

# AN INTRODUCTION TO THE FINITE VOLUME METHOD USING THE ELASTIC WAVE EQUATION \*

ANDREW NOLAN<sup>†</sup>

**Abstract.** An introductory description of finite volume methods for solving partial differential equations is discussed below. Conservation laws, the finite volume method, and the Riemann problem are all investigated via the one-dimensional elastic wave equation. We use the method of characteristic to reduce this couple set of equations too the advection equation which we the solve numerically via finite volume methods.

**Key words.** Finite Volume, PyClaw, Elastic Wave Equation

**1. Introduction.** Numerical methods for solving partial differential equations (PDEs) arise across all scientific fields. From modeling future predictions of global sea-level rise to heat flow through a rod, numerical solution to PDEs at the heart of modeling physical phenomena. The finite difference method, which approximates derivatives using the Taylor expansion, is a common starting point for numerical solutions to PDEs. By using the Taylor expansion ordinary differential equations or PDEs can be reduced to a series of algebraic statements for which matrix methods can be used to solve efficiently.

While the finite difference methods are a powerful starting point they have their limitations; one of which is dealing with discontinues. Due to the failure of the finite difference method and other numerical methods to deal with discontinuities, the finite volume method arises. The finite volume method is commonly used to solve hyperbolic PDEs, due to their susceptibility to developing shock waves, but can be used to solve all type of PDE. At the heart of the finite volume method is the *Riemann Problem*, an initial value problem with piecewise constant initial conditions that naturally arises due to the discretized domain on which the equations are solved.

We begin with an introduction to conservation laws and the finite volume method as motivated by the one-dimensional advection equation. We derive conservation laws based on physical intuition of fluid flow as guided by [3]. We then use the conservative form of the advection equation to derive the finite volume method. First and second-order numerical flux approximation methods are stated. We also investigate higher-order slope-limiters through the PyClaw package. We then derive an analytical solution to the homogeneous (*i.e.* constant-coefficient) elastic wave equation through the method of characteristics that we use to validate our numerical methods. Numerical experiments are conducted to investigate the performance of the numerical methods outlined. The code for the numerical experiments animations of the model runs can be found [here](#). We conclude this report with a discussion significance of the work below and some discussion of the avenues for future work.

**2. Conservation Laws.** We being by deriving conservation laws based on based on physical intuition of fluid flow. We consider a fluid flow such as dye tracer in a river or a supraglacial stream with a positive advection speed ( $\bar{a}$ ), that is the fluid is flowing from left to right. Our derivation closely follows that of [1] who in turn closely follow [3]. The total mass of the quantity in question (*i.e.* concentration of

---

\*Submitted on 11/29/2019

<sup>†</sup>[anolan@sfu.ca](mailto:anolan@sfu.ca)

the tracer) in a given unit volume (in the one-dimensional case just length) is

$$(2.1) \quad \int_{x_l}^{x_r} q(x, t) dx$$

where  $x_l$  and  $x_r$  are the left and right cell boundaries of unit volume respectively [1]. Under the assumption there is no source term (*e.g.* rainfall), the only changes in time will be through flux across the left or right cell boundaries. Therefore,

$$(2.2) \quad \frac{\partial}{\partial t} \int_{x_l}^{x_r} q(x, t) dx = F_l(t) - F_r(t)$$

where  $F_i(t)$  are mass fluxes rates across the cell boundaries [1]. Equation 2.2 represents the integral form of our conservation law [3]. The power of the finite volume method lies in the fact that Equation 2.2 still holds at discontinuities where PDEs is no longer valid [1]. Therefore, the finite volume method is able to accurately approximate solutions at discontinuities (given a high enough order approximation) while other numerical methods like the finite difference method will fail to produce accurate results. If we rewrite the fluxes  $F_i$  as functions of  $q(x, t)$  such that

$$(2.3) \quad F \rightarrow f(q(x, t))$$

we can write our the change in unit volume with time as

$$(2.4) \quad \frac{\partial}{\partial t} \int_{x_l}^{x_r} q(x, t) dx = f(q(x_l, t)) - f(q(x_r, t)).$$

Under the assumption that  $f$  and  $q$  are sufficiently smooth, the equation above can be rewritten as

$$(2.5) \quad \frac{\partial}{\partial t} \int_{x_l}^{x_r} q(x, t) dx = - \int_{x_l}^{x_r} \frac{\partial}{\partial x} f(q(x, t)) dx$$

using the definition of a definite integral [3]. Further simplification leads to

$$(2.6) \quad \int_{x_l}^{x_r} \left[ \frac{\partial}{\partial t} q(x, t) + \frac{\partial}{\partial x} f(q(x, t)) \right] dx = 0.$$

Since the definite integral of Equation 2.6 evaluated from  $x_l$  to  $x_r$  is equal to zero, the quantity being integrated must be identically zero [3]. This results in

$$(2.7) \quad \frac{\partial}{\partial t} q(x, t) - \frac{\partial}{\partial x} f(q(x, t)) = 0$$

known as the differential form of our conservation law [3]. The conservation law above can be written in form the advection equation as

$$(2.8) \quad \frac{\partial}{\partial t} q(x, t) - \bar{a} \frac{\partial}{\partial x} q(x, t) = 0$$

for a homogeneous material or as

$$(2.9) \quad \frac{\partial}{\partial t} q(x, t) - \frac{\partial}{\partial x} A(x, t) q(x, t) = 0$$

for a heterogeneous material. In the case of the homogeneous material the advection velocity is constant, while the advection velocity varies in space and time for the heterogeneous material. Both scenarios will be investigated below.

**3. Finite Volume.** We now derive the finite volume method for one-dimensional conservation laws on a numerical grid where the  $i$ -th grid cell,  $\mathcal{C}_i$  has cell interfaces  $x_{i-1/2}, x_{i+1/2}$ . We can approximate the value of the solution field  $q(x, t)$  by the average quantity  $Q_i^n$  within a given grid cell as

$$(3.1) \quad Q_i^n = \frac{1}{\Delta x} \int_{\mathcal{C}_i} q(x, t) dx \approx \frac{1}{\Delta x} \int_{x_l}^{x_r} q(x, t) dx$$

where the subscript denotes the grid cell, the superscript denotes the time integration step and  $\Delta x$  is the size of the grid cell [3, 1]. We use the above definition of average quantity along with the integral form of our conservation law (Equation (2.2)) to derive a time integration algorithm. By integrating Equation (2.4) from  $t_n$  to  $t_{n+1}$  we then get

$$(3.2) \quad \int_{x_l}^{x_r} q(x, t_{n+1}) dx - \int_{x_l}^{x_r} q(x, t_n) dx = \int_{t_n}^{t_{n+1}} f(q(x_l, t)) dt - \int_{t_n}^{t_{n+1}} f(q(x_r, t)) dt$$

In order for our integrated equation to match the cell averaged form derived above we divide all terms by  $\Delta x$  and then solve for  $q(x, t_{n+1})$  ending up with

$$(3.3) \quad \frac{1}{\Delta x} \int_{x_l}^{x_r} q(x, t_{n+1}) dx = \frac{1}{\Delta x} \int_{x_l}^{x_r} q(x, t_n) dx - \frac{1}{\Delta x} \left[ \int_{t_n}^{t_{n+1}} f(q(x_r, t)) dt - \int_{t_n}^{t_{n+1}} f(q(x_l, t)) dt \right].$$

Under the assumption we do not have exact form of  $q(x_i, t)$  we generally cannot exactly compute the time integration on the right hand side of the equation above [3]. The equation above does allude to the development of numerical solution of the form

$$(3.4) \quad Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} [F_{i+1/2}^n - F_{i-1/2}^n]$$

where  $F_{i\pm 1/2}$  is an approximation of the average flux [3] of the form

$$(3.5) \quad F_{i\pm 1/2} \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(q(x_{i\pm 1/2}, t)) dt.$$

Under the assumption that  $F_{i\pm 1/2}$  can be approximated by just using the flux at adjacent cells we can state the numerical flux as

$$(3.6) \quad F_{i+1/2}^n = \mathcal{F}(Q_{i+1}, Q_i)$$

The simplification of just using neighboring cell to approximate the average flux is appropriate for our purposes given that waves propagate at finite speeds in hyperbolic systems [3]. We explicitly demonstrate the elastic wave equation to be hyperbolic in the following section. This formulation of the numerical flux produces a numerical method of the form

$$(3.7) \quad Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} [\mathcal{F}(Q_{i+1}, Q_i) - \mathcal{F}(Q_i, Q_{i-1})].$$

How we choose to calculate the numerical flux  $\mathcal{F}$  determines the order of accuracy of finite volume method [3].

For example we have the centered, second-order Lax-Wendroff method

$$(3.8) \quad Q_i^{n+1} = Q_i^n - \frac{a\Delta t}{2\Delta x} [\Delta Q_{i+1} - \Delta Q_{i-1}] + \frac{1}{2} \left( \frac{a\Delta t}{\Delta x} \right)^2 (Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n)$$

which is equivalent to the Lax-Wendroff method for the finite difference method [1]. None the less the Lax-Wendroff is also a finite volume scheme, in the case above using downwind slopes [1], were the numerical fluxes are

$$(3.9) \quad \begin{aligned} F_{i-1/2}^n &= \frac{1}{2}a(Q_{i-1}^n + Q_i^n) - \frac{1}{2}\frac{\Delta t}{\Delta x}a^2(Q_i^n + Q_{i-1}^n) \\ F_{i+1/2}^n &= \frac{1}{2}a(Q_{i+1}^n + Q_i^n) - \frac{1}{2}\frac{\Delta t}{\Delta x}a^2(Q_{i+1}^n + Q_i^n). \end{aligned}$$

The statement of upwind method will follow below after we have derived analytical solution to the homogeneous equation, which makes the Riemann problem more evident.

**4. Homogeneous Elastic Wave Equation.** In order to test our derived numerical methods we investigate the one-dimensional elastic wave equation propagating through a homogeneous medium. The source free form of the elastic wave equation can be written as a coupled system of equations

$$(4.1) \quad \begin{aligned} \frac{\partial}{\partial t}\sigma - \mu \frac{\partial}{\partial x}v &= 0 \\ \frac{\partial}{\partial t}v - \frac{1}{\rho} \frac{\partial}{\partial x}\sigma &= 0, \end{aligned}$$

where  $\sigma = \sigma_{xy}$  is the shear stress component of the stress tensor,  $v$  is the transverse velocity,  $\rho$  is the density, and  $\mu$  is the bulk modulus [1]. Written in matrix-vector notation our elastic wave equation becomes

$$(4.2) \quad \partial_t Q + A \partial_x Q = 0$$

where

$$Q = \begin{bmatrix} \sigma \\ v \end{bmatrix}, \quad A = \begin{bmatrix} 0 & -\mu \\ -1/\rho & 0 \end{bmatrix}.$$

Equation 4.2 resembles the one-dimensional advection equation (Equation 2.8) except we have a coupled system of  $m = 2$  equations. In order to make use of the numerical methods outlined above we need to decouple the system of equations. In order to do so we find the eigendecomposition of  $A$  such that

$$(4.3) \quad A = X \Lambda X^{-1}$$

where  $X \in \mathbb{R}^{m \times m}$  is the matrix of eigenvectors of  $A$  and  $\Lambda \in \mathbb{R}^{m \times m}$  is the diagonal matrix with the eigenvalues of  $A$  along the diagonal. Given that the eigendecomposition exists this linear system is hyperbolic [3]. By replacing  $A$  by it's eigendecomposition and multiplying both sides of the equation by  $X^{-1}$  Equation 4.2 can be simplified to

$$(4.4) \quad \partial_t W + \Lambda \partial_x W = 0$$

where  $W = X^{-1}Q$  is the solution vector, comprised of the characteristic variables  $w_{1,2}$  [1, 3]. Since  $\Lambda$  is diagonal we now left with  $m = 2$  decoupled advection equations [3] for the characteristic variables in  $W$  such that

$$(4.5) \quad \partial_t w_p + \lambda_p \partial_x w_p \text{ for } p = 1, 2.$$

We have  $m$  waves traveling at characteristic speeds ( $\lambda_p$ ) [3].

We find the eigenvalues of  $A$  to be  $\lambda_{1,2} = \pm\sqrt{\mu/\rho} = \pm C$ , which by definition is the shear velocity [1]. With these eigenvalues we determine our eigenvectors to be

$$(4.6) \quad x_1 = \begin{bmatrix} Z \\ 0 \end{bmatrix}, \quad x_2 = \begin{bmatrix} -Z \\ 0 \end{bmatrix}$$

where  $Z = \rho c$  which by definition is the seismic impedance [1]. We now have a matrix of eigenvectors such that

$$(4.7) \quad X = \begin{bmatrix} Z & -Z \\ 1 & 1 \end{bmatrix}, \quad X^{-1} = \frac{1}{2Z} \begin{bmatrix} 1 & Z \\ -1 & Z \end{bmatrix}.$$

Now that we have our eigendecomposition of  $A$  we can find an analytical solution to Equations 4.1 that can be used to validate our numerical methods against. The simplest generic solution to Equations 4.1 is of the form  $w_{1,2} = w_{1,2}^0(x, t)$  where  $w_{1,2}^0$  are waveforms being advected [3, 1]. From Equation 4.4 we can solve for the characteristic variables

$$(4.8) \quad \begin{aligned} W = X^{-1}Q &= \frac{1}{2Z} \begin{bmatrix} 1 & Z \\ -1 & Z \end{bmatrix} \begin{bmatrix} \sigma^0(x \pm Ct) \\ v^0(x \pm Ct) \end{bmatrix} \\ &= \frac{1}{2Z} \begin{bmatrix} \sigma^0(x + Ct) + Zv^0(x + Ct) \\ -\sigma^0(x - Ct) + Zv^0(x - Ct) \end{bmatrix} \end{aligned}$$

based on our initial waveforms being advected. We then relate the characteristic variables to stress ( $\sigma$ ) and velocity ( $v$ ) by

$$(4.9) \quad Q = XW = \frac{1}{2Z} \begin{bmatrix} Z & -Z \\ 1 & 1 \end{bmatrix} \begin{bmatrix} \sigma^0(x + Ct) + Zv^0(x + Ct) \\ -\sigma^0(x - Ct) + Zv^0(x - Ct) \end{bmatrix}.$$

The product of this expression  $Q = XW$  produces the analytical solution

$$(4.10) \quad \begin{aligned} \sigma(x, t) &= \frac{1}{2}(\sigma^0(x + Ct) + \sigma^0(x - Ct)) + \frac{Z}{2}(v^0(x + Ct) - v^0(x - Ct)) \\ v(x, t) &= \frac{1}{2Z}(\sigma^0(x + Ct) - \sigma^0(x - Ct)) + \frac{1}{2}(v^0(x + Ct) + v^0(x - Ct)). \end{aligned}$$

We now have an analytical solution to validate our numerical results against. Our analytical solution (Equation 4.10) can be written as

$$(4.11) \quad Q(x, t) = \sum_{p=1}^m w_p(x, t)x_p$$

where  $x_p$  is an eigenvector and  $w_p(x, t)$  is the coefficient of the eigen vector [1, 3]. Therefore, we can think of our analytical solution as a sum of eigenvectors  $x_p$  of strength  $w_p(x, t)$  being advected at a velocity  $\lambda_p$  [3]. From our analytical solution we see that we have an initial wave  $w_p(x, 0)$  advected at speed  $\lambda_p$  along the characteristic curve  $X(t) = x_0 + \lambda_p t$  [3]. Given that we have a constant coefficient in Equation 4.2, our characteristic curves are straight lines. The assumptions used to derive Equation 4.10 only hold for a constant coefficient (homogeneous material) system of equations.

**5. Riemann Problem and the Upwind Method.** The Riemann problem consists of an initial value problem with piecewise constant initial data with a single discontinuity. If we set the location of discontinuity to be  $x = 0$  then we have

$$(5.1) \quad Q^n(x) = \begin{cases} Q_l & \text{if } x < 0, \\ Q_r & \text{if } x > 0 \end{cases}$$

which propagates along characteristic curves [3]. It is fairly clear how a situation such as the Riemann problem would arise given that we are using cell averaged fluxes to approximate the fluxes in space and time. For the elastic wave equation above, given the initial conditions  $Q_l$  and  $Q_r$  can we calculate the jump

$$(5.2) \quad \Delta Q = Q_l - Q_r = \alpha_1 x_1 + \alpha_2 x_2$$

where  $x_p$  are the eigenvectors of  $A$  and  $\alpha_p$  are weight as of the summation of eigenvectors as in Equation 4.11. We can solve for  $\alpha$  by

$$(5.3) \quad \alpha = X^{-1} \Delta Q$$

where  $X^{-1}$  are the left eigenvectors and  $\alpha$  is the vector containing the individual  $\alpha_p$  values [3]. For the case of the elastic wave equation, we can decompose the solution into left (negative) and right (positive) propagating eigenvalues:

$$(5.4) \quad \Lambda^- = \begin{bmatrix} -C & 0 \\ 0 & 0 \end{bmatrix}, \quad \Lambda^+ = \begin{bmatrix} 0 & 0 \\ 0 & c \end{bmatrix}.$$

Which in turn can be used to compute

$$(5.5) \quad \begin{aligned} A^+ &= X \Lambda^+ X^{-1} \\ A^- &= X \Lambda^- X^{-1} \end{aligned}$$

for the constant, scalar coefficient scenario detailed above [1]. With our derived left and right advection velocities and Equation 3.7 we can approximate our numerical fluxes

$$(5.6) \quad \begin{aligned} \mathcal{F}_l &= A^- \Delta Q_l = A^-(Q_i - Q_{i-1}) \\ \mathcal{F}_r &= A^+ \Delta Q_r = A^+(Q_{i+1} - Q_i) \end{aligned}$$

given the scalar constant [1]. Therefore we can plug our numerical fluxes into Equation 3.7 to produce

$$(5.7) \quad Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (A^- \Delta Q_l + A^+ \Delta Q_r)$$

which is the first-order upwind method [1, 3]. We know have two numerical methods, the first-order Upwind method and the second-order Lax-Wedfford method. We will now test these methods against our analytical solution for the one-dimensional elastic wave equation. We will additionally use the **ClawPack** software package for higher-order numerical flux approximations [2].

**6. Numerical Experiments.** Numerical experiments using the finite volume method were conducted with the one-dimensional elastic wave equation. Input parameters were chosen to resemble realistic Earth system values (Table 1) [1]. An initial condition of

$$(6.1) \quad Q(x, 0) = e^{-\beta(x-x_0)^2} \cos(\gamma(x-x_0))$$

centered at 4000 m is used given its resemblance to a Gaussian function. Parameter values for  $\beta$  and  $\gamma$  are listed in Table 1. Periodic boundary conditions were imposed such that

$$(6.2) \quad Q_0^n = Q_{-2}^n, \quad Q_{-1}^n = Q_1^n$$

having adopted a pythonic indexing method, where  $-2$  represents the second to last element in the solution vector. Periodic boundary conditions were chosen to allow the model to run for a relatively long time without having to discretize an absurdly large model domain. Therefore, we can relatively quickly and computationally efficiently observe the errors each of the respective method produces. All time integration's

Variable	Description	Value	units
$\rho$	Density	2500	kg/m <sup>3</sup>
$C$	Transverse velocity	2500	m/s
$L$	Length of spatial domain	10000	m
$nx$	Number of girdcells	800	
$s$	Courant number	0.5	
Initial Condition Parameters			
$\gamma$	parameter in $e^{(x)}$	$5 \times 10^{-6}$	
$\beta$	parameter in $\cos(x)$	$2 \times 10^{-5}$	
$x_0$	Initial position	4000	m

TABLE 1

*Material properties and model parameters used for in diagnostic numerical experiments.*

observed the CFL condition to ensure numerical stability (given that our numerical methods themselves are stable, [3]). Since our problem is hyperbolic the CFL condition can be defined as

$$(6.3) \quad s = \frac{\Delta t}{\Delta x} \max_p |\lambda_p| < 1$$

where  $\lambda_p$  is our maximum eigenvalue, since our eigenvalue are the speed at which our waves are advected [3]. We selected a Courant number ( $s$ ) of 0.5 and with our chosen spatial domain parameters we solve for our time integration step  $\Delta t$ .

Results after one second of time integration are shown in Figure 1. The PyClaw package was used to investigate higher-order slope limiters [2]. In the case of the simulations in Figure 1 a Supberbee slope limiter was used. After one second of time integration we can already observe a significant amount of error from the Upwind method. The error in the first-order method appears to be concentrated at steep gradients in the solution vectors, which intuitively makes sense since the first-order methods do not contain any information about higher-order derivatives which is crucial to approximations at steep gradients. The errors in the first-order method resemble artificial diffusion, where the amplitude of the wave being advected is decreasing artificially.

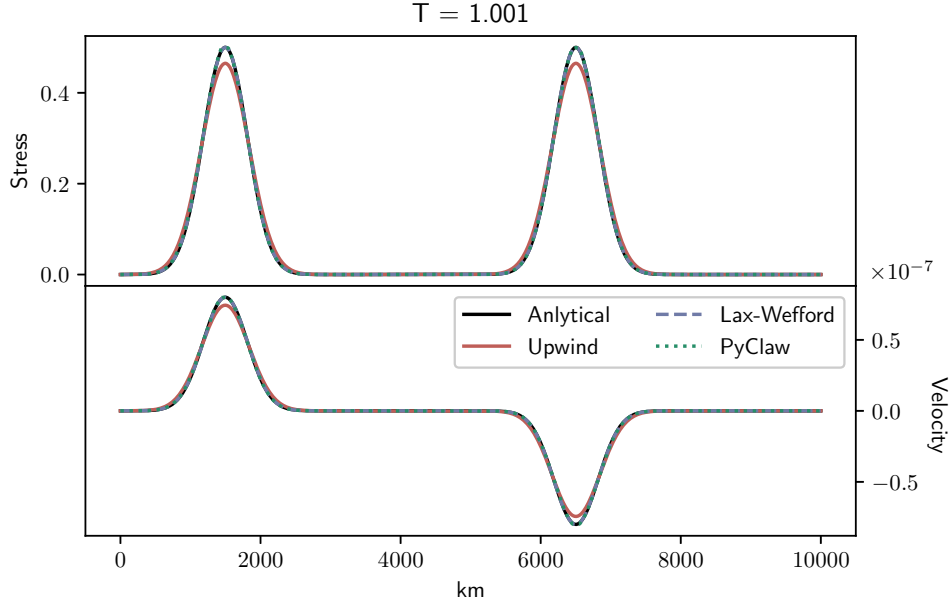


FIG. 1. *one-dimensional homogeneous elastic wave equation solutions for our three numerical methods plotted against that analytical solution. The PyClaw solution used a Superbee slope limiter. Results are plotted after  $\approx 1$  second of time integration.*

While one second of time integration made the error in the Upwind method apparent, there seemed to be little variability in the Lax-Wefford and Superbee (PyClaw) methods as compared to the analytical solution. To investigate the error in these higher-order methods more, we run the model for 20 seconds of time integration. The code used to run these numerical experiments, along with the figures produced are hosted online as a [Github repository](#). The repository includes HTML animations of the model run within a Jupyter notebook. Solution vectors at 5, 10, 15, and 20 seconds from the 20 second time integration are plotted in Figure 2. The artificial diffusion from Figure 1 is even more apparent by five seconds, while there is still strong agreement between the Lax-Wefford and Superbee methods. Note the analytical solution is not plotted in Figure 2, given that I was unable to implement the periodic boundary conditions for the analytical solution. Therefore, we are using the PyClaw solution as our reference which we compare the Upwind and Lax-Wefford solutions against. Note the velocity solution after 10 seconds of time integration in Figure 2. The amplitude of Lax-Wefford scheme is much larger than the than either the Upwind or Superbee solution. There appears to large error when the two waves traveling in opposite directions meet as observed in the velocity solutions at  $t = 10$  and  $t = 20$ . While the error appears to be large when two waves meet, there does not appear to be much disagreement between the Lax-Wefford and PyClaw solutions after the waves are advected and no longer interfering, as observed in the velocity solution at  $t = 15$  (Figure 2). It appears as though the Law-Wefford method does a sufficient job when compared to the Superbee method in the case of the initial condition used here.



**7. Discussion and Conclusion.** Numerical experiments using the elastic wave equation demonstrate large errors in the Upwind method, while there is a modest agreement between the Law-Wefford and Superbee methods, as implemented through PyClaw. While qualitatively there appeared to be agreement between the two methods, a method to quantify the error for each numerical method would have been useful. A method such as the Richardson extrapolation would be a well-suited candidate for such quantification, but was outside the scope of this work. Additionally, the qualitative agreement between the two higher-order methods may be in part due to the initial condition, and therefore waveform being advected, used in this study. The initial condition as represented by Equation 6.2 is relative smooth and while there may be steep gradients at points, there are no discontinuities. Therefore, to more robustly test the capabilities of the methods outlined above, one would want to use an end member waveform such as one with discontinuity. Another shortcoming of this study was the efficiency of the code used to run these experiments. Given the nature of the coupled equation and matrix multiplication, the equation was not simply vectorized and therefore solved via nested for loops. We also used explicit methods, as is common practice to hyperbolic systems [3], and therefore could not make use of iterative solvers such as GMRES or the conjugate gradient method. An immediate step to take in the future will be implementing the PyClaw package in parallel as is supported through PETSc.

While the code may not have as efficient as possible and no robust error quantification was done, the work none the less was fruitful. Based on physical intuition and guided by [3] we derived conservation laws from which we can formulate the advection equation (Equation 2.8). We then derived the framework of the finite volume method based on numerical approximations of fluxes into and out of our control volume. The sound physical framework on which conservation laws and finite volume methods are derived from has left us with an appreciation for the numerical method. Next, we derived an analytical solution to the one-dimensional elastic wave equation through the method of characteristics. This derivation of an analytical solution demonstrated the significance of the eigendecomposition to physical problems. Having seen material properties such as the shear velocity  $C$  and the seismic impedance  $Z$  stated in previous courses it was illuminating to see how these material properties are derived. Our derivation of the analytical solution and the decomposition of  $A$  also informed our discussion of the Riemann problem. Finally, we conducted the numerical experiment using the first and second-order finite volume methods stated above, and also successfully implemented higher-order solutions through the PyClaw package. The work done here represents a formal introduction to the finite volume method and the PyClaw package from which these methods could be used for genuine scientific inquiry.

**Supplementary material.** The code used to run these numerical experiments along with animations of the model runs can be found [here](#).

## REFERENCES

- [1] H. IGEL, *Computational Seismology: A Practical Introduction*, Oxford University Press, 10 2016.
- [2] D. I. KETCHESON, K. MANDLI, A. J. AHMADIA, A. ALGHAMDI, M. Q. DE LUNA, M. PARSANI, M. G. KNEPLEY, AND M. EMMETT, *PyClaw: Accessible, extensible, scalable tools for wave propagation problems*, SIAM Journal on Scientific Computing, 34 (2012), pp. C210–C231.
- [3] R. J. LEVEQUE, *Finite Volume Methods for Hyperbolic Problems*, Cambridge Texts in Applied Mathematics, Cambridge University Press, 2002.

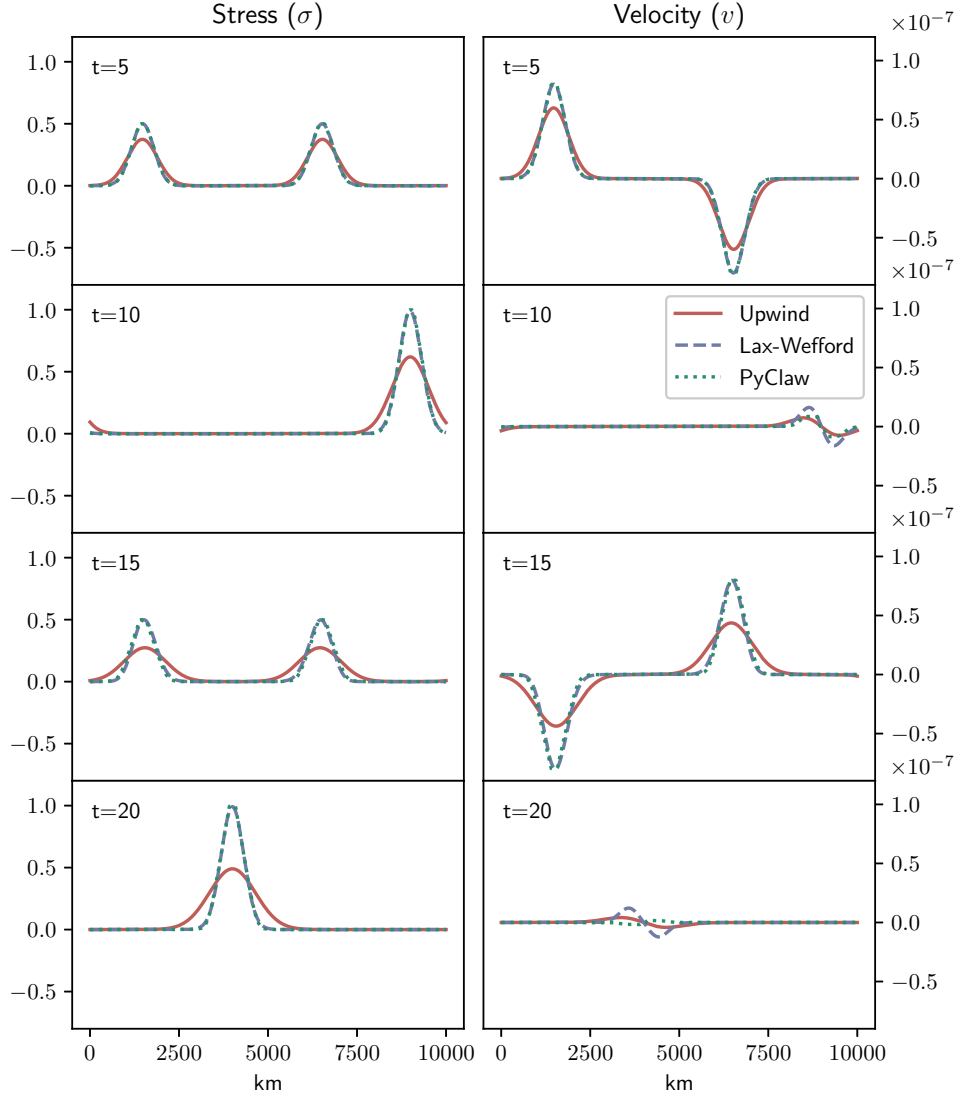


FIG. 2. Stress and velocity solutions, left and right columns respectively, for the three numerical methods plotted after 5, 10, 15, and 20 seconds of time integration. Note the analytical solution is not plotted. I was not able to implement the periodic boundary condition on the analytical solution and therefore after one full period, the results no longer appropriate to compare our numerical results against.