Computational Fluid Dynamics
Hydrodynamics

- When we discussed PDEs, we focused so far on scalar PDEs.
- Often we wish to study systems of PDEs.
- Here we'll look at the equations of hydrodynamics
  - Nonlinear system of hyperbolic PDEs
  - We'll see many of our ideas extend to this case
  - These same ideas can be applied to other hyperbolic systems
- We model our system of interest as a fluid
- Evolution dictated by conservation laws:

\[
\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u)}{\partial x} = 0 \quad \text{Conservation of mass}
\]

\[
\frac{\partial (\rho u)}{\partial t} + \frac{\partial (\rho uu + p)}{\partial x} = 0 \quad \text{Conservation of momentum}
\]

\[
\frac{\partial (\rho E)}{\partial t} + \frac{\partial (\rho u E + up)}{\partial x} = 0 \quad \text{Conservation of energy}
\]
Conservative System

- This can be written in conservation law form:
  \[ U_t + [F(U)]_x = 0 \]
  - With

\[
  U = \begin{pmatrix}
    \rho \\
    \rho u \\
    \rho E
  \end{pmatrix} \quad F(U) = \begin{pmatrix}
    \rho u \\
    \rho uu + p \\
    \rho uE + up
  \end{pmatrix}
\]

- To close this system, we need an equation of state
- Simplest: gamma-law

\[
p = \rho e(\gamma - 1)
\]

\[
  \rho e = \rho E - \frac{1}{2} \rho u^2
\]
Conservative System

- To make it look more like advection, we write this in quasi-linear form.

  Express flux vector in terms of $m \equiv \rho u$, $\mathcal{E} \equiv \rho \mathcal{E}$

  $$p = \rho e(\gamma - 1) = \left( \mathcal{E} - \frac{1}{2} \frac{m^2}{\rho} \right) (\gamma - 1)$$

  $$F(U) = \begin{pmatrix} m \\ \frac{1}{2} \frac{m^2}{\rho} (3 - \gamma) + \mathcal{E}(\gamma - 1) \\ \frac{m \mathcal{E}}{\rho} \gamma - \frac{1}{2} \frac{m^3}{\rho^2} (\gamma - 1) \end{pmatrix}$$
Conservative System

- Compute the Jacobian:

\[
A(U) = \frac{\partial F}{\partial U} = \begin{pmatrix}
0 & 1 & 0 \\
-\frac{1}{2} u^2 (3 - \gamma) & u(3 - \gamma) & \frac{\gamma - 1}{2} \\
\frac{1}{2} (\gamma - 2) u^3 - \frac{uc^2}{\gamma - 1} & \frac{3 - 2\gamma}{2} u^2 + \frac{c^2}{\gamma - 1} & u \gamma
\end{pmatrix}
\]

\[
c = \sqrt{\gamma p/\rho}
\]

- The system can now be written as:

\[
U_t + A(U)U_x = 0
\]

Note: this actually is only true in Cartesian coordinates, since the pressure enters the momentum equation as a gradient, not a divergence.
Primitive Variable Formulation

- We can instead cast things in terms of the **primitive variables**: density, velocity, and pressure

- Our system becomes *(blackboard)*:

  \[ q_t + A(q)q_x = 0 \]

  \[ q = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix} \quad A(q) = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \gamma p & u \end{pmatrix} \]

  - Notice that the Jacobian for this formulation is much simpler
Consider the full time-rate-of-change of a fluid quantity:

\[
\frac{d}{dt} \phi(x, t) = \frac{\partial \phi}{\partial t} + \frac{\partial \phi}{\partial x} \frac{dx}{dt}
\]

- This depends on the path \( x(t) \) we choose to follow.

**Eulerian frame:**
- We keep our reference position fixed in time, and we watch the flow move past us, so \( dx/dt = 0 \)

**Lagrangian frame:**
- We pick a path that is equal to the fluid velocity—this way we track the flow of an individual fluid element
- The Lagrangian derivative (also called the material, convective, advective, ... derivative) is

\[
\frac{D}{Dt} = \frac{\partial}{\partial t} + u \frac{\partial}{\partial x}
\]
Primitive Variable Formulation

- Note: we don't need to assume a gamma-law gas—we can derive this for a general equation of state too

- Express: \( p = p(\rho, s) \)

\[
\frac{Dp}{Dt} = \frac{\partial p}{\partial \rho} \left|_s \right. \frac{D\rho}{Dt} + \frac{\partial p}{\partial s} \left|_\rho \right. \frac{Ds}{Dt}
\]

- Where the full derivative of entropy is zero when there are no heat sources (reactions, diffusion, ...)

- The first adiabatic index is:

\[
\Gamma_1 \equiv \frac{\partial \log p}{\partial \log \rho} \left|_s \right.
\]

- Giving:

\[
\frac{\partial p}{\partial t} + u \frac{\partial p}{\partial x} + \Gamma_1 p \frac{\partial u}{\partial x} = 0
\]
Recall that a system is hyperbolic if the eigenvalues are real and finite.

For our system, the eigenvalues are:

\[ \lambda^{(-)} = u - c, \lambda^{(\circ)} = u, \lambda^{(+)} = u + c \]

- These are the speeds at which information propagates in our system
- Three distinct wave speeds for 3 equations
- We'd get the same eigenvalues from the Jacobian of the conserved system

There is a rich mathematical description of the theory of hyperbolic systems of conservation laws.

- The book by LeVeque is an excellent introduction.
Eigensystem

- We can also find the eigenvectors:

\[ A r^{(\nu)} = \lambda^{(\nu)} r^{(\nu)} ; \quad l^{(\nu)} A = \lambda^{(\nu)} l^{(\nu)} \]

\[
\begin{align*}
  r^{(-)} &= \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \end{pmatrix} \\
  r^{(\circ)} &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
  r^{(+)} &= \begin{pmatrix} 1 \\ c/\rho \\ c^2 \end{pmatrix}
\end{align*}
\]

\[
\begin{align*}
  l^{(-)} &= \begin{pmatrix} 0 & -\rho/2c & 1/2c^2 \end{pmatrix} \\
  l^{(\circ)} &= \begin{pmatrix} 1 & 0 & -1/c^2 \end{pmatrix} \\
  l^{(+)} &= \begin{pmatrix} 0 & \rho/2c & 1/2c^2 \end{pmatrix}
\end{align*}
\]

- These are normalized such that \( l^{(i)} \cdot r^{(j)} = \delta_{ij} \)
A final form of the system is in terms of the characteristic variables

- Construct matrices of the left and right eigenvectors

\[
R = \begin{pmatrix}
    r^{(-)} & r^{(\circ)} & r^{(+)}
\end{pmatrix}
\]

\[
L = \begin{pmatrix}
    l^{(-)} \\
    l^{(\circ)} \\
    l^{(+)}
\end{pmatrix}
\]

- Satisfy: \( LR = RL = I \)

- Define \( dw = L \, dq \)

- Our system can be written as:

\[
w_t + \Lambda w_x = 0
\]
Characteristic Variables

- Here, \( w \) are the characteristic variables
- The three equations are decoupled:

\[
\Lambda = \mathbf{LAR} = \begin{pmatrix}
\lambda^{(-)} & \lambda^{(o)} \\
\lambda^{(o)} & \lambda^{(+)}
\end{pmatrix}
\]

- If our system were linear, we'd be done:
  - Transform into characteristic variables
  - Each characteristic variable advects at a given wave speed, without interacting with one-another
  - Solve and then transform back to primitive form (or conserved form)
- We're non-linear: the wave-speeds and eigenvectors change with the solution
Jumps Across Waves

- The characteristic system is telling us something interesting already.
  - Consider an initial discontinuity in the primitive variables
  - Each wave will carry a jump in their associate characteristic quantity away from the discontinuity at their speed
  - The corresponding jump in the primitive variable is just \( dq = L^{-1}dw = R \, dw \)
Jumps Across Waves

\[ \lambda_1 = u - c \quad \lambda_2 = u \quad \lambda_3 = u + c \]

All primitive quantities jump across the \( u-c \) wave

Only density jumps across the middle \( (u) \) wave

All primitive variables jump across the \( u+c \) wave

\[
\begin{align*}
r^{(-)} &= \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \end{pmatrix} \\
r^{(\circ)} &= \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \\
r^{(+)} &= \begin{pmatrix} 1 \\ c/\rho \\ c^2 \end{pmatrix}
\end{align*}
\]
$$\rho_L = 1.0$$
$$\rho_R = 1/8$$
$$u_L = 0$$
$$u_R = 0$$
$$p_L = 1.0$$
$$p_R = 1/10$$
Solution Methodology

- This system has similarities to the advection equation we already studied.
  - We can take a similar approach
- We will use the finite-volume discretization:
  \[
  \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} U_t = -\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \frac{\partial}{\partial x} F(U) \, dx
  \]
  \[
  \frac{\partial}{\partial t} U_i = -\frac{1}{\Delta x} \left\{ [F(U)]_{i+1/2} - [F(U)]_{i-1/2} \right\}
  \]
  - Second-order in time requires time-centering the righthand side
  \[
  U_{i}^{n+1} = U_i^n + \frac{\Delta t}{\Delta x} (F_{i-1/2}^{n+1/2} - F_{i+1/2}^{n+1/2})
  \]
Solution Methodology

- As we saw in our homework (Burgers' eq.), we need to solve this in conservative form to get the shock speed correct.

- Basic idea: evaluate fluxes at interfaces by predicting the fluid state there, then solve Riemann problem:

\[
\begin{align*}
F_{i+1/2}^{n+1/2} &= F(U_{i+1/2}^{n+1/2}) \\
U_{i+1/2}^{n+1/2} &= \mathcal{R}(U_{i+1/2,L}^{n+1/2}, U_{i+1/2,R}^{n+1/2})
\end{align*}
\]
Our update looks like:

\[ U_i^{n+1} = U_i^n + \frac{\Delta t}{\Delta x} \left( F_{i-1/2}^{n+1/2} - F_{i+1/2}^{n+1/2} \right) \]

- Where are fluxes are:

\[ F_{i+1/2}^{n+1/2} = F(U_{i+1/2}^{n+1/2}) \]

\[ U_{i+1/2}^{n+1/2} = R(U_{i+1/2}^{n+1/2}, L, U_{i+1/2}^{n+1/2}, R) \]

- If we construct the interface to second-order, then the overall method will be second-order accurate
The simplest method to construct the interface states is to simply use the cell averages

\[ U_{i+1/2, L} = U_i ; \quad U_{i+1/2, R} = U_{i+1} \]

- This is first order accurate in space and time.
- This method is called Godunov's method.
- Doesn't consider how \( U \) is changing in space or over the timestep.
We can get second-order accuracy by reconstructing the cell-average data to be piecewise linear.

- It is easier to work with the primitive variables.
- We reconstruct each quantity and limit the slopes (just like we did with advection).
- Predict edge-centered, time-centered state via Taylor expansion (blackboard...).
• We have:

\[ q_{i+1/2, L}^{n+1/2} = q_i^n + \frac{1}{2} \left[ 1 - \frac{\Delta t}{\Delta x} A \right] \Delta q \]

- This is the amount of the primitive variable that reaches the interface over the timestep
- Recall that the jumps in \( q \) are carried by 3 waves
- We only want to include a jump if that wave was moving toward the interface—we need to do a characteristic decomposition

\[ q_{i+1/2, L}^{n+1/2} = q_i^n + \frac{1}{2} \left[ RL - \frac{\Delta t}{\Delta x} R\Lambda L \right] \Delta q \]

= \[ q_i^n + \frac{1}{2} \sum_{\nu} \left[ 1 - \frac{\Delta t}{\Delta x} \lambda^{(\nu)} \right] (l^{(\nu)} \cdot \vec{\Delta q}) r^{(\nu)} \]

(blackboard derivation...)
Piecewise Linear Reconstruction

- Notice the quantity: $\lambda t/\Delta x$ — this is the CFL number for each wave.

- To consider only the waves moving toward the interface, we alter the sum slightly:

$$q_{i+1/2,L}^{n+1/2} = q_i^n + \frac{1}{2} \sum_{\nu: \lambda^{(\nu)} \geq 0} \left( 1 - \frac{\Delta t}{\Delta x} \lambda^{(\nu)} \right) (l^{(\nu)} \cdot \overline{\Delta q}) r^{(\nu)}$$

- The right state at this interface is similarly constructed:

$$q_{i+1/2,R}^{n+1/2} = q_{i+1}^n - \frac{1}{2} \sum_{\nu: \lambda^{(\nu)} \leq 0} \left( 1 + \frac{\Delta t}{\Delta x} \lambda_{i+1}^{(\nu)} \right) (l_{i+1}^{(\nu)} \cdot \overline{\Delta q_{i+1}}) r_{i+1}^{(\nu)}$$
Piecwise Linear Reconstruction

• The decomposition of the jumps into the sum over the left and right eigenvectors is sometimes called a characteristic projection
  
  – Note that we explicitly see here that each wave carries a jump proportional to the right eigenvector
  
  – \((1 \cdot \Delta q)\) is the projection of the primitive variable jump into the characteristic variables

• Many sources introduce a reference state chosen so as to minimize the effect of the characteristic decomposition—this is because we are linearizing a nonlinear system

\[
q_{i+1/2,L}^{n+1/2} = \tilde{q}^+ - \sum_{\nu; \lambda^{(\nu)} \geq 0} l_i^{(\nu)} \cdot \left\{ \tilde{q}^+ - \left[ q_i^n + \frac{1}{2} \left( 1 - \frac{\Delta t}{\Delta x} \lambda_i^{(\nu)} \right) \Delta q_i \right] \right\} r_i^{(\nu)}
\]
• The most popular method in astrophysics is PPM (piecewise parabolic method)
  – Reconstruct the data in each cell as a parabola
  – The parabolic profiles are limited to prevent generation of new extrema (see Colella & Woodward 1984 for a description of the original method)
  – General form:

\[ q_i(x) = q_{-i} + \xi(x)(\Delta q_i + q_{6i}(1 - \xi(x))) \]

\[ \xi(x) = \frac{x - x_{i-1/2}}{\Delta x} \]
Piecewise Parabolic

- The reconstruction polynomial recovers the cell-average:

\[
\frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x) \, dx = q_i
\]

- To use the parabolic interpolant, we start with our interface state for piecewise linear:

\[
q_{i+1/2,L}^{n+1/2} = \tilde{q}_+ - \sum_{\nu; \lambda^{(\nu)} \geq 0} l^{(\nu)}_i \cdot \left\{ \tilde{q}_+ - \left[ q_i^n + \frac{1}{2} \left( 1 - \frac{\Delta t}{\Delta x} \lambda^{(\nu)}_i \right) \overline{\Delta q_i} \right] \right\} r^{(\nu)}_i
\]

- Recognize that for a linear profile:

\[
q_i^n + \frac{1}{2} \left( 1 - \frac{\Delta t}{\Delta x} \lambda^{(\nu)}_i \right) \overline{\Delta q_i} \approx \frac{1}{\lambda \Delta t} \int_{x_{i+1/2} - \lambda \Delta t}^{x_{i+1/2}} q(x) \, dx
\]
Piecewise Parabolic

- This suggests we can insert the average under our parabolic profile to construct the interface state:

$$I_+^{(\nu)}(q_i) = \frac{1}{\sigma^{(\nu)} \Delta x} \int_{x_{i+1/2}-\sigma^{(\nu)} \Delta x}^{x_{i+1/2}} q(x) dx$$

- Where

$$\sigma^{(\nu)} = |\lambda^{(\nu)}| \Delta t / \Delta x$$
• The PPM interface states are then:

\[ q_{i+1/2, L}^{n+1/2} = \tilde{q}_+ - \sum_{\nu; \lambda(\nu) \geq 0} l_i^{(\nu)} \cdot \left( \tilde{q}_+ - \mathcal{I}_+^{(\nu)}(q_i) \right) r_i^{(\nu)} \]

\[ q_{i-1/2, R}^{n+1/2} = \tilde{q}_- - \sum_{\nu; \lambda_\nu \leq 0} l_i^{(\nu)} \cdot \left( \tilde{q}_- - \mathcal{I}_-^{(\nu)}(q_i) \right) r_i^{(\nu)} \]

– Where

\[ \mathcal{I}_-^{(\nu)}(q) = \frac{1}{\sigma^{(\nu)} \Delta x} \int_{x_{i-1/2}}^{x_{i-1/2} + \sigma^{(\nu)} \Delta x} q(x) \, dx \]
Riemann Problem

- No matter the method used to predict the interface states, we now have left and right states at each interface.

\[
\begin{array}{c|cc|c}
U_i & U_i^{n+1/2} & \times U_{i+1/2}^{n+1/2} & U_{i+1} \\
i & i+1/2 & i+1 \\
\end{array}
\]

- Unlike the linear advection or Burger's equation, we rarely solve the Riemann problem for the Euler equations exactly.
  - We need to consider what is carried by each wave.
  - Different types of waves are present depending on the behavior of the characteristics.
**Riemann Problem**

- Wave type is determined by whether the characteristics converge, are parallel, or diverge

- We already saw shocks and rarefactions with Burger's equation
- A contact discontinuity is where the solution jumps, but there is no compression or expansion
Riemann Problem

- Across the middle wave ($\lambda = u$), only the density jumps

\[ r^{(-)} = \begin{pmatrix} 1 \\ -c/\rho \\ c^2 \end{pmatrix} \quad r^{(\circ)} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad r^{(+)} = \begin{pmatrix} 1 \\ c/\rho \\ c^2 \end{pmatrix} \]

- So the velocity is the same on either side—no convergence or divergence.

- The middle wave is always a contact discontinuity

- The left and right waves can be either a shock or rarefaction
The Riemann problem for the Euler equations looks like:

\[ dx = (u - c)dt \]

\[ t \]

\[ x = ut \]

\[ dx = (u + c)dt \]

\[ \rho_*, L, u_*, p_* \]

\[ \rho_L, u_L, p_L \]

\[ \rho_*, R, u_*, p_* \]

\[ \rho_R, u_R, p_R \]

- Solving the Riemann problem means finding the 4 quantities:
  \[ \rho_*, L, \rho_*, R, u_*, p_* \]
Wave Structure

- There are several different wave configurations

1-rarefaction and 3-shock

1-shock and 3-rarefaction

two shocks

two rarefactions
Wave Structure

- And the can span the initial interface or all be on one side

supersonic flow to the left

supersonic flow to the right
Riemann Problem

- The basic solution idea is to link the left and right states (which we know) to the star state using our understanding of what happens across the left and right waves.
If we are not compressing, then we are a rarefaction.

Let's look at the system of equations with entropy instead of pressure as a variable.

- Entropy evolution: $D\frac{s}{Dt} = 0$
- We need to replace the pressure gradient in the momentum equation:

$$\frac{\partial p(\rho, s)}{\partial x} = \frac{\partial p}{\partial s}\frac{\partial s}{\partial x} + \frac{\partial p}{\partial \rho}\frac{\partial \rho}{\partial x} = \frac{\partial p}{\partial s}\frac{\partial s}{\partial x} + \frac{p\Gamma_1}{\rho} \frac{\partial \rho}{\partial x}$$
Riemann Problem: Rarefaction

- System becomes:

\[
\frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial x} + \rho \frac{\partial u}{\partial x} = 0
\]

\[
\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + \frac{1}{\rho} \left[ \frac{\partial p}{\partial s} \left|\begin{array}{c}
\frac{\partial s}{\partial \rho} \left|\begin{array}{c}
\frac{\partial \rho}{\partial s} + p \Gamma_1 \frac{\partial \rho}{\partial x}
\end{array}\right.
\end{array}\right. = 0
\]

\[
\frac{\partial s}{\partial t} + u \frac{\partial s}{\partial x} = 0
\]

or

\[
\begin{pmatrix}
\rho \\
u \\
s
\end{pmatrix}_t + \begin{pmatrix}
u & \rho & 0 \\
c^2/\rho & u & 1/\rho \frac{\partial p}{\partial s} \\
0 & 0 & u
\end{pmatrix} \begin{pmatrix}
\rho \\
u \\
s
\end{pmatrix}_x = 0
\]
Riemann Problem: Rarefaction

- The eigenvalues of that matrix are the same, but the eigenvectors will tell us how entropy behaves across each wave:

\[ r^{(-)} = \begin{pmatrix} 1 \\ -c/ho \\ 0 \end{pmatrix} \quad r^{(\circ)} = \begin{pmatrix} 1 \\ 0 \\ -c^2/p_s \end{pmatrix} \quad r^{(+) = \begin{pmatrix} 1 \\ c/ho \\ 0 \end{pmatrix} \]

- Entropy only jumps across the contact, not the left and right way

- Considering entropy is only really useful for rarefactions
  - Shocks involve dissipative processes, so entropy needs to increase
  - This is why you need to solve a total energy instead of an entropy equation for shocks
Riemann Problem: Rarefaction

- Consider waves originating at a discontinuity (this is the Riemann problem)
- We come in with a jump in the primitive variables, $q$
  - The corresponding jump in characteristic variables is $\Delta w = L \Delta q$
  - Each wave carries a jump in its characteristic variable
- Consider the right (+) wave:
  - It moves at speed $u + c$
  - A piece of $\Delta q$ will jump across this wave (let's call the quantity that jumps $w^{(+)}$)
  - But $w^{(+)}$ will be constant across the other waves then
  - We can use this to link the states across the left (-) wave
Riemann Problem: Rarefaction

- $w^{(+)}$ is constant across the (-) wave. This is defined as

$$l^{(+)} \cdot dq = 0$$

- This is:

$$
\begin{pmatrix}
0 & \frac{\rho}{2c} & \frac{1}{2c^2}
\end{pmatrix}
\begin{pmatrix}
\frac{d\rho}{du} \\
\frac{du}{dp}
\end{pmatrix} = 0
$$

$$du + \frac{dp}{\rho c} = 0$$

- The general solution to this is:

$$u = - \int \frac{dp}{\rho c}$$

- We can't integrate this for a general EOS
Riemann Problem: Rarefaction

- If we assume a gamma-law ideal gas, then we know

\[ p = K \rho^\gamma \]

- for isentropic flows
- \( \gamma \) is the constant ratio of specific heats

- Integration of this gives (blackboard):

\[ u + \frac{2c}{\gamma - 1} = \text{constant} \quad \text{across left rarefaction} \]

- Similarly, we'd find

\[ u - \frac{2c}{\gamma - 1} = \text{constant} \quad \text{across right rarefaction} \]
Riemann Problem: Rarefaction

- We can now connect the left and star state across a rarefaction:

\[ u_L + \frac{2c_L}{\gamma - 1} = u_* + \frac{2c_*}{\gamma - 1} \]

\[ u_* = u_L + \frac{2c_L}{\gamma - 1} \left[ 1 - \left( \frac{c_*}{c_L} \right) \right] \]

- We also know:

\[ \frac{p_L}{\rho L} = \frac{p_*}{\rho_*} \]

- So:

\[ \frac{c_*}{c_L} = \left( \frac{p_*}{p_L} \frac{\rho_l}{\rho_*} \right)^{1/2} = \left( \frac{p_*}{p_L} \right)^{(\gamma - 1)/2\gamma} \]

and

\[ u_* = u_L + \frac{2c_L}{\gamma - 1} \left[ 1 - \left( \frac{p_*}{p_L} \right)^{(\gamma - 1)/2\gamma} \right] \]
Riemann Problem: Shock

- If the flow is compressing, then we have a shock
  - Rankine-Hugoniot jump conditions provide the solution
  - We already saw this when we did Burgers' equation

- We'll work in the frame of the shock, and we'll only consider the left wave
  - Jump conditions become:
    \[
    \frac{\hat{F}_* - \hat{F}_L}{\hat{U}_* - \hat{U}_L} = 0
    \]
  - Where the hat denotes the frame of the shock,
    \[
    \hat{u}_L = u_L - S_L \\
    \hat{u}_* = u_* - S_L
    \]
Riemann Problem: Shock

- Jump conditions are:

\[
\begin{align*}
\rho_L \hat{u}_L &= \rho_* \hat{u}_* \\
\rho_L \hat{u}_L^2 + p_L &= \rho_* \hat{u}_*^2 + p_* \\
\rho_L e_L \hat{u}_L + p_L \hat{u}_L + \frac{1}{2} \rho_L \hat{u}_L^3 &= \rho_* e_* \hat{u}_* + p_* \hat{u}_* + \frac{1}{2} \rho_* \hat{u}_*^3
\end{align*}
\]

- With a bit of algebra (blackboard), we have:

\[
\begin{align*}
[\tau] &= -\frac{[p]}{W_L^2} \\
[u] &= -\frac{[p]}{W_L} \\
e \tau] &= -\bar{p}[\tau] \\
[q] &\equiv q_* - q_L \\
\tau &= 1/\rho \\
\bar{p} &= \frac{1}{2}(p_L + p_*) \\
W_L &= \rho_L \hat{u}_L = \rho_* \hat{u}_*
\end{align*}
\]

Note that: \( \hat{u}_* - \hat{u}_L = u_* - u_L \) 

This is the mass flux
Riemann Problem: Shock

- How do we solve these?
  - We will come in with a \( p_\star \) (more on this in a minute)
  - Root find on:

\[
\begin{align*}
  e_\star &= e(p_\star, \rho_\star) \\
  e_\star - e_L &= -\frac{1}{2}(p_\star + p_L) \left( \frac{1}{\rho_\star} - \frac{1}{\rho_L} \right)
\end{align*}
\]

- Get the mass flux,

\[
\frac{1}{W_L^2} = -\frac{[\tau]}{[p]}
\]

- Get the star velocity:

\[
u_\star = u_L - \frac{p_\star - p_L}{W_L}
\]
Summary of Riemann Problem

- We need to consider the action of all 3 waves
  - Middle wave is always a contact
  - Left and right can be either shock or rarefaction

- Rarefaction:
  - Entropy is constant across the wave
  - Riemann invariants tell us how to connect the solution across the wave to the star region

- Shock:
  - Must be dissipative
  - Jump conditions tell us how to connect to the star state across the shock
Wave Structure

- The characteristic structure becomes:

Using the correct relation across each wave allows us to link the 4-states and solve the Riemann problem.
For a gamma-law gas, the shock solution can be found analytically, and we get:

\[
\begin{align*}
   u_{*,L}(p_*) &= u_L + \begin{cases} 
      \frac{2c_L}{\sqrt{2\gamma(\gamma-1)}} \frac{1-p_*/p_L}{\sqrt{1+\frac{\gamma+1}{\gamma-1} \frac{p_*}{p_L}}} & p_* > p_L \\
      \frac{2c_L}{\gamma-1} \left[ 1 - \left(\frac{p_*/p_L}\right)^{(\gamma-1)/(2\gamma)} \right] & p_* \leq p_L
   \end{cases} \\
   u_{*,R}(p_*) &= u_R - \begin{cases} 
      \frac{2c_R}{\sqrt{2\gamma(\gamma-1)}} \frac{1-p_*/p_R}{\sqrt{1+\frac{\gamma+1}{\gamma-1} \frac{p_*}{p_R}}} & p_* > p_R \\
      \frac{2c_R}{\gamma-1} \left[ 1 - \left(\frac{p_*/p_R}\right)^{(\gamma-1)/(2\gamma)} \right] & p_* \leq p_R
   \end{cases}
\end{align*}
\]

(LeVeque Eq. 14.51, 14.52)

- The constancy of the pressure and velocity across the contact gives:

\[
   u_{*,L}(p_*) = u_{*,R}(p_*)
\]
Riemann Solution

- This can be solved using standard root finding techniques
- Gets you $u_*, p_*$
- The density to the left and right of the contact can be found via the Riemann invariants (rarefaction) or jump conditions (shock)
  - This completes the entire solution
Riemann Solution

- We can look at the solution to the Riemann problem graphically
- Sod problem:
  \[ \rho_L = 1, \ u_L = 0, \ p_L = 1 \]
  \[ \rho_R = 1/8, \ u_R = 0, \ p_R = 1/10 \]
  - Plot shows the curves that each state can reach through a shock (dashed) or rarefaction (solid)
  - Solution is the intersection
  - Here we must have a left rarefaction and a right shock

\[ \text{code: riemann-phases.py} \]
\[ \begin{align*}
  p^* &= 0.303130178051 \\
  u^* &= 0.927452620049
\end{align*} \]
\[ \rho_L = 1.0 \]
\[ \rho_R = 1/8 \]
\[ u_L = 0 \]
\[ u_R = 0 \]
\[ p_L = 1.0 \]
\[ p_R = 1/10 \]
Sampling the Solution

- We only need to know the Riemann solution on our interface

- Which region are we in?
  - We need to evaluate the various speeds
Sampling the Solution

- If the left or right wave is a shock, the speed comes from our jump conditions
  - Left: we had
    \[ W_L = \rho_L \hat{u}_L = \rho_L (u_L - S) \quad \rightarrow \quad S = u_L - \frac{W_L}{\rho} \]
  - An analytic expression for this can be written down for a gamma-law gas
  - A similar expression (but with a '+' sign) is found for a shock at the right wave

- For a contact, we just use \( S_0 = u_\ast \)

- For a rarefaction, there are 2 speeds
  - Consider the left rarefaction
    - Head travels at: \( S_h = u_L - c_L \)
    - Tail travels at: \( S_t = u_\ast - c_\ast \)
Sampling the Solution

- To complete the rarefaction speed we need the density in the star region:
  - Gamma-law gas:
    - We know entropy is constant, so we can just do:
      \[ \frac{p_L}{\rho_L^\gamma} = \frac{p_\star}{\rho_\star^\gamma} \]
  - General gas:
    - Use the contact's characteristic variable:
      \[ l^{(0)} \cdot dq = \begin{pmatrix} 1 & 0 & -\frac{1}{c^2} \end{pmatrix} \cdot \begin{pmatrix} \frac{dp}{d\rho} \\ du \\ dp \end{pmatrix} = 0 \]
      \[ d\rho = \frac{dp}{c^2} \]
      - This can be integrated across the rarefaction
Sampling the Solution

- We evaluate the solution to the Riemann problem along $x/t = 0$

- If $S_0 > 0$, then we are either the left or left-star state
  - The check the speed of the left shock / rarefaction to determine left or left-star
    - This gives you the state on the interface
  - If $S_0 < 0$, then we are either the right or right-star state
Sampling the Solution

- Caveat: the rarefaction could span the axis
  - \( S_{\text{head}} < 0 < S_{\text{tail}} \)
  - We can integrate the Riemann invariants to get the state inside the rarefaction
Riemann Solution

Once we get the state on the interface, we can construct the fluxes through the interface:

\[
F_{i+1/2}^{n+1/2} = \begin{pmatrix}
\rho_{i+1/2}^{n+1/2} u_{i+1/2}^{n+1/2} \\
\rho_{i+1/2}^{n+1/2} (u_{i+1/2}^{n+1/2})^2 + p_{i+1/2}^{n+1/2} \\
\frac{u_{i+1/2}^{n+1/2} p_{i+1/2}^{n+1/2}}{\gamma - 1} + \frac{1}{2} \rho_{i+1/2}^{n+1/2} (u_{i+1/2}^{n+1/2})^3 + u_{i+1/2}^{n+1/2} p_{i+1/2}^{n+1/2}
\end{pmatrix}
\]
General Solution

- We solve the Euler equations for a general problem (not a single jump) by constructing the interface states and then solving a Riemann problem at each interface
  - The Riemann problem captures the shock—we get the shock speed correct
  - The overall method is conservative
Conservation

- Recall that when you did your Burgers' equation homework, the non-conservative differencing gave you the wrong shock speed.
The structure of your code is nearly identical to the codes we did for advection:

- **Initialize**
  - We need to initialize all the conserved state, \( U \)

- **Evolve**
  - **Compute timestep**
    - We restrict things so that no information can cross more than one zone in a timestep:
      \[
      \Delta t = \min_i \frac{\Delta x}{|u_i| + c}
      \]
  - **Fill ghost cells**
    - The number of ghost cells will depend on the width of our stencil
    - PPM requires 4 ghost cells on each end of the domain
– Evolve (continued)
  • Construct interface states
    – We looked at piecewise constant, piecewise linear, and piecewise parabolic reconstruction
    – In all cases, we wind up with a left and right primitive variable state at every interface

  ![Diagram showing the solve Riemann problem process](image)

  • Solve Riemann problem
    – This takes the time-centered, edge-centered primitive variable states and returns the state on the interface
Structure of a Simulation Code

- Evolve (continued)
  - Conservative update:

\[
U_{i}^{n+1} = U_{i}^{n} + \frac{\Delta t}{\Delta x} \left( F_{i-1/2}^{n+1/2} - F_{i+1/2}^{n+1/2} \right)
\]
Boundary Conditions

- Boundary conditions should match the physics you are modeling
- Common ones:
  - **Reflecting / solid wall**: ghost cells reflect their partner across the physical boundary. Normal velocity component changes sign
  - **Outflow**: zero-gradient given to all ghost cells
    - Note: this is not perfect, because at the boundary, some characteristic waves may flow inward (for subsonic flows)
  - **Hydrostatic**: in the presence of gravity, provide pressure support to hold up an atmosphere
Solving the Riemann problem (even for a gamma-law gas) can be expensive

- Often we use approximations
- **Approximate state methods:**
  - Approximate the primitive variable state in the star region
  - Some examples: two-shock and two-rarefaction solvers
- **Approximate flux methods:**
  - Approximate the flux directly from the jump conditions without getting an approximate state
  - Some examples: HLL methods
- **Linearized methods:**
  - Linearize the system and solve the Riemann problem as you would for a linear hyperboic system
  - Example: Roe solver
Bells and Whistles

- We described the basics, but there are some other features typically implemented in hydro codes:
  - **Flattening**: shocks self-steepen (characteristics converge), and can become too sharp. Flattening adds some dissipation near shocks to smear them out a bit.
  - **Species advection**: nuclear or chemical species can be advected with the flow:
    \[
    \frac{\partial X_k}{\partial t} + u \frac{\partial X_k}{\partial x} = 0
    \]
  - **Source terms**: source terms (e.g. gravity) do not change the characteristic structure, but do participate in the prediction to the interfaces.
  - **Other geometries**: extensions to cylindrical and spherical geometries involves volume and area factors throughout the code.
Test Problems

- There are several standard test problems with known analytic solutions
- Simplest is to just run a Riemann problem (shock tube) with your solver
Sod Problem

- Sod problem exhibits all 3 types of waves
  \[ \rho_L = 1, u_L = 0, p_L = 1 \]
  \[ \rho_R = 1/8, u_R = 0, p_R = 1/10 \]

- Setup:
  - 128 zones
  - Piecewise constant reconstruction
  - 2-shock approximate Riemann solver

- Notice how smeared out the contact is

\( \text{code: hydro1d} \)
**Sod Problem**

- With piecewise parabolic profiles, all the waves look better
- Contact is still the worst—no self-steepening

code: hydro1d
Double Rarefaction

- Divergent flow:
  \[ \rho_L = 1, u_L = -2, p_L = 0.4 \]
  \[ \rho_R = 1, u_R = 2, p_R = 0.4 \]

- This generates a vacuum state

- Setup:
  - 128 zones
  - Piecewise constant reconstruction
  - 2-shock approximate Riemann solver

\[ \text{code: hydro1d} \]
Double Rarefaction

- Much better flow with ppm

code: hydro1d
Double Rarefaction

- But solvers have a hard time getting the internal energy right in the vacuum region
  - $e$ from ppm solution
Strong Shock

- Strong shock followed closely by contact
  \[ \rho_L = 1, \ u_L = 0, \ p_L = 1000 \]
  \[ \rho_R = 1, \ u_R = 0, \ p_R = 0.01 \]

- Very difficult to get the shock structure correct

- Setup:
  - 128 zones
  - Piecewise constant reconstruction
  - 2-shock approximate Riemann solver

code: hydro1d
Strong Shock

- We do better with ppm reconstruction
Different Coordinate Systems

- Consider 1-d spherical coordinates
  \[
  \frac{\partial U}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 F) = 0 \quad \rightarrow \quad \frac{\partial U}{\partial t} + \frac{\partial F}{\partial r} = -\frac{2F}{r}
  \]
  - This looks like a Cartesian system with a geometric source term

- Strategy:
  - When doing the prediction to the interfaces, explicitly include the geometric source term

  \[
  \frac{\partial \rho}{\partial t} + u \frac{\partial \rho}{\partial r} + \rho \frac{\partial u}{\partial r} = -\frac{2\rho u}{r}
  \]
  \[
  \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial r} + \frac{1}{\rho} \frac{\partial p}{\partial r} = 0
  \]
  \[
  \frac{\partial p}{\partial t} + u \frac{\partial p}{\partial r} + \gamma p \frac{\partial u}{\partial r} = -\frac{2\gamma pu}{r}
  \]
  - Eigenvectors don't change here
Different Coordinate Systems

- Do the conservative update with the actual spherical divergence
  - Note: the pressure term does not appear in the momentum flux anymore
  - We difference the momentum equation as:

\[
(\rho u)_i^{n+1} = (\rho u)_i^n - \Delta t \frac{(r^2 \rho u^2)_{i+1/2}^{n+1/2} - (r^2 \rho u^2)_{i-1/2}^{n+1/2}}{r_i^2 \Delta r} - \Delta t \frac{p_{i+1/2}^{n+1/2} - p_{i-1/2}^{n+1/2}}{\Delta r}
\]
Sedov-Taylor Blast Wave

- Dump a lot of energy into a single point
- Ambient pressure is negligible
- Analytic solution for spherical and cylindrical geometries was worked out by Sedov (1959) Taylor, and others
- Can be difficult for a code to model since the point-energy is spread over a zone
  - Standard method for initializing spreads the energy over a few zones, gets the corresponding pressure, and initializes using pressure
Sedov-Taylor Blast Wave

- PPM solution (with 2-shock Riemann solver)
- 128 zones on [0, 0.5]
Multidimensions

- What happens to the transverse velocity in a multi-d system?
- 2-d conserved system:

\[ U_t + [F^x(U)]_x + [F^y(U)]_y = 0 \]

\[
F^x = \begin{pmatrix}
\rho u \\
\rho u^2 + p \\
\rho uv \\
\rho uE + up \\
\end{pmatrix}, \quad F^y = \begin{pmatrix}
\rho v \\
\rho uv \\
\rho v^2 + p \\
\rho vE + vp \\
\end{pmatrix}
\]
Multidimensions

- Primitive variable system:

\[ q_t + A^{(x)} q_x + A^{(y)} q_y = 0 \]

\[ q = \begin{pmatrix} \rho \\ u \\ v \\ p \end{pmatrix} \quad A^{(x)} = \begin{pmatrix} u & \rho & 0 & 0 \\ 0 & u & 0 & 1/\rho \\ 0 & 0 & u & 0 \\ 0 & \gamma p & 0 & u \end{pmatrix} \quad A^{(y)} = \begin{pmatrix} v & 0 & \rho & 0 \\ 0 & v & 0 & 0 \\ 0 & 0 & v & 1/\rho \\ 0 & 0 & \gamma p & v \end{pmatrix} \]
Multidimensions

- Consider flow in the x-direction
  - Eigenvalues of $A^{(x)}$ are: $u-c$, $u$, $u$, $u+c$
    - Middle eigenvalue is degenerate
  - Right eigenvectors are:

$$
\begin{align*}
\rho^{(-)} &= \begin{pmatrix} 1 \\ -c/\rho \\ 0 \\ c^2 \end{pmatrix}, \\
\rho^{(o),1} &= \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \\
\rho^{(o),2} &= \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}, \\
\rho^{(+)} &= \begin{pmatrix} 1 \\ c/\rho \\ 0 \\ c^2 \end{pmatrix}
\end{align*}
$$

- The transverse velocity only jumps over the contact
- In this sense, it is simply advected and either its left or right value is used
Multidimensions

- Interface prediction proceeds as we saw with advection
  - Here's the unsplit version:

\[
U^{n+1/2}_{i+1/2,j,L} = U^n_{i,j} + \frac{\Delta x}{2} \frac{\partial U}{\partial x} + \frac{\Delta t}{2} \frac{\partial U}{\partial t} + \ldots
\]

\[
= U^n_{i,j} + \frac{\Delta x}{2} \frac{\partial U}{\partial x} - \frac{\Delta t}{2} \frac{\partial F^{(x)}}{\partial x} - \frac{\Delta t}{2} \frac{\partial F^{(y)}}{\partial y}
\]

\[
= U^n_{i,j} + \frac{1}{2} \left[ 1 - \frac{\Delta t}{\Delta x} A^{(x)}(U) \right] \Delta U - \frac{\Delta t}{2} \frac{\partial F^{(y)}}{\partial y}
\]

- Conservative update:

\[
U^{n+1}_{i,j} = U^n_{i,j} + \frac{\Delta t}{\Delta x} \left( F^{(x),n+1/2}_{i-1/2,j} - F^{(x),n+1/2}_{i+1/2,j} \right)
\]

\[
+ \frac{\Delta t}{\Delta y} \left( F^{(y),n+1/2}_{i,j-1/2} - F^{(y),n+1/2}_{i,j+1/2} \right)
\]
Multidimensional Hydro

- Unsplit methods generally preserve symmetries on the grid better

This is a standard test problem where there are 4 states (2x2) initially, with the upper left and lower right state identical—the solution should be symmetric about the diagonal. These calculations use 256x256 zones. Aside from the splitting, the algorithm is identical.
Source Terms

Consider gravitational source:

\[ \rho_{i}^{n+1} = \rho_{i}^{n} - \Delta t \frac{(\rho u)_{i+1/2}^{n+1/2} - (\rho u)_{i-1/2}^{n+1/2}}{\Delta x} \]

\[ (\rho u)_{i}^{n+1} = (\rho u)_{i}^{n} - \Delta t \frac{(\rho u^2 + p)_{i+1/2}^{n+1/2} - (\rho u^2 + p)_{i-1/2}^{n+1/2}}{\Delta x} + \frac{\Delta t}{2} (\rho_{i}^{n} + \rho_{i}^{n+1})g \]

\[ (\rho E)_{i}^{n+1} = (\rho E)_{i}^{n} - \Delta t \frac{(\rho u E + pu)_{i+1/2}^{n+1/2} - (\rho u E + pu)_{i-1/2}^{n+1/2}}{\Delta x} + \frac{\Delta t}{2} \left[ (\rho u)_{i}^{n} + (\rho u)_{i}^{n+1} \right] g \]

- This looks like it requires an implicit update, but we update these in turn, and we have the new time-level state as needed.
I have 2 codes online that implement these methods:

- **hydro1d**:
  - Implements the Godunov, PLM, and PPM hydro methods in 1-d
  - Gravity source terms
  - Written in Fortran 95/2003

- **pyro**:
  - Implements 2-d PLM compressible hydro (dimensionally unsplit)
  - Also does 2-d advection, 2-d incompressible, 2-d multigrid, 2-d diffusion
  - Written mostly in python, with some Fortran (through f2py)
Ex: Low Mach Hydro

- How do the ideas we discussed in this course enter into real research?
- What I do: low Mach number hydrodynamics
  - Reformulate the equations of hydrodynamics in the limit that the Mach number is very small
  - Apply this to low speed convective burning in astrophysical environments (SNe Ia, X-ray bursts, stellar evolution, ...)

PHY 604: Computational Methods for Physics and Astrophysics II
Why Low Mach Number Hydro?

- We know that for explicit methods, we are limited by the timestep constraint:

\[ \Delta t = \min \left\{ \frac{\Delta x}{|u| + c} \right\} \]

- Highly subsonic flow, \( M \equiv \frac{|u|}{c} \ll 1 \)

\[ \Delta t \approx \frac{\Delta x}{c} \]

- We want:

\[ \Delta t \approx \frac{\Delta x}{|u|} \]

- Either go implicit or remove sound waves from the system. We do the latter...

▶ A Mach 0.01 front moving to the right (a) initially, (b) after 1 step, (c) after 100 steps.
Low Mach Approximation

- Reformulation of compressible Euler equations
  - Retain compressibility effects due to heating and stratification
  - Asymptotic expansion in Mach number decomposes pressure into thermodynamic and dynamic parts
  - Analytically enforce hydrostatic equilibrium through base state:
    \[ \nabla p_0 = \rho_0 g \]

- Require pressure stay close to hydrostatic pressure:
  \[
  \frac{Dp_0}{Dt} = \left[ \frac{\partial p}{\partial \rho} \right]_s \frac{D\rho}{Dt} + \left[ \frac{\partial p}{\partial s} \right]_\rho \frac{Ds}{Dt} = \frac{\partial p_0}{\partial t} + U \cdot \nabla p_0
  \]
  - with continuity
    \[
    \nabla \cdot U + \frac{1}{\Gamma_1 p_0} U \cdot \nabla p_0 = \frac{1}{\Gamma_1 p_0} \left[ \left[ \frac{\partial p}{\partial s} \right]_\rho \frac{Ds}{Dt} - \frac{\partial p_0}{\partial t} \right]
    \]
Low Mach Approximation

- For gamma-law gas, this can be expressed as:

\[
\nabla \left( p_0^{\frac{1}{\gamma}} U \right) = \frac{p_0^{\frac{1}{\gamma}}}{\gamma p_0} \left( S - \frac{\partial p_0}{\partial t} \right)
\]

- For a general gas:

\[
\nabla \cdot (\beta_0 U) = \beta_0 \left( S - \frac{1}{\Gamma_1 p_0} \frac{\partial p_0}{\partial t} \right)
\]

  - $\beta_0$ is density like variable
  - $S$ represents heating

- Sound waves decoupled from the system
Filtering Sound Waves

- Elliptic constraint: instantaneous acoustic equilibration
We solve:

\[ \frac{\partial \rho X_k}{\partial t} + \nabla \cdot (U \rho X_k) = \rho \dot{\omega}_k \]

\[ \frac{\partial \rho h}{\partial t} + \nabla \cdot (U \rho h) = \frac{Dp_0}{Dt} + \rho H_{\text{nucl}} \]

\[ \frac{\partial U}{\partial t} = -U \cdot \nabla U - \frac{\beta_0}{\rho} \nabla \left( \frac{p'}{\beta_0} \right) - \frac{\rho - \rho_0}{\rho} g e_r \]

\[ \nabla \cdot (\beta_0 U) = \beta_0 \left( S - \frac{1}{\Gamma_1 p_0} \frac{\partial p_0}{\partial t} \right) \]
Solution Technique

- React for $\Delta t/2$
  - Strang-splitting: solve ODEs corresponding to the reaction parts only
- Advect for $\Delta t$
  - Scalar equations are done just like the linear advection equation
  - Velocity equation is done like Burger's equation
- React for final $\Delta t/2$
- Enforce the constraint
  - Velocity field does not satisfy constraint
  - Perform an approximate projection
- Actually we do things slightly more complicated than this, but this gets the gist
Start of Timestep
\( \rho^n, (ph)^n, (px)^n, T^n, U^n \)
\( \rho_0^n, P_0^n, w_0^n \)

Step 1. React for \( \Delta t/2 \)
\( \rho^{n,1}, (ph)^{n,1}, (px)^{n,1}, T^{n,1}, U^n \)
\( \rho_0^{n,1}, P_0^{n,1}, w_0^{n,1} \)

Step 2. Compute \( S^{n,1 sol}, w_0^{n,1} \)
\( \rho^{n,1}, (ph)^{n,1}, (px)^{n,1}, T^{n,1}, U^n \)
\( \rho_0^{n,1}, P_0^{n,1}, w_0^{n,1} \)

Step 3. Construct \( U^{n sol} \)
\( \rho^{n,1}, (ph)^{n,1}, (px)^{n,1}, T^{n,1}, U^n \)
\( \rho_0^{n,1}, P_0^{n,1}, w_0^{n,1} \)

Step 4. Advect for \( \Delta t \)
\( \rho^{n+1,1}, (ph)^{n+1,1}, (px)^{n+1,1}, T^{n+1,1}, U^n \)
\( \rho_0^{n+1,1}, P_0^{n+1,1}, w_0^{n+1,1} \)

Step 5. React for \( \Delta t/2 \)
\( \rho^{n+1,1}, (ph)^{n+1,1}, (px)^{n+1,1}, T^{n+1,1}, U^n \)
\( \rho_0^{n+1,1}, P_0^{n+1,1}, w_0^{n+1,1} \)

Step 6. Compute \( S^{n+1 sol}, w_0^{n+1} \)
\( \rho^{n+1}, (ph)^{n+1}, (px)^{n+1}, T^{n+1}, U^n \)
\( \rho_0^{n+1}, P_0^{n+1}, w_0^{n+1} \)

Step 7. Construct \( U^{n+1 sol} \)
\( \rho^{n+1}, (ph)^{n+1}, (px)^{n+1}, T^{n+1}, U^n \)
\( \rho_0^{n+1}, P_0^{n+1}, w_0^{n+1} \)

Step 8. Advect for \( \Delta t \)
\( \rho^{n+2,1}, (ph)^{n+2,1}, (px)^{n+2,1}, T^{n+2,1}, U^n \)
\( \rho_0^{n+2,1}, P_0^{n+2,1}, w_0^{n+2,1} \)

Step 9. React for \( \Delta t/2 \)
\( \rho^{n+2,1}, (ph)^{n+2,1}, (px)^{n+2,1}, T^{n+2,1}, U^n \)
\( \rho_0^{n+2,1}, P_0^{n+2,1}, w_0^{n+2,1} \)

Step 10. Define \( S^{n+1} \)
\( \rho^{n+1}, (ph)^{n+1}, (px)^{n+1}, T^{n+1}, U^n \)
\( \rho_0^{n+1}, P_0^{n+1}, w_0^{n+1} \)

Step 11. Update \( U \)
\( \rho^{n+1}, (ph)^{n+1}, (px)^{n+1}, T^{n+1}, U^{n+1} \)
\( \rho_0^{n+1}, P_0^{n+1}, w_0^{n+1} \)

Step 12. Compute new \( \Delta t \)
\( \rho^{n+1}, (ph)^{n+1}, (px)^{n+1}, T^{n+1}, U^{n+1} \)
\( \rho_0^{n+1}, P_0^{n+1}, w_0^{n+1} \)