value $\alpha = 4/3$ is used, which yields fourth-order accuracy on uniform grids [24]. This scheme has been found very robust for unstructured grids [25] and used in a practical threedimensional implicit viscous solver [26]. An implicit solver is employed to solve the global system of residual equations over each physical time step with the Jacobian constructed based on the damping term only [26, 27]. The solver parameters such as the tolerance and the maximum number of linear relaxations are set exactly the same values as explained for the implicit hyperbolic solver at the end of Section 6.2. See Ref.[27] for further details.

Iterative convergence histories are shown in Figure 21 for the first 10 time steps. Sub-iterations are indicated as fractions over each time step. Note that convergence histories obtained with three different grid units are plotted with different symbols; they overlap one another due to the scale-invariance property. All results are similar in the sense that 10-15 sub-iterations are required to reduce the residual by three orders of magnitude, except that Alpha4/3 and L_{opt} achieve nearly a constant residual-reduction at each time step, and the residual increases at every first sub-iteration from the second time step in Alpha4/3. Shown in Figure 22(a) is the number of linear relaxation per iteration. Typically, the number of linear relaxations per iteration is of $O(N_c)$ in conventional methods, where N_c is the total number of cells, and of $O(\sqrt{N_c})$ in the hyperbolic method

	Grid1	Grid2	Grid3	Grid4
$L = L_{opt}$	9.2	64.9	400.3	2677.9
L = d	11.7	97.4	775.8	5203.6
$L=\ell$	21.5	227.9	1708.5	12548.0
Alpha4/3	9.6	143.0	1677.1	23407.4

Table 1: CPU time (sec) required to perform an unsteady simulation (D = 1).

by the elimination of second derivatives. The results indicate that the conventional solver took slightly more and the hyperbolic solver less relaxations than expected over the grids tested. Actual CPU time required for each unsteady computation is compared in Table 1 and Figure 22(b). Results are nearly as expected: the total CPU time grows like $N_c^{1.5}$ for the hyperbolic solver and N_c^2 for the conventional solver. The fastest convergence obtained by $L = L_{opt}$ demonstrates the effectiveness of the proposed reference length. The speedup in computing time is $O(\sqrt{N_c})$ and grows with grid refinement: the hyperbolic solver becomes more and more efficient for finer grids. Therefore, the hyperbolic solver, even if it does not have optimal performance, overwhelms the conventional solver eventually as the grid becomes fine enough. See $L = d, \ell$ in Table 1 or Figure 22(b). This O(1/h)-type convergence acceleration is a well-known feature of the hyperbolic method.

As is well known for the hyperbolic method, the faster computing time comes with superior accuracy in the gradients. Figure 23 shows error convergence results measured at the final time. Second-order error convergence is observed for the hyperbolic diffusion scheme for both solution and gradient variables. On the other hand, the conventional scheme achieves second-order accuracy in the solution, but first-order accuracy for the *x*-derivative computed by the LSQ method as typical on unstructured grids. All results have been confirmed to be scale-invariant.

8 Concluding Remarks

In this paper, we have discussed a problem encountered in solving a heat conduction problem in the dimensional form by the hyperbolic method. It has been shown that the relaxation length associated with the hyperbolic diffusion system is a dimensional quantity and must be properly scaled by a reference length for a given problem in order to guarantee scale-invariance of numerical solutions. For each reference length chosen, an optimal value of the corresponding non-dimensionalized relaxation length needs to be found to achieve optimal performance of a hyperbolic solver. The optimal value is given by $(2\pi)^{-1}$ for a unit square domain, but it is not necessarily optimal for a general domain. As a practical approach, we have proposed to determine a reference length such that $(2\pi)^{-1}$ would be close to optimal. This reference length is called an optimal reference length. To develop this approach, we have derived an optimal reference length for a general rectangular domain, and arranged it into a form that can be easily evaluated for a given computational grid of arbitrary domains. Numerical results have shown that the relaxation length L_r scaled by the optimal reference length vields scale-invariant solutions and fast iterative convergence for both steady and unsteady heat conduction problems, including a steady nonlinear problem, for non-rectangular domains. For an unsteady problem, the hyperbolic solver has been demonstrated to yield significantly faster computing time than a conventional solver, with an additional advantage of one-order-higher accuracy in the gradients on unstructured grids. The proposed approach is expected to allow the hyperbolic method to be successfully applied to a wide variety of diffusion problems in dimensional or non-dimensional coordinates.

References

- H. Nishikawa. A first-order system approach for diffusion equation. I: Second order residual distribution schemes. J. Comput. Phys., 227:315–352, 2007.
- [2] Yi Liu and Hiroaki Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional unsteady inviscid and viscous flows. In 55th AIAA Aerospace Sciences Meeting, AIAA Paper 2017-0738, Grapevine, Texas, 2017.

- [3] H. Nishikawa and P. L. Roe. Third-order active-flux scheme for advection diffusion: Hyperbolic diffusion, boundary condition, and Newton solver. *Computers and Fluids*, 125:71–81, 2016.
- [4] A. Mazaheri and H. Nishikawa. First-order hyperbolic system method for time-dependent advectiondiffusion problems. NASA-TM-2014-218175, March 2014.
- [5] C. Cattaneo. A form of heat-conduction equations which eliminates the paradox of instantaneous propagation. Ct. R. Acad. Sci., Paris, 247:431–433, 1958.
- [6] P. Vernotte. Les paradoxes de la theorie continue del'equation de la chaleur. Comptes rendus de l'Académie des sciences, 246:3154–3155, 1958.
- [7] H. Nishikawa. First-, second-, and third-order finite-volume schemes for diffusion. J. Comput. Phys., 256:791–805, 2014.
- [8] H. Nishikawa. First, second, and third order finite-volume schemes for advection-diffusion. J. Comput. Phys., 273:287–309, 2014.
- H. Nishikawa. Alternative formulations for first-, second-, and third-order hyperbolic Navier-Stokes schemes. In Proc. of 22nd AIAA Computational Fluid Dynamics Conference, AIAA Paper 2015-2451, Dallas, TX, 2015.
- [10] R. Kawashima, K. Komurasaki, and T. Schonherr. A hyperbolic-equation system approach for magnetized electron fluids in quasi-neutral plasmas. J. Comput. Phys., 284:59–69, 2015.
- [11] Jialin Lou, Xiaodong Liu, Hong Luo, and Hiroaki Nishikawa. Reconstructed discontinuous Galerkin methods for hyperbolic diffusion equations on unstructured grids. In 55th AIAA Aerospace Sciences Meeting, AIAA Paper 2017-0310, Grapevine, Texas, 2017.
- [12] H. Nishikawa. First, second, and third order finite-volume schemes for Navier-Stokes equations. In Proc. of 7th AIAA Theoretical Fluid Mechanics Conference, AIAA Aviation and Aeronautics Forum and Exposition 2014, AIAA Paper 2014-2091, Atlanta, GA, 2014.
- [13] Y. Nakashima, N. Watanabe, and H. Nishikawa. Hyperbolic Navier-Stokes solver for three-dimensional flows. In 54th AIAA Aerospace Sciences Meeting, AIAA Paper 2016-1101, San Diego, CA, 2016.
- [14] Yi Liu and Hiroaki Nishikawa. Third-order inviscid and second-order hyperbolic Navier-Stokes solvers for three-dimensional inviscid and viscous flows. In 46th AIAA Fluid Dynamics Conference, AIAA Paper 2016-3969, Washington, D.C., 2016.
- [15] Hubert Baty and Hiroaki Nishikawa. Hyperbolic method for magnetic reconnection process in steady state magnetohydrodynamics. Mon. Not. R. Astron. Soc., 459:624–637, 2016.
- [16] R. Kawashima, K. Komurasaki, and T. Schonherr. A flux-splitting method for hyperbolic-equation system of magnetized electron fluids in quasi-neutral plasmas. J. Comput. Phys., 310:202–212, 2016.
- [17] A. Mazaheri, Mario Ricchiuto, and H. Nishikawa. A first-order hyperbolic system approach for dispersion. J. Comput. Phys., 321:593–605, September 2016.
- [18] E. F. Toro and G. I. Montecinos. Advection-diffusion-reaction equations: Hyperbolisation and high-order ADER discretizations. SIAM J. Sci. Comput., 36:A2423–A2457, 2014.
- [19] I. Peshkov and E. Romenski. A hyperbolic model for viscous newtonian flows. Continuum. Mech. Therm., 28:85–104, 2016.
- [20] W. L. Briggs, V. E. Henson, and S. F. McCormick. A Multigrid Tutorial. SIAM, second edition, 2000.
- [21] E. Lee, H. T. Ahn, and H. Luo. Cell-centered high-order hyperbolic finite volume method for diffusion equation on unstructured grids. J. Comput. Phys., 2017, in press.
- [22] H. Nishikawa. New-generation hyperbolic Navier-Stokes schemes: O(1/h) speed-up and accurate viscous/heat fluxes. In *Proc. of 20th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Reston, VA, 2011. submitted for publication.



Figure 6: Grid1 for a unit square domain in meter.

- [23] H. Nishikawa and Y. Liu. Hyperbolic advection-diffusion schemes for high-Reynolds-number boundary-layer problems. J. Comput. Phys., 352:23–51, 2018.
- [24] H. Nishikawa. Robust and accurate viscous discretization via upwind scheme I: Basic principle. Comput. Fluids, 49(1):62–86, October 2011.
- [25] Alireza Jalali, Mahkame Sharbatdar, and Carl Ollivier-Gooch. Accuracy analysis of unstructured finite volume discretization schemes for diffusive fluxes. *Comput. Fluids*, 101:220–232, September 2014.
- [26] Y. Nakashima, N. Watanabe, and H. Nishikawa. Development of an effective implicit solver for generalpurpose unstructured CFD software. In *The 28th Computational Fluid Dynamics Symposium*, C08-1, Tokyo, Japan, 2014.
- [27] H. Nishikawa, Y. Nakashima, and N. Watanabe. Effects of high-frequency damping on iterative convergence of implicit viscous solver. J. Comput. Phys., 348:66–81, 2017.

Acknowledgments

The first author gratefully acknowledges support from the U.S. Army Research Office under the contract/grant number W911NF- 16-1-0108 with Dr. Matthew Munson as the program manager, also from NASA under Contract No. NNL09AA00A with Dr. Veer N. Vatsa as the technical monitor, and support from Software CRADLE.



Figure 7: Exact temperature contours in a square domain in three different units.



Figure 8: Convergence histories for Grid3 in a square-domain problem. Solid lines are the residual norm histories. Dashed lines indicate the number of linear relaxations per iteration.



Figure 9: L_1 error convergence results for a square-domain problem. Results are shown for the temperature u[K] and the gradient variable $p(=\partial_x u)[K/m]$. Units indicated in the legend correspond to the grid unit used.



Figure 10: Grid1 for a rectangular-grid case. Not drawn to scale.



Figure 11: Iterative and error convergence results for a rectangular domain case. Solid and dashed lines are the residual convergence and linear relaxations, respectively. Error convergence results are shown for the temperature u[K] and the gradient variable $p(=\partial_x u)[K/m]$.



Figure 12: Grid1 in the rectangular-Cartesian grid case. The domain extends in the positive x-direction towards x = 1.



Figure 13: Rectangular-Cartesian grid case. Iterative convergence for Grid3 and error convergence.



Figure 14: Grid1 for a non-square domain in meter.



Figure 15: Exact temperature contours in a non-square domain in three different grid units.



Figure 16: Convergence histories for Grid3 in a non-square-domain problem. Solid lines are the residual norm histories. Dashed lines indicate the number of linear relaxations per iteration.



Figure 17: L_1 error convergence results for a square-domain problem. Results are shown for the temperature u[K] and the gradient variable $p(=\partial_x u)[K/m]$. Units indicated in the legend correspond to the grid unit used.



Figure 18: Nonlinear diffusion problem in a non-square-domain problem: (a)Convergence history for Grid3, where solid lines indicate the residual norm convergence and dashed lines indicate the number of linear relaxations per iteration. (b) L_1 error convergence results for the temperature u[K] and the gradient variable $p(=\partial_x u)[K/m]$. The grid unit is indicated in the legend.



Figure 19: Grid1 for an unsteady heat conduction problem in a non-square domain in meter.



Figure 20: Exact temperature contours at t = 0.15 for an unsteady heat conduction problem in a non-square domain in three different units.



Figure 21: Convergence histories for Grid3 for an unsteady heat conduction problem over the first 10 time steps of the total 151 time steps. Inner iterations are plotted as fractions over each physical time step. At every time step (integers on the horizontal axis), two residual norms are plotted: one is the final residual in the last inner iteration and the other is the initial residual norm for the next iteration towards the next time level.



Figure 22: Comparison of the hyperbolic solver and a conventional solver for the number of linear relaxations per iteration and CPU time in seconds required for each unsteady computation from $t = t_0$ to $t = t_f$.



Figure 23: L_1 error convergence results for an unsteady problem, measured at the final time t = 0.15. Results are shown for the temperature u[K] and the gradient variable $p(=\partial_x u)[K/m]$. The dashed line in (b) represents the slope of 1 (first-order accuracy). Units indicated in the legend correspond to the grid unit used. For Alpha4/3, the x-derivative is computed by a linear LSQ method.

Appendix A: A Derivation of Equation (20)

In Ref.[7], the following Fourier transformed first-order hyperbolic diffusion scheme was obtained on a regular quadrilateral grid of spacing h:

$$\frac{d\mathbf{U}_0}{d\tau} = \mathbf{M}\mathbf{U}_0,\tag{A.1}$$

where \mathbf{M} is given for smooth components as

$$\mathbf{M} = \begin{bmatrix} -\frac{\nu\beta^2}{2hL_r} & \frac{i\nu\beta_x}{h} & \frac{i\nu\beta_y}{h} \\ \frac{i\nu\beta_x}{hL_r^2} & -\frac{\nu}{L_r^2} & 0 \\ \frac{i\nu\beta_y}{hL_r^2} & 0 & -\frac{\nu}{L_r^2} \end{bmatrix},$$
(A.2)

where β_x and β_y are frequencies of a Fourier mode in x and y directions, respectively, and $\beta^2 = \beta_x^2 + \beta_y^2$. This operator is valid for a rectangular domain $(x, y) \in [0, L_x] \times [0, L_y]$ with a uniform spacing $h = L_x/N_x = L_y/N_y$, where N_x and N_y are the number of grid spacings in x and y directions, respectively. The eigenvalues of **M** are given by

$$-\frac{\nu}{L_r^2}, \quad -\frac{\nu}{2L_r^2} \left(1 + \frac{\beta^2}{2h} \pm \sqrt{1 - \frac{L_r \beta^2}{h} + \frac{L_r^2 \beta^2}{4h^2} (\beta^2 - 16)} \right).$$
(A.3)

An optimal length L_r was derived in Ref.[7] by requiring the expression inside the square root to be non-positive, so that the Fourier mode propagates rather than is purely damped:

$$1 - \frac{L_r \beta^2}{h} + \frac{L_r^2 \beta^2}{4h^2} (\beta^2 - 16) \le 0, \tag{A.4}$$

which we write

$$f(L_r) \le 0, \quad f(L_r) = 1 - \frac{L_r \beta^2}{h} + \frac{L_r^2 \beta^2}{4h^2} (\beta^2 - 16) = \frac{1}{4h^2} \left[(\beta - 4)\beta L_r - 2h \right] \left[(\beta + 4)\beta L_r - 2h \right].$$
(A.5)

For smooth components, e.g., $(\beta_x, \beta_y) \in (-\pi/2, \pi/2) \times (-\pi/2, \pi/2)$, we have $\beta < 4$, and thus $f(L_r) \leq 0$ if

$$L_r \le \frac{2h}{(\beta - 4)\beta} \le 0, \quad L_r \ge \frac{2h}{(\beta + 4)\beta} \ge 0, \tag{A.6}$$

but since $L_r > 0$, we are left with

$$L_r \ge \frac{2h}{(\beta+4)\beta}.\tag{A.7}$$