1. Every bullet is an unroll
2. Blue colour indicates where to hover the mouse, what text to select, etc.

Title

Thank you for attending my TMC meeting, I'd be speaking about Lax-Wendroff schemes in FR framework. I'd make my proposals on how to limit them and study their relation with ADER.

Outline

- Review of previous year's work
- I'll show the kind of limiting we are doing for Lax-Wendroff schemes
- Use blending limiter to extend Zhang-Shu's limiter to Lax-Wendroff, leading to a provably admissibility preserving Lax-Wendroff scheme
- I'd showing how to blend with MUSCL-Hancock while still keeping admissibility preservation.
- Study relation between ADER and Lax-Wendroff
- 1-D and 2-D numerical results showing accuracy and robustness
- A bit about how we wrote an efficient code
- Ending with a summary and reading activities

Flux Reconstruction : grid

- I explain Flux Reconstruction for hyperbolic conservation laws,
- We break the domain into cells I_e, in each of which we put $N+1$ degrees of freedom so that the approximate solution u_h is locally a polynomial of degree N. The faces of the e^{th} cell are $x_{e\pm\frac{1}{2}}$, where the solution is allowed to be discontinuous. The reference cell is $[0,1]$ with reference variable ξ.

Flux reconstruction : ODE

The basic idea of FR is to construct a degree $N+1$ continuous flux approximation taking numerical flux values at the interfaces and then using it to convert the conservation law to an ODE. The reason to keep it at degree $N+1$ is to get the same degree for LHS and RHS here.

Flux reconstruction : Continuous flux

To construct the continuous flux, we require the discontinuous flux, numerical flux and correction functions. Discontinuous flux function is the degree N polynomial obtained which agrees with the flux function at the solution points. Corrector functions g_L, g_R are degree $N+1$ polynomials that take the value 1 at one interface and 0 at the other; so we can use them to set the corresponding numerical flux value at the left select left correction in f_h definition and right interface select left correction in f_h definition respectively. This gives us the continuous flux polynomial and we can use it to performed evolution as shown in… Hover over to 4 to go to the previous slide, don't press Page up the previous slide.

Lax-Wendroff Flux Reconstruction (LWFR) with D2 dissipation

- Performing a Taylor expansion in time and using the conservation law to replace one derivative with spatial, we can write evolution to the next time as follows. To approximate these time derivatives of the flux, we will have to use the conservation law again. There's various ways of doing it, we choose

- the Approximate Lax-Wendroff procedure where we use finite difference in space and approximate the time nodal values by Taylor's theorem to reduce to spatial derivatives. We chose this approach to avoid computing large Jacobian matrices.
- This procedure will give us discontinuous flux polynomial, which we will correct with Flux Reconstruction.
- In the past works, the dissipative part of numerical flux has been computed using the solution polynomial.
- However, in our last work, we proposed computing the dissipation with the time averaged polynomial which is already available through the Lax-Wendroff procedure. This gives makes Rusanov's flux upwind for constant linear advection and leads to higher CFL numbers.

Extension to 2-D

- For 2-D conservation laws, our reference map is a tensor product of 1-D reference maps
- Conversion of temporal derivatives to spatial now has to be performed with both the x, y fluxes,
- using which we get the discontinuous approximation to x, y fluxes Select F_h^δ, G_h^δ.
- At each of the solution points on the face, we construct the continuous x, y fluxes,
- and then we can perform the FR corrected update as in 1-D.

Blending limiter (Hennemann Et Al.)

- Let's see a brief overview of subcell based blending limiters. The high order update can be written like hover over $\boldsymbol{u}_e^{H,n+1} = \boldsymbol{u}_e^n - \frac{\Delta t}{\Delta x_e} \boldsymbol{R}_e^H$.
- We will create a subcell based lower order scheme which can also be written in the same way ,
- and then we can naturally blend the two residuals and perform a limited update.
- There's two things that we need to specify. First, how we compute the lower order residual and how we choose the blending factor α_e. Let's first decide how we choose the α_e.

Choice of α_e

- We follow the procedure of Hennmann Et Al. We choose the blending coefficient by a smoothness indicator based on Legendre expansion select $\epsilon = \sum_{j=1}^{N+1} m_j L_j, \qquad m_j = \langle \epsilon, L_j \rangle_{L^2},$ The measure of smoothness is dictated by the content of highest order coefficient in comparison to the rest.
- We have put parameters β_1, β_2 as coefficients of the constant nodes. We have found them to be problem dependent, while they have always been kept to be equal to 1 in the literature. I'd justify that in the next slide.
- After this, we decide the quantity whose smoothness we are going to be measure. For Euler's equations, $\epsilon = \rho p$ was used by Henneman Et Al, and we found the same to perform well.
- Finally, we map these values to the interval $[0, 1]$ to be used as blending coefficients.

Choosing β_1, β_2

- Here we test a composite signal on constant linear advection equation. We should the signifiance of these coefficients of constant nodes by comparing the performance of blending limiter. We keep $\beta_2 = 1$ in both and compare $\beta_1 = 0.0$ and $\beta_1 = 1.0$. Since all regions other than the jump are well resolved, we zoom in.

- We can clearly see that $\beta_1 = 0$ is able to suppress all the oscillations while $\beta_1 = 1$ doesn't. Another thing to note is that this indicator is clearly not frame invariant. For instance, if we just translate the solution point values from $[0, 1]$ to $[100, 101]$,
- the behaviour of limiter changes. I am not arguing that indicator should be frame invariant, just that the optimal indicator would be problem dependent.

Lower order update

- We begin to understand the lower order update by understand what the precise subcells are.
- We choose them so that size of j^{th} subcell would be of the j^{th} quadrature weight. Thus, the subcells are not uniform and that the solution values, staying the same, are not corresponding to cell centres of the subcells.
- In the inner subfaces, we pick the numerical flux according to the lower order scheme, which would be MUSCL-Hancock in our case. At the faces which are shared by the FR cells, we use a blended numerical flux. Even the numerical flux used by the higher order scheme will be replaced by the blended numerical flux. The benefit of the choice of subcells would be that both the cell averages of both high order and low order schemes would evolve -
- - like this. That is, conservation is maintained. But, we have to be carefully choose this blended flux. Otherwise, our lower order residual may not be admissibility preserving. More specifically, we first make corrections to preserve admissibility at the first and last solution points.

Interface numerical flux

- We pick an initial candidate where the blending coefficient is chosen to be the average of neighbours. This choice already gives a balance between accuracy and robustness.
- Recall that we are interested in making corrections for the last solution point. With this candidate, the candidate update would look like this. It is *not* guaranteed to be admissible.
- We assume that admissibility can be written as positivity of a concave function.
- Now, a purely low order flux update would preserve admissibility. So, we do have an admissible reference where blended flux has been replaced with the purely low order one. Ideally, we'd like to find the optimal, admissible convex combination between the two. But, that would be an expensive, problem dependent procedure. So, we instead do find the θ that is used for Zhang-Shu's limiter in practice.
- With this, we can replace the candidate flux with a flux that preserves admissibility of the last solution point like this
- The complete procedure would require the same procedure on the first point on the $e + 1^{\text{th}}$ cell. Also, if there are multiple variables, we correct them successively. With this, we have chosen the numerical flux that will be shared by both the residuals.

Extension of Zhang-Shu's limiter to Lax-Wendroff schemes

- Our lower order update explicitly looks as follows,
- and the way we corrected the time averages flux now ensures admissibility,
- so that the conservation property actually implies that the cell averages of blended scheme now preserve admissibility.
- We can now apply Zhang-Shu's limiter to obtain an admissibility preserving Lax-Wendroff scheme.

- That is, their idea too was of doing something to keep cell averages admissible to use Zhang-Shu's limiter. However, their idea was to directly correct cell averages which would require additional cell loop and storage. The advantage of our proposed approach is that we don't require any additional cell/face loops and there are very little storage requirements.
- Other approach is to aposteriorily modify α_e, which will also require additional storage.

Low order residual : MUSCL-Hancock

- Now we explain the final ingredient in understanding the scheme which is how we do the MUSCL-Hancock reconstruction. The derivation starts by integrating the conservation law in space-time.
- We approximate the time integral with mid-point rule,
- leaving us with these time level $n + 1/2$ quantities to compute at the faces,
- which we compute by a finite difference method applied to the conservation law,
- where we require reconstruction of approximate solution to the face. Here we have to account for non-centred nature of the grid. Also, we use the β-minmod limiter for the grid, where the β parameter is chosen according to the smoothness indicator.

  Now, we want this procedure to be admissibility preserving. It is not guaranteed by default so we have to limit this slope further to ensure it.

  Our arguments will first ensure that evolution to $n+1/2$ is admissible and then the final FVM type update.

Admissibility of low order method

We will understand it in isolation of blending. Basically, we'd prove admissibility of MUSCL-Hancock procedure for finite volume grids which are not cell-centred; the solution point could be anywhere.

This theorem gives the final requirement for admissibility of MUSCL-Hancock, for conservative reconstruction select $u_i^{n,\pm}$ definitions. The fundamental quantity is this $\boldsymbol{u}_i^{*,\pm}$, it is all about limiting the slope so that this guy becomes admissible, along with some time step restrictions, which we'll see in a while. Now we briefly see the proof.

Generalizing Berthon's proof

- Berthon had proven the previous theorem for cell-centred grids, where $\boldsymbol{u}_i^{*,\pm}$ is defined as follows,
- while our generalization is chosen with these factors μ^-, μ^+ so that
- in the conservative case, $\boldsymbol{u}_i^{*,\pm}$ still has the natural expression. That is, in Berthon's case, these were $\pm 2\Delta x_i$, making this a natural generalization.

Idea of proof

- We split the domain into sizes μ^-, μ^+ creating Riemann problems at each of their centers. The left and right most state of the Cauchy problem are the reconstructions and center state is the $\boldsymbol{u}_i^{*,+}$. By choosing time step restriction so that the Riemann problems are non-interacting, we are actually able to prove that average of this Riemann problem gives us admissibility, which proves that evolution to $n + 1/2$ remains admissible.
- After that, we introduce a new center state $\boldsymbol{u}_i^{n+\frac{1}{2},*}$ and this state helps us to write the FVM update as a convex combination, giving admissibility there too.

Enforcing slope restriction

Slope reconstruction is performed by Zhang-Shu type limiting. We have a candidate slope σ_i and want to find θ so that this quantity becomes admissible. For that, we do just do the Zhang-Shu type limiting between the $\theta = 0$ quantity which is admissible and $\theta = 1$ which need not be admissible. Finally, this will be the slope we use.

Non-conservative reconstruction

For non-conservative variables, all our previous lemmas apply. We just have to enforce slope restrictions to ensure admissibility on these quantities separately.

ADER-DG : Predictor step

- We now do a compare of ADER and LWFR schemes. This is the degree N solution polynomial at the current time level, ℓ_i denotes the i^{th} Lagrange polynomial.
- And then we define the cell-local space time predictor solution and the flux. The unknowns are the space-time predictor values.
- We solve for predictor values by this cell local Galerkin method, performing a formal integration by parts in time
- where we impose the information of the previous time level.

ADER correction step

- To perform the correction, we take the spatial test function ℓ_i and integrate in space time
- We do the formal integration by parts in space and bring in numerical fluxes to couple neighbouring cells. To make the equivalence proof simpler, we'd do another integration parts and take quadrature points to be the solution points, enabling us to write in the FR form. Now, if we look at this equation for constant linear advection with unit speed, the $\tilde{f}_h$ will be $\tilde{u}_h$, numerical flux function would be linear and we'd be able to bring the numerical flux function out of the integral by linearity.

ADER and LWFR for constant linear advection

- the update becomes this. We can see that the time average of the predictor is showing up everywehre. For instance, the numerical flux function is using the time average.
- Let's also recall the LWFR-D2 update. And we can see, the only difference is that the LWFR scheme has replaced time average of the predictor with time average of solution, as computed by the Lax-Wendroff procedure. So, all we have to prove is that these two quantitites are the same.

Equivalence of ADER-FR and LWFR for linear case

- For the proof, we look at the local solution polynomial as a polynomial defined on the whole real line.
- We then see that the advection of that polynomial gives us a solution to the predictor.
- Since the predictor has a unique solution, it is this.
- Then, being a degree N polynomial, we also know that the Taylor's expansion terminates at N. Then, knowing the predictor exactly, we do know that we can replace the temporal derivatives with spatial we do get that the time average of the predictor equals the time average of solution approximated by Lax-Wendroff procedure, and we get our claim.

Equivalence of ADER-FR and LWFR for linear case

We can also see it numerically in these time V/S error plots. We see that the equivalence is not occuring for $N=4$. That will be understood from

Equivalence of ADER-FR and LWFR for linear case

these time very change in energy plots, which show dissipation of energy for $N=1,2,3$ but for $N=4$, we see that the energy actually increases. That is, there is a mild instability. Something similar has already been noted in literature for certain RK schemes, as we show here.

ADER-FR and LWFR for non-linear case

- For non-linear case, we can't expect equivalence, but we do have this measure of closeness upto the optimal order of accuracy, saying that both schemes are equally reliable.
- The idea is again to space time derivatives with spatial. We use the fact that the predictor equation gives this identity for all time levels greater than initial. We can extrapolate to the initial time level to approximately get
- and thus we'd get the same approximation when trying to do the Taylor's expansion.

Numerical results

Now, we see numerical results of the blending limiter, to see how it performs, starting with

Constant linear advection

constant linear advection

Convergence of pure MUSCL-Hancock method

Here, we do a convergence test where we set the blending coefficient $\alpha_e=1$ everywhere so that it is just MUSCL-Hancock with those non-cell centred grids. And, we do see the optimal second order convergence.

Optimal convergence with limiter

Here we show that optimal convergence of our limited scheme, when we are using a broad band spectrum initial data. This shows that our indicator doesn't activate when dealing with smooth data.

Composite signal

This is a composite signal which contains profiles with various levels of regularity. We compare the blending limiter with the TVB limiter. We see that the TVB limiter is developing oscillations while the blending limiter is not. The blending limiter is also more accurate than TVB in all the profiles. This is one of the only two tests where I had to tune the indicator factors, i.e., change the coefficients of constant node from 1 to something else.

1-D Euler's equations

We now see the results for 1-D Euler's equations with primitive variables being density, velocity and pressure. Here, we will test accuracy that is, resolution of smooth extrema and contact discontinuities while also testing robustness against strong shocks where there is a high risk of pressure turning negative leading to a code crash.

Shu-Osher

In this test, the initial data has a jump from a constant state to a sinusoidal state, so this is a test to see how well the limiter can maintain smooth information carried through shocks. We have zoomed in at the smooth extremas. FO means first order blending scheme, while MH is the one with MUSCL-Hancock reconstruction. We are comparing them with the TVB limiter. We can see that both the blending are a lot better than TVB and the MH blending is slightly better than FO.

Blast wave

This is a three state Cauchy problem with solid wall boundary conditions to make the two shocks interact, making it a tough test. We see that the MUSCL-Hancokc blending is clearly the best at resolving the smooth extrema and contact discontinuity.

Sedov's blast wave

In this three state problem, very high energy is stored at the single centre of the grid while energy is negligible everywhere. It's a pressure ratio of 10^6 and almost zero, so it's a very tough test of robustness and thus valididty of admissibility preservation.

Double rarefaction

In this Riemann problem, there are no shocks, just rarefactions. They are occuring because the flow is moving across opposite directions. This is a low density test to show robustness at a high resolution.

Leblanc's test

This Riemann problem is another test with very high pressure ratios at high resolution to show robustness.

2-D Results

We now look at the 2-D results

2-D composite signal

This is a linear test where the initial state consists of three profiles which rotate and return to their initial state. This cylindrical profile gets dissipated a lot when we use the TVB limiter, but the MUSCL-Hancock blending does a much better job. Looking at the maximum and minimum, we also see that blending is able to control oscillations a lot better.

Double Mach reflection

Here, a Mach 10 shock is going over a wedge inclined at 30°, which we simulate by alrigning the coordinate system with the wedge making the shock angled at 60°. In the new frame, the wedge is simulated by giving solid wall boundary conditions while the region before it is given outflow conditions. The simulation looks like this, there 2 triple points and slip line near the primary triple point is generating vorticity. To test how limiter, we see how well it resolves the voriticity.

Rest is easy to explain

If asked, vorticity is caused by shearing which is caused by opposite forces in tangential direction. That is occuring here.

Kelvin-Helmholtz stability

This is a test where vorticity is generated by putting a jump in tangential velocity in opposite directions. So, naturally, this will lead to very high vorticity generation and this will be another test to compare resolution of small scales.

2-D Sedov blast

This is another robustness test where very high energy is concentrated in the bottom left corner with negligible energy everywhere else. This leads to very high pressure ratios.

Astrophysical jet

Here we simulate jet flows at very high numbers, leading to strong bow shocks, certainly making it a robustness test. But, because of the shearing, it also becomes a test of accuracy.

Astrophysical jet Mach 80

We can see that our code is able to capture a lot more of the small scale structures.

Astrophysical jet Mach 2000

We see the same for a Mach 2000 jet.

Implementation in Julia

Type instabilities

As I said, we are satisfied by the speed of our code. But, it was not always this way. Julia has a rich type system and being a dynamic language, it is even capable of inferring object types at run time. That reduces developement time, but if we want efficiency, we have to communicate types to the compiler at all the performance-crtical parts. When object types are not inferrable at the compile time, we call it a type instability.

Here is an example to get a feel. Julia is not object oriented, so to pass the many objects around, we use containers. Using dictionaries is type unstable because dictionaries don't carry type information. So, this is not a good code. However, named tuples do carry type information, so this is type stable.

This is one example. Type instabilities can be subtle, this tool catches them well and points out exactly where you have done something wrong. You must also keep measuring the memory allocations of the code. With this tool, we can put counters in various parts of your code. If it intuitively seems there shouldn't be memory allocations somewhere but there are, something needs to be fixed. If you are not able to fix issues discovered by either of these tools, you can ask help at these three places. With this, you should be able to make the best out of Julia.

Cache blocking (picture)

Now, I wish to show one thing we learnt which also applies to other languages, which is cache blocking. We learnt it from `Trixi`, this explanation is from Akkurt Et Al. It is required when, for example computing the derivative of a flux polynomial, something done a lot in the Lax-Wendroff procedure. To compute the derivative of flux polynomial produced by evaluating a vector U, we could compute all the flux in a vector it all the points, store it in a vector F and multiply by the differentiation matrix. It works, but we are unecessarily writing to memory in the process. The right way to do it is to do it all in cache - compute flux for one point and compute its contribution to derivative and add it to the flux. SLides need to be fixed. This IS updating every flux point. This way, no unnecessary writing in the memory would be needed, we can do it all in the cache. Let's understand it through some pseudocode.

CACHE BLOCKING (CODE)

Let's understand it through code. Derivative of flux has to be computed for every cell. The bad approach would be to first compute the flux at every point and then multiply by the differentiation matrix. This is inefficient even if we use a BLAS kernel.

At each solution point, there's all conservative variables values. We load them into this static array. By passing this `eq` object, the size of this static array, which is the number of conservative variables, is known at the time of compilation. This enables it to be loaded directly into cache. We compute the flux and add the derivative contribution to the residual. We could do it by simple vector operation, but we do it efficiently by passing reminding the compiler of the size of this `f_node`.

CONCLUSIONS

-
-
- MUSCL-Hancock for non-cell centred grids proved to be admissibility preserving following the ideas of Berthon.
- By correcting Lax-Wendroff time averaged flux during the face loop, Admissibility preserving Lax-Wendroff scheme constructed.

1. REMOVED STUFF

1.1. Admissibility proof

STEP 1 : EVOLUTION TO $n+1/2$

We are breaking the proof into parts, starting by proving that the evolution to $n+1/2$ is admissible. By using the general definition of $\boldsymbol{u}_i^{*,\pm}$, we are not restricted to conservative reconstruction, and need to separately enforce admissibility constraint on the reconstructions to the face. The CFL conditions ensure characteristics of Riemann problems between these two states wouldn't intersect in appropriately sized domain with half the time step.

STEP 1 : EVOLUTION TO $n+1/2$ (2)

- This is the proof that evolution on the right is admissibile, left shall follow similarly.

  The proof requires two non-interacting Riemann problems on domains with sizes μ^-, μ^+. The left most and right most states are the left and right reconstructions respectively and $\boldsymbol{u}_i^{*,+}$ being the centre.
- And, by using the CFL condition and the definition of $\boldsymbol{u}_i^{*,+}$, we can write the cell average of the solution of this problem as the $n+\frac{1}{2}$ evolution. Since the conservation law preserves admissibility, so does this average and thus the evolution step is admissible.

STEP 2 : FVM TYPE UPDATE

- Since we have understood the admissibility to time level $n+1/2$, we now see admissibility of the FVM type update. For that we define $\boldsymbol{u}_i^{n+\frac{1}{2},*}$ in the similar spirit as before, but now with the natural objective of breaking into appropriate finite volume updates.

- We split the cell in three constant states; left, right evolution on left, right respectively with $n + 1/2, *$ in the center. Of course, to perform evolution, we will require states from the neighbouring cells as well.
- The update of three FVMs looks like this
- and when we multiply each with the weights according to the definition of $\boldsymbol{u}_i^{n+\frac{1}{2},*}$, we get the final MUSCL update to be a convex combination of these FVM updates. Thus, if these updates are admissible, MUSCL-update is admissible. For that, we require admissibility of $\boldsymbol{u}_i^{n+\frac{1}{2},*}$ and CFL conditions corresponding to the numerical flux. We write all of that in a lemma.

Step 2 : FVM type update

We now have a condition that guarantees admissibility of the MUSCL-Hancock. We just need these states to be admissible and these CFL conditions. Typically, CFL conditions would be dependent on the numerical flux, we are assuming this special case. Admissiblity of evolution to the faces comes from previous lemma. So, what we need to do next is to find conditions when $\boldsymbol{u}_i^{n+\frac{1}{2},*}$ will be admissible.

Final admissibility condition

- We expand this definition of $\boldsymbol{u}_{\mathrm{i}}^{\mathrm{n}+\frac{1}{2},*}$ to write it explicitly as
- We can see that it looks similar to the finite difference update. Thus, the proof of its admissible will go exactly the same as the first admissibility lemma we did, just that the left and right reconstructions have been swapped.
- Just as in the first lemma, we have to introduce a new state to obtain admissibility, called $\boldsymbol{u}_i^{*,*}$ and assume its admissibile with the same CFL conditions.
- And, by our choice of $\mu^{\pm}$, we do get $\boldsymbol{u}_i^{*,*} = \boldsymbol{u}_i$ for conservative reconstruction. So, this theorem is not really placing new admissibility conditions

Final admissibility condition (2)

- The proof goes the same way, the states just have to be swapped so the picture looks like a mirror image of the previous one.
- The cell average now becomes $\boldsymbol{u}_i^{n+\frac{1}{2},*}$ and we get the claim.

Final admissibility condition (3)

- Now with the previous lemmas, we have already proven that, for the conservative case, once the reconstructions and the $\boldsymbol{u}_i^{*,\pm}$ are admissible, the MH procedure preserves admissibility. Now, we are saying that the admissibility of reconstructions doesn't need to be separately enforced. That is easy because
- for the conservative case, the reconstructures are a convex combination of $\boldsymbol{u}_i^{*,\pm}$ and $\boldsymbol{u}_i^n$. Thus, that is the admissibility criterion we needed.

2. ADER-FR

ADER correction step

- To perform the correction, we take the spatial test function ℓ_i and integrate in space time

- We do the formal integration by parts in space and bring in numerical fluxes to couple neighbouring cells. Now, this is enough to define the scheme, but for easier comparison, I'd write it as an FR scheme. So, I do another integration by parts in space
- Now, to make it a collocation type scheme, we do quadrature on the solution points.
- Now, if the solution points are GL or GLL, this can be written as the FR scheme with appropriate corrector functions like this
- Now that we have the needed form, let us see what it looks like with constant linear advection equation with unit speed. The $\tilde{f}_h$ will be $\tilde{u}_h$, numerical flux function would be linear and we'd be able to bring the numerical flux function out of the integral by linearity. Overall,