

Exercises

Numerical Models

The conservation laws

The Einstein Field Equations

$$G_{ab} = \kappa T_{ab}$$

automatically give us conservation of stress energy,

$$\nabla_a T^a_b = 0.$$

Unfortunately this abstract form is not suitable for numerical evolution. We want to phrase everything as partial differential equations.

1. Introduce an *tetrad*: four vectors $e^a_{(m)}$ where (m) is a label. The spacetime coordinate metric g_{ab} is linked to the tetrad through the tetrad reference metric $\hat{\gamma}_{(mn)}$ as

$$g_{ab} = e^a_{(m)} e^b_{(n)} \hat{\gamma}_{(mn)}$$

or

$$g^{ab} = e^a_{(m)} e^b_{(n)} \hat{\gamma}^{(mn)}.$$

If the tetrad is *orthonormal* then $\hat{\gamma}$ is the identity matrix. Construct the orthonormal tetrad for Minkowski spacetime in Cartesian and spherical coordinates.

2. One route to converting the covariant derivative to partial derivatives suitable for numerical implementation is the identity

$$\nabla_a V^a = \frac{1}{\sqrt{-g}} \partial_a (\sqrt{-g} V^a),$$

where $\sqrt{-g}$ is the square root of the determinant of the metric. By contracting T^a_b with an arbitrary vector W^b , show that

$$\begin{aligned} \nabla_a T^a_b &= 0 \\ \Rightarrow \partial_a (\sqrt{-g} T^a_b W^b) &= \sqrt{-g} T^a_b \nabla_a W^b. \end{aligned}$$

3. Choose coordinates aligned with the *orthonormal* tetrad, so that $e_{(m)} = \{\partial_t, \partial_j\}$ and

$$\nabla_a e^b_{(m)} = \partial_a e^b_{(m)} + \Gamma^b_{ac} e^c_{(m)} = \Gamma^b_{ac} e^c_{(m)},$$

to show that stress energy conservation implies

$$\partial_a (\sqrt{-g} T_b^a e_{(m)}^b) = \sqrt{-g} T_b^a \Gamma_{ac}^b e_{(m)}^c,$$

or in the looser coordinate notation

$$\begin{aligned}\partial_a (\sqrt{-g} T_t^a) &= \sqrt{-g} T_b^a \Gamma_{at}^b, \\ \partial_a (\sqrt{-g} T_j^a) &= \sqrt{-g} T_b^a \Gamma_{aj}^b.\end{aligned}$$

4. Show that, for Minkowski spacetime in standard Cartesian coordinates, the four equations of motion can be written in *conservation form*

$$\partial_t \mathbf{q} + \partial_k \mathbf{f}^{(k)}(\mathbf{q}) = \mathbf{0}.$$

This is the conservation of spatial 3-momentum (the j equations) and energy (the t equation). The \mathbf{q} variables are the *conserved* variables, and the $\mathbf{f}^{(k)}$ terms are the *fluxes*

5. Show that, for Minkowski spacetime in standard *spherical* coordinates, the equations of motion are instead in *balance law form*

$$\partial_t \mathbf{q} + \partial_k \mathbf{f}^{(k)}(\mathbf{q}) = \mathbf{s}(\mathbf{q}),$$

where the *sources* \mathbf{s} contain no derivatives of the matter variables. This shows that the source terms need not just be due to gravity: geometric effects can induce source terms by themselves.

6. The original balance law form for relativistic fluids is called the *Valencia* form: see the [review by Font](#) for more details. Working with coordinates *not* aligned with tetrads can be useful, when working with spherical coordinates (see work by Montero, Cordeiro-Carillon, Baumgarte and Müller in [general](#) and [specifically fluid](#) cases) or with multiple patches (see work Pollney, Reisswig, Schnetter and others in [vacuum and theory](#) and [fluid](#) cases).

The perfect fluid

The simplest model for neutron star matter is as a perfect fluid, where

$$T_{ab} = \rho_0 h u_a u_b + p g_{ab}.$$

Here ρ_0 is the proper rest mass density, u^a the four velocity of the fluid, h the specific enthalpy and p the isotropic pressure. Thermodynamic considerations will link the density, enthalpy and pressure. Write

$$h = 1 + \epsilon + \frac{p}{\rho_0},$$

where ϵ is the specific internal energy density. Then we assume (in the simplest case) that the system is closed by an *equation of state* linking the pressure to two of the other thermodynamic variables. For now, assume $p \equiv p(\rho_0, \epsilon)$.

1. Check that there are *five* independent degrees of freedom.
2. In the $3 + 1$ split choose n^a to be the unit normal to the spatial slices, $n^a = \alpha^{-1}(1, -\beta^i)$ where α, β^i are the lapse and shift, and γ_{ij} to be the spatial metric. Split the four velocity into a piece along n^a and a piece orthogonal to it,

$$u_a = W(n_a + v_a), \quad n^a v_a = 0.$$

Show that the *Lorentz factor* W is

$$W = -u^a n_a,$$

and show that

$$\begin{aligned} u^a &= \frac{W}{\alpha} (1, \alpha v^i - \beta^i) \\ u_a &= W (-\alpha + v_j \beta^j, v_i). \end{aligned}$$

Check that

$$W = \frac{1}{(1 - \gamma_{ij} v^i v^j)^{1/2}}.$$

3. Choose the tetrad and coordinates so that $\mathbf{e} = \{\mathbf{n}, \partial_j\}$. This means

$$e_{(0)}^a = -\alpha g^{0a} = n^a, \quad e_{(j)}^a = \delta_j^a.$$

Using that the determinant of the spatial 3-metric is related to the determinant of the full 4-metric through $\sqrt{-g} = \alpha\sqrt{\gamma}$, show that the conserved variables are

$$\begin{aligned} t : \quad \sqrt{-g} T_b^t e_{(0)}^b &= \alpha^{-1} \sqrt{-g} (\rho h W^2 - p) \\ &= \sqrt{\gamma} E, \\ j : \quad \sqrt{-g} T_b^t e_{(j)}^b &= \alpha^{-1} \sqrt{-g} \rho h W^2 v_j \\ &= \sqrt{\gamma} S_j, \end{aligned}$$

and the fluxes in the i direction are

$$\begin{aligned} t : \quad \sqrt{-g} T_b^i e_{(0)}^b &= \sqrt{-g} \left[S_j \left(v^i - \frac{\beta^i}{\alpha} \right) + p \delta_j^i \right], \\ j : \quad \sqrt{-g} T_b^i e_{(j)}^b &= \sqrt{-g} \left[E \left(v^i - \frac{\beta^i}{\alpha} \right) + p v^i \right]. \end{aligned}$$

4. We now have four equations of motion for the five degrees of freedom. We enforce conservation of mass through

$$\nabla_a (\rho_0 u^a) = 0.$$

Show that this can be written in coordinate form as

$$\partial_a (\sqrt{\gamma} D) + \partial_i \left(\sqrt{-g} D \left(v^i - \frac{\beta^i}{\alpha} \right) \right) = 0,$$

where $D = \rho_0 W$.

5. Restrict to Cartesian coordinates in Minkowski spacetime with the standard gauge. Check you recover the equations of motion for a special relativistic fluid

$$\partial_t \begin{pmatrix} D \\ S_j \\ E \end{pmatrix} + \partial_i \begin{pmatrix} D v^i \\ S_j v^i + p \delta_j^i \\ (E + p) v^i \end{pmatrix} = \mathbf{0}.$$

Take the slow motion approximation ($v \ll 1, \epsilon + p/\rho_0 \ll 1$) to show that the Newtonian Euler equations

$$\partial_t \begin{pmatrix} \rho_0 \\ \rho_0 v_j \\ \rho_0 \left(\epsilon + \frac{1}{2} v_k v^k \right) \end{pmatrix} + \partial_i \begin{pmatrix} \rho_0 v^i \\ \rho_0 v_j v^i + p \delta_j^i \\ \left(\rho_0 \left(\epsilon + \frac{1}{2} v_k v^k \right) + p \right) v^i \end{pmatrix} = \mathbf{0}$$

are recovered at leading order for the momentum equations, but only at second order for the energy equation.

6. Go back to the (SR or GR) equations and define $\tau = E - D$. Replace the energy equation with the appropriate equation for τ , take the Newtonian limit, and show the correct equation appears at leading order. This can be important in simulations: near equilibrium the continuity and energy equations can degenerate without this modification.

Action principles

Lagrangian and Hamiltonian methods are incredibly powerful, but can be tricky to apply to certain models.

To use them for relativistic fluids, we define the number flux 4-vector n^a . Its norm n is found from $n^2 = -n^a n_a$ is related to the specific rest mass density ρ_0 using the mass per particle m_b , as $\rho_0 = m_b n$. The full number flux $n^a = n u^a$, where u^a is the standard 4-velocity.

We then look at the energy Λ which will be a function only of the number density, $\Lambda(n)$. This will play the role of our Lagrangian. The *conjugate momentum*

$$\mu_a = \frac{\partial \Lambda}{\partial n^a}$$

is linked to the specific enthalpy, as $\mu = m_b h$.

1. Start by assuming the action principle is given by

$$S = \int \sqrt{-g} \Lambda(n).$$

Assuming the spacetime is fixed, use the standard calculus of variations approach to show

$$\mu_a \equiv 0.$$

2. The previous result is clearly not what we want. By allowing the small variations to be unconstrained, we have allowed too much freedom. We know that the number flux must be conserved, $\nabla_a n^a = 0$. By adding this constraint as a Lagrange multiplier,

$$S = \int \sqrt{-g} [\Lambda(n) + \lambda \nabla_a n^a],$$

show that this implies

$$\omega_{ab} = \nabla_{[a} \mu_{b]} = 0.$$

The quantity ω_{ab} is the *vorticity* and this is showing the fluid is irrotational.

3. Again, this is stricter than we want, because we have still allowed too much freedom. It turns out that we want to impose that the variations act to perturb the flow lines normal to themselves. This imposes that

$$\delta n^a = -\mathcal{L}_\xi n^a - n^a \nabla_b \xi^b,$$

where the Lagrangian displacements ξ^a can be varied freely. Show that this implies

$$f_b = n^a \omega_{ab} = 0.$$

4. This equation is equivalent to the Euler equation found from stress energy conservation, provided the pressure satisfies

$$\delta p = -n^a \delta \mu_a,$$

so that

$$\nabla_b p = -n^a \nabla_b \mu_a.$$

Check that this holds true for differentiable solutions.

5. We can now extend this approach to multiple, distinguishable fluid species. This is needed for electrodynamics or superfluids. Assume we have two species with number fluxes n_x^a , where $x \in \{1, 2\}$ is the species label. Assume the energy depends on both the n_x^2 and $n_{12}^2 = -g_{ab} n_1^a n_2^b$. Show that the equations of motion are still

$$f_b^x = n_x^a \nabla_{[a} \mu_{b]}^x = 0.$$

6. Check that the conjugate momentum for a species is only parallel to its number flux (in general) if

$$\frac{\partial^2 \Lambda}{\partial n_1^a \partial n_2^b} = 0.$$

This is the *entrainment* effect. The second species “drags” the first along with it via their interaction in the energy.

7. For more detail on multifluids see the [Living Review of Andersson and Comer](#). The impact of the entrainment on numerical simulations can be messy. See, for example, the argument in [Andersson et al](#) that entrained systems cannot be written in balance law form.

Numerical Theory

Finite differencing

Remember our favourite tool for numerical methods is the Taylor expansion, which can be phrased either as an expansion as

$$f(x_0 + \Delta x) = f(x_0) + \sum_{n=1} \frac{(\Delta x)^n}{n!} \left. \frac{d^n f}{dx^n} \right|_{x=x_0},$$

or using the remainder term as

$$f(x_0 + \Delta x) = f(x_0) + \sum_{n=1}^k \frac{(\Delta x)^n}{n!} \left. \frac{d^n f}{dx^n} \right|_{x=x_0} + \frac{(\Delta x)^{k+1}}{(k+1)!} \left. \frac{d^{k+1} f}{dx^{k+1}} \right|_{x=\xi},$$

where $\xi \in [x_0, x_0 + \Delta x]$.

1. Check that the *forward differencing* formula,

$$\left. \frac{df}{dx} \right|_{x=x_0} \rightarrow \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

is *first order accurate*. That is, show that

$$\left. \frac{df}{dx} \right|_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} + \mathcal{O}((\Delta x)^1).$$

2. Check that the *second order central differencing* formula,

$$\left. \frac{df}{dx} \right|_{x=x_0} \rightarrow \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x}$$

is *second order accurate*. That is, show that

$$\left. \frac{df}{dx} \right|_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} + \mathcal{O}((\Delta x)^2).$$

3. Take the differential equation

$$\frac{dy}{dx} = F(x, y(x)).$$

Using the forward differencing formula, derive Euler's method for solving the differential equation,

$$y(x_0 + \Delta x) = y(x_0) + \Delta x F(x_0, y(x_0)).$$

Show that the remainder term is $\mathcal{O}((\Delta x)^2)$. This means the scheme will be *first order* after taking $N \propto (\Delta x)^{-1}$ steps.

4. Derive Euler's method by integrating the original ODE, by using a suitable approximation of the integral of $F(x, y(x))$.
5. Take the advection equation

$$\partial_t q - v \partial_x q = 0,$$

where v is a constant (the speed of propagation). Using the forward differencing formula, derive the scheme

$$q(x_0, t_0 + \Delta t) = q(x_0, t_0) + \frac{v \Delta t}{\Delta x} (q(x_0 + \Delta x, t_0) - q(x_0, t_0)).$$

Check that it is first order.

6. The scheme for the advection equation is *stable* (i.e., doesn't blow up) **only if** the CFL number $\frac{v \Delta t}{\Delta x} < 1$. Draw a spacetime diagram of the points involved in the scheme. Link the CFL condition to the relativistic idea of the null cone, or domains of dependence, by remembering that v is the speed of propagation.

Modified equation

The advection equation

$$\partial_t q + v \partial_x q = 0$$

can be modelled using the simple first order upwind scheme

$$q(x_0, t_0 + \Delta t) = q(x_0, t_0) + \frac{v \Delta t}{\Delta x} (q(x_0 - \Delta x, t_0) - q(x_0, t_0)),$$

subject to the CFL condition.

1. Apply Taylor's theorem to *the upwind scheme*, to show that the scheme obeys the *modified equation*

$$\partial_t q + v \partial_x q = \frac{v \Delta x}{2} \left(1 - \frac{v \Delta t}{\Delta x}\right) \partial_{xx} q.$$

Convince yourself that the new term $\beta \partial_{xx} q$ where $\beta = \frac{v \Delta x}{2} \left(1 - \frac{v \Delta t}{\Delta x}\right)$, the *numerical viscosity*, behaves as you would expect with Δx .

2. Using your knowledge of the stability of the *heat* equation, check what happens to the modified equation when the CFL condition is violated.
3. Check that $q(\eta) = f(x - vt)$ is a solution of the original advection equation, for any differentiable function f .
4. Check that $q(\xi) = f(x/t)$ is a solution of the original advection equation along lines where $\xi = v$. These are *similarity solutions*.
5. Motivated by the solution for the advection equation, look for *travelling wave* solutions of the *modified* equation. That is, assume the solution has the form $q(\eta) = f([x - vt]/t^\alpha)$ and show that

$$\frac{\beta}{t^{2\alpha}} q'' + \frac{\alpha \eta}{t} q' = 0.$$

By choosing $\alpha = 1/2$, show that the solution can be written in terms of error functions.

6. Show that if the initial data is $q(x, 0) = 2$ for $x < 0$ and $q(x, 0) = 0$ for $x > 0$ then the modified equation is solved exactly by

$$q_{\text{modified}}(x, t) = \text{erfc} \left(\frac{x - vt}{\sqrt{4\beta t}} \right),$$

where the *complementary error function* is given by

$$\text{erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^\infty dz \exp(-z^2).$$

7. From the solutions to the original advection equation, we can write the solution of the original problem as

$$q_{\text{original}}(x, t) = 2H(vt - x).$$

Show that

$$\begin{aligned} \|q_{\text{modified}}(x, T) - q_{\text{original}}(x, t)\|_1 &= 2 \int_0^\infty ds \text{erfc} \left(\frac{s}{\sqrt{4\beta T}} \right) \\ &= C_1 \sqrt{\beta T} \\ &\simeq C_2 \sqrt{\Delta x T}, \end{aligned}$$

for small Δx , assuming $\Delta x / \Delta t$ is fixed. This shows that the *first order* method has worse convergence behaviour near shocks, if measured in the 1-norm.

Phase errors and neutron stars

We want to estimate what amount of numerical effort is required to evolve a neutron star.

Thinking of a neutron star as a fluid, obeying Euler's equations (coupled to gravity), it evolves through waves moving back and forth. In *vacuum* numerical relativity LIGO requires a model to have a certain accuracy by fixing the allowable *phase error*: how much can the phase of a gravitational wave be different between the model (or simulation) and the "real thing". We can do a similar analysis for a neutron star.

Massively simplify the problem to the advection equation

$$\partial_t q + v \partial_x q = 0$$

on the domain $x \in [0, 2\pi]$ with periodic boundaries, $q(x+2\pi, t) = q(x, t)$. Introduce the grid $x_j = j \Delta x$ with $\Delta x = (2\pi)/(N+1)$. Set the initial data to a single Fourier mode,

$$q(x, 0) = \exp(i\ell x)$$

for some integer ℓ .

1. Check that the exact solution is

$$q_e(x, t) = \exp[i\ell(x - vt)].$$

2. Assume that the solution produced by a finite difference method has the form

$$q_{m,\Delta x}(x, t) = \exp[i\ell(x - v_m(\ell)t)].$$

Here m denotes the method. Show that when using second order central differencing ($m \rightarrow 2cd$), where

$$\partial_x q|_{x=x_k} \rightarrow \frac{q(x_k + \Delta x, t) - q(x_k - \Delta x, t)}{2\Delta x},$$

that

$$v_{2cd}(\ell) = v \frac{\sin(\ell \Delta x)}{\ell \Delta x}.$$

3. The *phase error* can be written as the relative difference

$$e_m(\ell) = \left| \frac{q_e(x, t) - q_{m,\Delta x}(x, t)}{q_e(x, t)} \right|.$$

Show that, at a fixed time $t = T$, the phase error is roughly

$$e_m(\ell) \simeq |\ell(v - v_m(\ell))T|.$$

4. We expect to care most about the small ℓ modes, so $\ell\Delta x$ will likely be small. Expand out $v_{2\text{cd}}(\ell)$ in powers of $\ell\Delta x$ to show that

$$e_{2\text{cd}}(\ell) \simeq v\ell \frac{(\ell\Delta x)^2}{6} T.$$

5. To get the explicit dimensions out of the problem, introduce the *points per wavelength*

$$p = \frac{2\pi}{\ell\Delta x}$$

and *number of evolution periods*

$$\nu = \frac{\ell v T}{2\pi}.$$

Show that the phase error is

$$e_{2\text{cd}}(\ell) \simeq \frac{\pi\nu}{3} \left(\frac{2\pi}{p} \right)^2.$$

Re-arrange to get the points per wavelength required for given phase error,

$$p_{2\text{cd}} \gtrsim 2\pi \sqrt{\frac{\nu\pi}{3e_{2\text{cd}}}}.$$

6. To estimate the number of evolution periods required, we need the *fundamental mode frequency* of a neutron star. Check the discussion in the [Living Review of Kokkotas and Schmidt](#) to find this is around 3kHz. Assuming we simulate around 10ms before merger, show that $\nu \simeq 30$.
7. Assuming a phase error of 1%, show that $p_{2\text{cd}} \simeq 350$.
8. The fundamental mode has a wavelength roughly the diameter of the neutron star. Assuming this is $\sim 25\text{km}$, show that we need $\Delta x \simeq 70\text{m}$.
9. Relax the error restrictions to 10%. Show this needs to $\Delta x \simeq 230\text{m}$.
10. Show that when using *fourth* order central differencing ($m \rightarrow 4\text{cd}$), where

$$\partial_x q|_{x=x_k} \rightarrow \frac{-q(x_k + 2\Delta x, t) + 8q(x_k + \Delta x, t) - 8q(x_k - \Delta x, t) + q(x_k - 2\Delta x, t)}{12\Delta x},$$

that

$$v_{4\text{cd}}(\ell) = v \frac{8 \sin(\ell\Delta x) - \sin(2\ell\Delta x)}{6\ell\Delta x},$$

and hence

$$e_{4cd}(\ell) \simeq \frac{\pi\nu}{15} \left(\frac{2\pi}{p} \right)^4$$

implying

$$p_{4cd} \gtrsim 2\pi \sqrt[4]{\frac{\nu\pi}{15e_{2cd}}}.$$

Show that for evolving a neutron star through inspiral with a phase error of 1% we need $\Delta x \sim 800\text{m}$.

11. **Going further:** Estimate the required grid spacing, and hence practicality, of numerically simulating the $p - g$ instability using a nonlinear code (see the [paper by Weinberg, Arras and Burkart](#) as a starting point).

Vacuum part 1

Some of the biggest problems for nonlinear simulations come in the low density regions: either near the surface of the neutron star, or in the artificial atmosphere that surrounds it. This exercise uses the Newtonian Euler equations for simplicity: you can repeat it with the relativistic equations, but the conclusions are similar.

Rarefaction waves are continuous self-similar solutions to conservation laws. Take the Euler equations in one spatial dimension. The primitive variables are $\mathbf{u} = \rho, v, e$: the density, velocity and internal energy. The conserved variables are

$$\mathbf{q} = (\rho, S, E)^T = \left(\rho, \rho v, \rho \left(e + \frac{1}{2} v^2 \right) \right)^T,$$

the density, momentum and energy. The fluxes are

$$\mathbf{f}(\mathbf{q}) = \begin{pmatrix} S \\ Sv + p \\ (E + p)v \end{pmatrix},$$

where the pressure p is determined (in this simple case) from the density and internal energy, and will be fixed to the γ -law equation of state $p = (\gamma - 1)\rho e$

1. Show that the conservation law

$$\partial_t \mathbf{q} + \partial_x \mathbf{f} = \mathbf{0}$$

has a continuous self-similar solution if $\xi = x/t$ is an eigenvalue of the Jacobian matrix,

$$J = \frac{\partial \mathbf{f}}{\partial \mathbf{q}}.$$

You should derive the relation

$$(J - \lambda \text{Id}) \partial_\xi \mathbf{q} = \mathbf{0}.$$

2. Show that the eigenvalues of J for the Newtonian Euler equations are v and $v \pm c_s$, where the *speed of sound* is

$$c_s^2 = \frac{\gamma p}{\rho}.$$

3. The eigenstructure of the Jacobian gives us equations for how the quantities vary along a rarefaction. In particular, let (p) denote the wave number (associated with the p^{th} eigenvalue). From the relation derived in part 1, show that

$$\partial_\xi \mathbf{q} = \alpha \mathbf{r}^{(p)}$$

where α is a normalization constant and $\mathbf{r}^{(p)}$ a right eigenvector of J . By differentiating

$$\lambda^{(p)}(\mathbf{q}) = \xi$$

with respect to ξ , show that

$$\partial_\xi \mathbf{q} = \frac{\mathbf{r}^{(p)}}{\mathbf{r}^{(p)} \cdot \partial_{\mathbf{q}} \lambda^{(p)}}.$$

4. Show that this result can be generalized as follows. Let \mathbf{U} be any set of variables such that $\partial_{\mathbf{q}} \mathbf{U}$ is invertible, and $s(\xi)$ be any monotonic parameter along the characteristics. Then

$$\partial_s \mathbf{U} = \frac{\mathbf{r}^{(p)}}{\mathbf{r}^{(p)} \cdot \partial_{\mathbf{U}} \lambda^{(p)}}.$$

Here the eigenvalues and eigenvectors are associated with the Jacobian $\partial_{\mathbf{U}} \mathbf{f}$. Show that, for the Euler equations,

$$\begin{aligned} \frac{dv}{dp} &= \mp \frac{1}{\rho c_s} \\ \frac{d\rho}{dp} &= \frac{1}{c_s^2} \end{aligned}$$

along the rarefactions associated with the *acoustic* waves $\lambda = v \pm c_s$.

5. Along a rarefaction wave the entropy is constant (as the solution is continuous) so, for the γ law equation of state we can write $p = K\rho^\gamma$ giving $c_s^2 = K\gamma\rho^{\gamma-1}$. Assume that we know the values of a state $\mathbf{u}_{\text{known}} = (\rho_{\text{known}}, v_{\text{known}}, p_{\text{known}})$. Assume that this state is connected to another state with pressure p_* by a rarefaction. Show that

$$\rho_* = \rho_{\text{known}} \left(\frac{p_*}{p_{\text{known}}} \right)^{1/\gamma},$$

$$v_* = v_{\text{known}} \mp \frac{2c_{s,\text{known}}}{\gamma-1} \left[1 - \left(\frac{p_*}{p_{\text{known}}} \right)^{(\gamma-1)/2\gamma} \right].$$

6. Take two known states with the same density and pressure, and velocities of the same magnitude but different sign, so that the states are separating:

$$\begin{aligned}\rho_L &= \rho_R = \rho_K, \\ p_L &= p_R = p_K, \\ -v_L &= v_R = V.\end{aligned}$$

Show that the $v_* = 0$, and that the state connecting them will be *vacuum* ($p_* = 0$) if

$$V > \frac{2c_{s,\text{known}}}{\gamma-1}.$$

7. Show that when we are near the surface of the star, or in the artificial atmosphere, it becomes more likely that the *correct solution* is the generation of vacuum.

Vacuum part 2

High accuracy numerical methods often depend explicitly on the characteristic information: the eigenvalues and eigenvectors of the Jacobian matrix. This can cause additional problems for neutron stars.

1. Start from the conservation law

$$\partial_t \mathbf{q} + \partial_x \mathbf{f}(\mathbf{q}) = \mathbf{0}.$$

Show that, as long as the solution remains differentiable, this can be written as

$$\partial_t \mathbf{q} + J \partial_x \mathbf{q} = \mathbf{0}.$$

Hence show that, as long as the system is *hyperbolic*, so that the Jacobian J is diagonalizable, the system can be *locally approximated* by

$$\partial_t \mathbf{w} + \Lambda \partial_x \mathbf{w} = 0,$$

where Λ is a diagonal matrix with entries corresponding to the eigenvalues of J . \mathbf{w} are the *characteristic* variables. As each (roughly!) obeys an uncoupled advection equation it's easier to apply high order methods to these variables.

2. The characteristic form is useful as the equations (locally) decouple. Many high-order accuracy methods rely on it, explicitly or implicitly. Converting between conserved (or primitive) variables and characteristic requires the eigenvectors of the Jacobian. Show that the Jacobian for the Newtonian Euler equations is

$$\frac{\partial \mathbf{f}}{\partial \mathbf{q}} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{v^2(\gamma-3)}{2} & v(3-\gamma) & \gamma-1 \\ \frac{v(-2c_s^2 + \gamma^2 v^2 - 3\gamma v^2 + 2v^2)}{2(\gamma-1)} & \frac{c_s^2 - \gamma^2 v^2 + \frac{5\gamma v^2}{2} - \frac{3v^2}{2}}{\gamma-1} & \gamma \end{pmatrix}.$$

You may find it easiest to use computer algebra, and to compute $\partial_{\mathbf{u}} \mathbf{f}$ and $\partial_{\mathbf{u}} \mathbf{q}$, where $\mathbf{u} = \rho, v, e$ are the primitive variables, then eliminating the internal energy in favour of the speed of sound.

3. Show that the eigenvectors associated with the eigenvalues $\lambda_{\pm} = v \pm c_s$ can be written

$$\mathbf{r}_{\pm} = \begin{pmatrix} 1 \\ v \pm c_s \\ \frac{v^2(3-\gamma)}{2(\gamma-1)} + \frac{v \pm c_s}{\gamma-1} (v - v(3-\gamma) \pm c_s) \end{pmatrix}.$$

4. Note that as we approach the surface of the neutron star $\rho, p \rightarrow 0$ and particularly $c_s \rightarrow 0$. Show that the eigenvectors degenerate and so the conversion to characteristic variables breaks down.
5. Consider whether changing the evolution variables would avoid this problem (whilst potentially introducing others). For example, introduce the “entropy” variable $K = p/\rho^\gamma$, and look at (K, v, p) .

Well balancing

During inspiral, each individual neutron star almost perfectly balances its internal hydrodynamic forces (e.g., the pressure gradients) against the spacetime curvature. Numerically, this balance is more difficult to achieve. Numerical schemes that perfectly preserve equilibrium solutions are called *well balanced*.

1. Show that the advection equation with source,

$$\partial_t q + \partial_x q = q$$

has time-independent equilibrium solutions $q = Ce^x$.

2. Extend the upwind scheme from above to show that it gives

$$q(x_0, t_0 + \Delta t) = q(x_0, t_0) - \frac{\Delta t}{\Delta x} (q(x_0, t_0) - q(x_0 - \Delta x, t_0)) + \Delta t q(x_0, t_0).$$

3. Show that this scheme does not preserve the equilibrium, but that the error after one step is $\mathcal{O}(\Delta t \Delta x)$.
4. Show that the scheme

$$q(x_0, t_0 + \Delta t) = q(x_0, t_0) - \frac{\Delta t}{\Delta x} (q(x_0, t_0) - e^{\Delta x} q(x_0 - \Delta x, t_0))$$

does perfectly preserve the equilibrium.

5. Rewrite this well-balanced scheme as

$$q(x_0, t_0 + \Delta t) = q(x_0, t_0) - \frac{\Delta t}{\Delta x} (q(x_0, t_0) - q(x_0 - \Delta x, t_0)) + \frac{\Delta t}{\Delta x} (e^{\Delta x} - 1) q(x_0 - \Delta x, t_0).$$

Check that the final source term is a first order approximation to the source term $\Delta t q(x_0, t_0)$ used in the original scheme.

6. This is hard to apply in general; see [Parés](#) for a starting point and further references.

Shocks

When the neutron stars merge, a shock sweeps through the matter. The numerical model and method needs to be able to deal with this discontinuity.

1. Take the scalar conservation law

$$\partial_t q + \partial_x f(q) = 0.$$

Assume that the solution is *piecewise constant*, so

$$q = \begin{cases} q_L & x < s(t) \\ q_R & x > s(t) \end{cases},$$

where $s(t)$ is the shock location and $\dot{s} = V_s$ is the speed of the shock. By integrating over a small region of spacetime containing the shock, derive the *Rankine-Hugoniot* conditions

$$V_s[q] = [f],$$

where $[\cdot] = (\cdot)_R - (\cdot)_L$.

2. Show that, for any real number $n > 0$, the differentiable solutions of

$$\partial_t (q^n) + \frac{n}{n+1} \partial_x (q^{n+1}) = 0$$

are solutions of Burgers equation

$$\partial_t q + q \partial_x q = 0,$$

and hence equivalent.

3. Using the Rankine-Hugoniot conditions, show that the shock speeds for shocks solving the n -dependent conservation law *do* depend on n . Assuming $q_L = 2$ and $q_R = 1$, plot how this varies.
4. The usual punchline here is to say that the non-conservative form (such as the form for Burgers equation above) is not suitable for evolving solutions with shocks, as it may be equivalent to an infinite number of conservation law forms each with different shock speeds. The full answer is more complex. The non-conservative form *can* be used, provided a prescription (usually via an explicit limiting viscosity) for the shock speed is given: see, for example, work by [Parés](#).

Telescoping

Our earlier derivations of simple numerical schemes used Taylor expansions, which assume the solution is differentiable. This will fail at shocks. We need to know why, before we know how to do better. We also need to ensure that shocks propagate at the right speed.

From now on we will introduce a grid in spacetime. The spatial and time locations will be

$$\begin{aligned} x_i &= x_L + \left(i + \frac{1}{2}\right) \Delta x, & i &= 0, 1, \dots, N-1, & \Delta x &= \frac{x_R - x_L}{N}, \\ t^n &= n \Delta t, & n &= 0, 1, \dots \end{aligned}$$

The solution q at a specific spacetime point will be denoted

$$q_i^n = q(x_i, t^n).$$

Any other functions will use similar notation. If the grid marks the center of *cells*, or *volumes*, in finite volume style, then the cell boundaries to the right of the cell centers will be at $x_i + \Delta x / 2 = x_{i+1/2}$.

With this setup, $x_{-1/2} = x_L$ is the left edge of the domain (and the left cell boundary of the first cell), and $x_{N+1/2} = x_R$ is the right edge of the domain (and the right cell boundary of the last cell).

1. Extend the upwind scheme used above to the scalar conservation law

$$\partial_t q + \partial_x f(q) = 0,$$

to show the scheme is

$$q_i^{n+1} = q_i^n + \frac{\Delta t}{\Delta x} (f_{i-1}^n - f_i^n).$$

2. Integrate the conservation law over the cell $[x_{i-1/2}, x_{i+1/2}] \times [t^n, t^{n+1}]$. Denote the integral average of q as \hat{q} , so that

$$\hat{q}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} dx q(x, t^n).$$

Show that this results in the scheme

$$\hat{q}_i^{n+1} = \hat{q}_i^n + \frac{\Delta t}{\Delta x} (f_{i-1/2}^n - f_{i+1/2}^n).$$

3. Compare the two schemes, noting that they match if the flux at the cell boundary matches the flux at the cell centre to the left of that boundary. Using Taylor expansions, check that

$$q_i^n = \hat{q}_i^n + \mathcal{O}((\Delta x)^2).$$

4. Integrate the conservation law over $[x_L, x_R]$. Sum the integral average scheme over $i = 0, \dots, N-1$. Show that, in both cases, the only contributions are from fluxes through the boundaries of the domain. This *telescoping* property of the numerical scheme means that it should obey the Rankine-Hugoniot conditions, and so shock speeds should be correct.

Monotonicity

Whilst getting the shock speed right is important, so is getting the shock amplitude. Numerical schemes often introduce oscillations near shocks (thanks to *Gibb's oscillations*). We would like a scheme to be *monotone*: if q_j^{n+1} depends on (say) $q_{j-1,j,j+1}^n$ then we want

$$\min_{k \in S_j} q_k^n \leq q_j^{n+1} \leq \max_{k \in S_j} q_k^n,$$

where S_j is the *stencil* $((j-1, j, j+1))$ of points on which q_j^{n+1} depends.

We write our update scheme as

$$\hat{q}_i^{n+1} = G(\hat{q}^n) = G(\hat{q}_{j-1}^n, \hat{q}_j^n, \hat{q}_{j+1}^n).$$

The scheme is *consistent* if

$$G(Q, Q, Q) = Q.$$

The scheme is *monotone* if G is monotonically non-decreasing in all of its arguments. Symbolically this is written $G(\uparrow, \uparrow, \uparrow)$.

1. Assume that the cell boundary flux $f_{i-1/2}$ is a function of the solution in the neighbouring cells,

$$f_{i-1/2} = F(\hat{q}_{i-1}, \hat{q}_{i+1}).$$

Show that the update scheme is monotone if F is increasing in its first argument and decreasing in its second, $F(\uparrow, \downarrow)$.

2. Show that, if the scheme is monotone and two sets of data U_i^n, V_i^n are such that $U_i^n \leq V_i^n$, then

$$G(U_{j-1}^n, U_j^n, U_{j+1}^n) \leq G(V_{j-1}^n, V_j^n, V_{j+1}^n).$$

3. Consider V such that it matches U *except* within the stencil S_j , where it matches the maximum of U :

$$V_i^n = \begin{cases} \max_{k \in S_j} U_k^n & \text{if } i \in S_j \\ U_i^n & \text{otherwise} \end{cases}.$$

Show that

$$G(U_{j-1}^n, U_j^n, U_{j+1}^n) \leq G(V_{j-1}^n, V_j^n, V_{j+1}^n) \leq \max_{k \in S_j} U_k^n.$$

Using a similar method for the minimum, show the scheme is monotone.

4. Find a suitable reference for *Godunov's theorem*, which shows that a *linear* monotone scheme is at most first order accurate.
5. The *linear* in Godunov's theorem means that the stencil (the points that it contains and the weights associated with each point) cannot depend on the data. The crucial step in developing useful codes for evolving matter is to get around this: the schemes will depend on the data.

Stiffness

Most of the discussion above has been about the flux terms, linked to the hyperbolic partial derivatives in space. However, there can be numerical problems from the sources terms as well.

A specific example arises in electromagnetism where the electric field is sourced by the charge current. In models including *resistivity*, the charge current has a term proportional to η^{-1} , where η quantifies the amount of resistance to charge flowing. In the ideal MHD limit (which applies to *most* of the problems we're interested in over *most*, but *not all* of the spacetime) $\eta \rightarrow 0$.

As a toy model consider

$$\frac{dq}{dt} = -\frac{1}{\eta}q,$$

where η is small and positive.

1. Write down Euler's method to solve this ODE, to get

$$q^{n+1} = \left(1 - \frac{\Delta t}{\eta}\right) q^n.$$

2. Assume the initial data is $q(0) = 1$. Solve the ODE exactly and show that $q \rightarrow 0$ as $t \rightarrow \infty$.
3. Solve the discrete relation from Euler's equation explicitly. Show that $q^N \rightarrow 0$ as $N \rightarrow \infty$ *only if* $\Delta t < \eta$.
4. Apply the backward difference formula to the original ODE to get the implicit backward Euler method. Show that this has the correct late time behaviour for all timesteps.
5. The η term here is setting a timescale in the problem. Near the ideal MHD it would be very short: much shorter than the timescales resulting from the stability requirements of the hyperbolic flux terms. Systems that need to cover a wide

range of timescales are often called *stiff*, and are more complex and expensive to solve. The use of implicit methods outlined here is standard, but expensive when applied to nonlinear problems. This will be important when including short-range reaction and interaction terms that rapidly push the system to an equilibrium, often called *relaxation problems*.

Numerical Implementation

There should be code that you can use and extend available in the repository.

Upwind scheme

For this exercise we are interested in the advection equation,

$$\partial_t q + \partial_x q = 0$$

on $x \in [0, 1]$, using periodic boundary conditions, with initial data

$$q(x, 0) = \exp(\sin(2\pi x)).$$

1. Implement the upwind scheme

$$q_i^{n+1} = q_i^n + \frac{\Delta t}{\Delta x} (q_{i-1}^n - q_i^n).$$

For re-use in later exercises, you may want to think of the scheme in the form

$$q_i^{n+1} = q_i^n + \Delta t L(\mathbf{q})_i^n,$$

and implement a function for L .

2. Using a grid of 20 points, evolve for one period and compare to the initial data. Use a CFL factor of 0.9 (so $\Delta t = 0.9\Delta x$). Identify the errors due to numerical dissipation and the phase errors.
3. By eye, estimate how much damping occurs each period. Then evolve for ten periods and check how it matches your estimate.
4. Experiment with the CFL factor. See how different the results after one period are with CFL factors of 0.99, 0.5 or 0.1. Check that the code is unstable for CFL factors larger than 1.
5. Check how the scheme converges with grid resolution, measuring the errors in the $p = 1, 2, \infty$ norms, where the discrete error is

$$\mathcal{E}_p = \left(\Delta x \sum_i |q_i^{(\text{exact})} - q_i^{(\text{numerical})}|^p \right)^{1/p}.$$

This is best done by plotting \mathcal{E}_p against Δx on logarithmic scales, and checking how the slope of the line compares to an appropriate power of Δx .

Upwind and linear discontinuities, and Burgers equation

1. Use the advection equation. Change the initial data to a square wave,

$$q(x, 0) = \begin{cases} 1 & |x - \frac{1}{2}| < \frac{1}{4} \\ 0 & \text{otherwise} \end{cases}.$$

Repeat the tests in the previous exercise. Are the qualitative errors clear? Is the convergence behaviour clear?

2. How does modifying the CFL factor change this behaviour?

Upwind and Burgers equation

For this exercise we will also be interested in Burger's equation in the form

$$\partial_t q + \partial_x f(q) = 0, \quad f(q) = \frac{1}{2}q^2.$$

We will think of the upwind scheme as

$$q_i^{n+1} = q_i^n + \frac{\Delta t}{\Delta x} L(\mathbf{q})_i^n,$$
$$L(\mathbf{q})_i^n = \frac{1}{\Delta x} (f_{i-1}^n - f_i^n),$$

remembering that $f = q$ for the advection equation.

1. Write a code to solve Burgers equation using the upwind method. Remember that the CFL condition implies that

$$\Delta t < \frac{\Delta x}{\max |\lambda|},$$

where λ is the characteristic speed (here $\lambda = f' = q$).

2. Apply your method to the smooth initial data

$$q(x, 0) = \exp(\sin(2\pi x)),$$

comparing how it evolves at $t = 0.05, 0.1, 0.15$ and 0.2 . Check this matches your expectations. Using 200 cells is reasonable here.

3. Look at the results with very few cells to see how and when the errors come in. Cell numbers of 10×2^k for $k = 0, \dots, 3$ are reasonable here. Why should you expect the shock to be well captured even at low resolution?
4. Apply your method to the discontinuous initial data

$$q(x, 0) = \begin{cases} 1 & |x - \frac{1}{2}| < \frac{1}{4} \\ 0 & \text{otherwise} \end{cases}.$$

Do the results make sense?

Godunov's method

Godunov's method is the scheme that started shock-capturing methods. By integrating the conservation law in spacetime over a finite-volume cell we get

$$\hat{q}_i^{n+1} = \hat{q}_i^n + \frac{\Delta t}{\Delta x} (f_{i-1/2}^n - f_{i+1/2}^n).$$

The key point is that the intercell fluxes are given as some function of the neighbouring cells,

$$f_{i-1/2} \equiv F(\hat{q}_{i-1}, \hat{q}_i).$$

Here we will use the *Rusanov* or *global Lax-Friedrichs* flux approximation,

$$f_{i-1/2} = \frac{1}{2} \left(f(\hat{q}_{i-1}) + f(\hat{q}_i) + \frac{\Delta x}{\Delta t} (\hat{q}_{i-1} - \hat{q}_i) \right).$$

1. Implement Godunov's method with the Rusanov flux for Burgers equation. Solve the same initial data as above.
2. This flux approximation is *very diffusive*, and produces a characteristic "staircase" pattern, compared to the upwind method. How does this appear in the solution?
3. Can you see which flux approximation makes Godunov's method equivalent to the upwind scheme for the advection equation? How about for Burgers equation?

Systems and Euler's equations

We are really interested in solving systems of equations, particularly fluids. We can directly extend Godunov's method with the Rusanov flux to the Newtonian Euler equa-

tions

$$\partial_t \begin{pmatrix} \rho \\ \rho v \\ \rho(e + \frac{1}{2}v^2) \end{pmatrix} + \partial_x \begin{pmatrix} \rho v \\ \rho v^2 + p \\ (\rho(e + \frac{1}{2}v^2) + p)v \end{pmatrix} = \mathbf{0}$$

with the γ law equation of state $p = (\gamma - 1)\rho e$. The maximum characteristic speed is $|v| + c_s$ where the speed of sound is

$$c_s^2 = \frac{\gamma p}{\rho}.$$

1. Implement Godunov's method with the Rusanov flux and apply it to Sod's problem, where the initial data is

$$\begin{pmatrix} \rho \\ v \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix} & x < 0.5 \\ \begin{pmatrix} \frac{1}{8} \\ 0 \\ \frac{1}{10} \end{pmatrix} & x > 0.5 \end{cases}.$$

Here $\gamma = \frac{7}{5}$, $x \in [0, 1]$, and the system is solved up to $t = 0.2$. Solve using 40 grid cells. You should see three waves: a rarefaction moving to the left, a contact wave (jump in density but not pressure or velocity) moving slowly right, and a shock moving quickly to the right.

2. Check (qualitatively!) how the solution converges with resolution by looking at 100, 400 and 1600 cells.
3. Modify your code to use the HLLC flux. For this we need to know the largest and smallest signal speeds ξ_{\pm} , which for the Euler equations are

$$\xi_{\pm} = v \pm c_s.$$

We then have the flux

$$f_{i-1/2} = \frac{\hat{\xi}_+ f(\hat{q}_{i-1}) - \hat{\xi}_- f(\hat{q}_i) - \hat{\xi}_+ \hat{\xi}_- (\hat{q}_{i-1} - \hat{q}_i)}{\hat{\xi}_+ - \hat{\xi}_-},$$

where

$$\hat{\xi}_+ = \max(0, \xi_+), \quad \hat{\xi}_- = \min(0, \xi_-).$$

The values of ξ can be computed globally (better than Rusanov but can be improved) or locally (the smallest and largest over the values considered, which is the most complex but most accurate). Compare the results.

4. Check theoretically that there is a limit where HLLE reduces to Rusanov.
5. Note that in relativistic cases, particularly MHD, computing the spectral information can be costly, and we already know that the extreme signal speeds are going to be bounded by c . For this reason Rusanov or similar fluxes are more popular in relativistic codes than the rest of astrophysics.

SR Hydrodynamics

The Euler equations in special relativity (Minkowski spacetime, Cartesian coordinates, $1 + 1$ dimensions) appear as a “minor” extension of the Newtonian case. We will use $c = 1$ everywhere in what follows. The equations are

$$\partial_t \begin{pmatrix} D \\ S \\ \tau \end{pmatrix} + \partial_x \begin{pmatrix} Dv \\ Sv + p \\ (\tau + p)v \end{pmatrix} = \mathbf{0},$$

where the conserved variables are

$$\begin{pmatrix} D \\ S \\ \tau \end{pmatrix} = \begin{pmatrix} \rho_0 W \\ \rho_0 h W^2 v \\ \rho_0 h W^2 - p - D \end{pmatrix},$$

with the auxilliary variables being the Lorentz factor $W = (1 - v^2)^{-1/2}$ and the specific enthalpy $h = 1 + \epsilon + p/\rho_0$ and with the γ law equation of state $p = (\gamma - 1)\rho\epsilon$. The maximum characteristic speed is

$$\max |\lambda| = \max \left| \frac{v \pm c_s}{1 \mp v c_s} \right|,$$

which extends the Newtonian case using the SR velocity addition rule, and where the speed of sound is

$$c_s^2 = \frac{\gamma p}{\rho_0 h}.$$

Note that the maximum characteristic speed is always bounded by $c = 1$.

The main additional difficulty compared to the Newtonian case is the conversion between conserved and primitive variables. In general, and in contrast with the conversion from primitive to conserved, this cannot be done in closed form, but needs an iterative algorithm. One algorithm is written as follows. Assume we are given physical values for

the conserved variables D, S, τ . Guess the value of the pressure, calling this guess \bar{p} . Compute a guess for the velocity magnitude from

$$\bar{v}^2 = \frac{S^2}{(\tau + \bar{p} + D)^2}.$$

From this compute a guess for the Lorentz factor as

$$\bar{W} = (1 - \bar{v})^{-1/2}.$$

From this compute guess for the density

$$\bar{\rho}_0 = \frac{D}{\bar{W}},$$

the specific enthalpy

$$\bar{h} = \frac{\tau + \bar{p} + D}{\bar{\rho}_0 \bar{W}^2},$$

and hence the specific internal energy

$$\bar{\epsilon} = \bar{h} - 1 - \frac{\bar{p}}{\bar{\rho}_0}.$$

We can then compare the pressure guess to the pressure from the equation of state,

$$f(\bar{p}) = \bar{p} - p(\bar{\rho}_0, \bar{\epsilon}) = (\gamma - 1)\bar{\rho}_0\bar{\epsilon}.$$

When this nonlinear function $f(\bar{p}) = 0$ then the guess for \bar{p} is consistent (but needs to be checked to ensure it's physical, i.e., that $\bar{p} > 0$). At that point we have the primitive variables.

1. Write a function to convert conserved to primitive variables. There are many libraries for finding the root of a nonlinear function that can be used. Check it by converting random (physical!) primitive variables to conserved variables and then converting back. Note: as an initial guess for the root-finder it can be helpful to use that the physical range is $p > \max(0, |S| - \tau - D)$.
2. Extend your Godunov solver with the Rusanov flux to the special relativistic case. Solve the Sod problem with the same initial data and value for γ as the Newtonian case, but evolve to $t = 0.4$. Check the qualitative similarity with the Newtonian results. You may want to reduce the number of cells used.

3. Attempt the blast wave problem with initial data

$$\begin{pmatrix} \rho \\ v \\ p \end{pmatrix} = \begin{cases} \begin{pmatrix} 1 \\ 0 \\ 1000 \end{pmatrix} & x < 0.5 \\ \begin{pmatrix} \frac{1}{8} \\ 0 \\ \frac{1}{100} \end{pmatrix} & x > 0.5 \end{cases}.$$

For this problem use $\gamma = \frac{5}{3}$ and an end time of $t = 0.4$. Compare to the solution in [Martí and Müller's Living Review \(Problem 2, Section 6\)](#), and see just how difficult this problem is to capture.

Slope limiting

All the codes implemented above are first order accurate, as we saw when checking convergence for the advection equation. There are two key steps to improve accuracy: better accuracy in space, and better accuracy in time. Here we focus on the first.

Consider our update scheme

$$q_i^{n+1} = q_i^n + \frac{\Delta x}{\Delta t} (f_{i-1/2} - f_{i+1/2})$$

where the intercell flux $f_{i-1/2}$, such as the Rusanov flux, is a function of two arguments,

$$f_{i-1/2} = F(q_L, q_R).$$

In the Godunov scheme the left and right states were given by the cell values, so $q_L = q_{i-1}$, for example. We can modify that by using a *reconstruction* that instead interpolates the “true” solution q^n from both sides of the cell interface, replacing $q_{L,R}$ with the one-sided limits of this interpolation.

In slope limiting we say, for example,

$$q_L = q_{i-1} + \frac{\Delta x}{2} \sigma_{i-1}$$

where σ is the *slope* of q within that cell. There are three obvious approximations:

$$\begin{aligned}\sigma_{\text{upwind}} &= \frac{q_i - q_{i-1}}{\Delta x}, \\ \sigma_{\text{downwind}} &= \frac{q_{i-1} - q_{i-2}}{\Delta x}, \\ \sigma_{\text{centred}} &= \frac{1}{2} (\sigma_{\text{upwind}} + \sigma_{\text{downwind}}) \\ &= \frac{q_i - q_{i-2}}{2\Delta x}.\end{aligned}$$

None of these will work in general, as all will cause oscillations at shocks. The idea is to *limit* the slope to avoid the oscillations. We write the limited slope as

$$\bar{\sigma} \equiv \bar{\sigma}(\sigma_{\text{upwind}}, \sigma_{\text{downwind}}),$$

and choose the limiting slope to be close to the centred slope (which is second order accurate) where possible, but close to zero (which reproduces Godunov's algorithm, which is stable) where we may be at a shock.

The simplest limiter is *minmod*. The minmod function is defined as

$$\text{minmod}(a, b) = \begin{cases} 0 & a \cdot b \leq 0 \\ a & a \cdot b > 0 \text{ and } |a| < |b| \\ b & \text{otherwise} \end{cases}$$

Applied to the slopes it picks the smallest slope in magnitude, unless the slopes have different signs, when it picks zero.

Note that the factors of Δx cancel out in the steps above, so need not be included in implementations.

1. Implement a slope-limited code for the advection and Burgers equations. The starting point should be the Godunov code first implemented for Burgers equation. Note that you will now need to use two ghostzones at the boundaries.
2. Check the convergence rate of the smooth solution for the advection equation as above. Compared to the upwind scheme the results will be poor: this is largely due to the Rusanov flux.
3. Check the behaviour of the algorithm at discontinuities for both advection and Burgers equations.

Strong stability preservation

We now need to improve the behaviour in time. The simplest way to do this that extends to complex systems in multiple dimensions is to use the *method of lines*. Start from the conservation law

$$\partial_t q + \partial_x f(q) = 0.$$

Introduce a (finite volume style) grid in space and integrate over it, leading to

$$\partial_t \hat{q}_i = L(\mathbf{q}) = \frac{1}{\Delta x} (f_{i-1/2} - f_{i+1/2}).$$

This is now an *ordinary* differential equation for the cell averages \hat{q} . However, it is one ODE for *each* cell.

We can now use a standard ODE solver for this problem. However, again shocks can give problems. We can avoid problems by using a *strong stability preserving* (SSP) solver. The simplest explicit solver is a second order Runge-Kutta method. This has the form (dropping hats)

$$\begin{aligned}\mathbf{q}^{(1)} &= \mathbf{q}^n + \Delta t L(\mathbf{q}^n), \\ \mathbf{q}^{n+1} &= \frac{1}{2} (\mathbf{q}^n + \mathbf{q}^{(1)} + \Delta t L(\mathbf{q}^{(1)})).\end{aligned}$$

This is a two stage method. Note that the boundary conditions need to be imposed on $\mathbf{q}^{(1)}$ before the second stage can be computed.

1. Implement a solver using RK2 in time, minmod limiting in space, and the Rusanov solver. Apply it to the standard advection and Burgers equation tests.
2. The errors may appear disappointingly large, and the convergence rate qualitatively little better than before. Try implementing the upwind flux formula (which is generically unstable, but will work for advection)

$$F(q_L, q_R) = f(q_L)$$

and repeat the convergence test for the advection equation. Check that the result is qualitatively better than previous algorithms.

Slope limiting and systems

The advantage of using the Rusanov flux is that all the steps in the previous exercise extend to systems.

1. Extend the RK2, minmod, Rusanov algorithm to the various Euler equations, and see how much difference it makes.
2. You may find that improving the accuracy in space and time whilst keeping the very diffusive Rusanov solver has helped less than the change from Rusanov to HLL solver did. You may want to try this or a more complex solver.

Flux split finite differencing

The algorithms we have implemented so far *reconstruct* some variable(s) and use the reconstructions to *solve* for the intercell flux. The accuracy of the schemes depends on the accuracy of the reconstruction and the accuracy of the flux approximation.

An alternative approach is to reconstruct the flux directly. With these schemes our interpretation of the update formula changes. In

$$\partial_t \hat{q}_i = L(\mathbf{q}) = \frac{1}{\Delta x} (f_{i-1/2} - f_{i+1/2})$$

the values $f_{i\pm 1/2}$ are no longer *intercell* fluxes, but approximations such that the L terms approximates the flux derivative to the appropriate order. This makes it easier to accurately extend to higher dimensions.

The problem is reconstructing the fluxes in a way that is stable.

1. Define the *upwind* and *downwind* parts of the flux through

$$f^{(\pm)} = \frac{1}{2} (f(q) \pm \alpha q) .$$

Here α is a constant. Show theoretically that, if $\alpha \geq \max |\lambda|$,

$$\partial_q f^{(+)} \geq 0$$

$$\partial_q f^{(-)} \leq 0.$$

This means that $f^{(+)}$ propagates in the “positive” direction and $f^{(-)}$ in the negative direction. Think how this extends to the system case.

2. Define

$$f_{i-1/2} = f_L^{(+)} + f_R^{(-)}$$

where the L, R subscript indicates whether the variable has been reconstructed *from* the left or right. This indicates that the positively moving part of the flux,

$f^{(+)}$, is reconstructed from “left to right”, and the negatively moving part of the flux, $f^{(-)}$, is reconstructed from “right to left”. The flux is then recombined. Check that if the reconstruction is piecewise constant (Godunov method style) that the Godunov method with Rusanov flux approximation is recovered.

3. Use the minmod reconstruction applied to the split fluxes, together with the second order Runge-Kutta method, to get a flux split finite difference method. Apply it to the advection equation and check the convergence rate.
4. Use the provided WENO reconstruction codes (which are third or fifth order accurate and need two or three ghostzones) and the three stage third order SSP RK3 method

$$\begin{aligned} \mathbf{q}^{(1)} &= \mathbf{q}^n + \Delta t L(\mathbf{q}^n), \\ \mathbf{q}^{(2)} &= \frac{1}{4} (3\mathbf{q}^n + \mathbf{q}^{(1)} + \Delta t L(\mathbf{q}^{(1)})) \\ \mathbf{q}^{n+1} &= \frac{1}{3} (\mathbf{q}^n + 2\mathbf{q}^{(2)} + 2\Delta t L(\mathbf{q}^{(2)})) \end{aligned}$$

to get higher order schemes. Check the convergence rate.

Higher order Euler methods

When we move to systems performing higher order reconstruction, whether to use as input to a flux function, or when applied to the flux directly, can cause problems.

1. Extend your WENO code to apply to the Euler equations. Check the results on the Sod problem.
2. Extend it again to apply to the SR Euler equations. Check the results on the Sod and blast wave problems. What problems do you see?
3. Oscillations from applying the high order method can be reduced and sometimes completely eliminated by transforming (locally, about each cell interface) to characteristic variables. Check the earlier exercises to see the issues with this transformation.

Toy stars

Evolving neutron stars means dealing with compact objects with a surface. [Price introduced toy stars](#) to test the behaviour of codes near these surfaces.

Take the Newtonian Euler equations and add a source term that depends on position:

$$\partial_t \begin{pmatrix} \rho \\ \rho v \\ \rho(e + \frac{1}{2}v^2) \end{pmatrix} + \partial_x \begin{pmatrix} \rho v \\ \rho v^2 + p \\ (\rho(e + \frac{1}{2}v^2) + p)v \end{pmatrix} = \begin{pmatrix} 0 \\ -\rho x \\ -\rho v x \end{pmatrix}.$$

Fixing $\gamma = 2$ so the equation of state is $p = \rho e$ means there is a static solution with

$$\begin{pmatrix} \rho \\ v \\ p \end{pmatrix} = \begin{pmatrix} 1 - x^2 \\ 0 \\ \frac{1}{4}(1 - x^2)^2 \end{pmatrix}.$$

This has positive density in $x \in [-1, 1]$. Outside of this region we assume the solution is vacuum, so that $\rho = 0 = p$. It gives a toy test that only considers the hydrodynamic reaction to a source.

The codes implemented so far will fail in vacuum, as the conversion between conserved and primitive variables is singular. The simplest solution is to add an *artificial atmosphere*: a low density region outside the star. When a point falls below a cutoff value, the state vector is set to the atmosphere values.

1. Extend your WENO Newtonian Euler code to solve the toy star. You will need to modify the update step to include a source term, and the functions converting between conserved and primitive values to account for the atmosphere. It will probably be necessary to impose an atmosphere cutoff on both ρ and e . A reasonable cutoff value is $\rho, e \sim 10^{-6}$.
2. Evolve the toy star data at low resolution (say 40 cells) until $t = 1$ and $t = 10$. Does the star settle down? What problems do you see?
3. Study the convergence behaviour with grid resolution at $t = 1$ (using say 100, 200 and 400 cells). In what sense do the results converge?
4. Revisit the questions on vacuum generation and well balancing in the light of your results here.

Higher dimensions

Extending an evolution code to higher dimensions can be done straightforwardly using *dimensional splitting*. Focus on the advection equation to start. Write the advection equation as

$$\partial_t q + \partial_x f^{(x)}(q) + \partial_y f^{(y)}(q) = \partial_t q + \partial_x q + \partial_y q = 0.$$

Using periodic boundaries, this will advect the initial data round a periodic domain with speed 1 in each direction.

Using a standard finite volume grid and the Method of Lines, write this in the semi-discrete form

$$\frac{d}{dt} \hat{q}_{i,j} + \frac{1}{\Delta x} (f_{i+1/2,j}^{(x)} - f_{i-1/2,j}^{(x)}) + \frac{1}{\Delta y} (f_{i,j+1/2}^{(y)} - f_{i,j-1/2}^{(y)}) = 0.$$

We need to be slightly careful when writing this down. If we are integrating over a finite volume cell (using a finite volume approach, as in slope limiting) then the flux terms are not point approximations to the intercell fluxes. Instead, they approximate the *integral average* of the flux over the cell face. Using the midpoint rule (which is the point-value interpretation) is second order accurate. The finite difference interpretation does not have this restriction.

In dimensional splitting each flux term is computed separately without considering any contributions that may come in “tangentially”. This is simple and does not reduce the *order* of accuracy, but may reduce the *absolute* accuracy, and may reduce the stable timestep.

1. Extend any of your advection codes to the two dimensional case.
2. Evolve both smooth and discontinuous initial data (like a Gaussian pulse, or a top hat function) around one period. See how it smears out. Note that typically the timestep will need to be reduced by a factor of D^{-1} or $D^{-1/2}$ where D is the spatial dimension. It is usually easiest to find this by low-resolution experiments.

Higher dimensions and MHD

To really test your codes, try implementing Newtonian MHD in two dimensions. The simplest implementation uses divergence cleaning, by introducing the field $\psi = \nabla \cdot B$. The resulting equations of motion are

$$\partial_t \begin{pmatrix} \rho \\ \rho v_j \\ \rho (e + \frac{1}{2} v^2) + \frac{1}{2} B^2 \\ B_j \\ \psi \end{pmatrix} + \partial_i \begin{pmatrix} \rho v^i \\ \rho v^i v_j + (p + \frac{1}{2} B^2) \delta_j^i - B^i B_j \\ (\rho (e + \frac{1}{2} v^2) + p + B^2) v^i - B^i B^k v_k \\ v^i B_j - B^i v_j + \psi \delta_j^i \\ B^i \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\frac{\psi}{c_p^2} \end{pmatrix}.$$

Here c_p is a free parameter controlling how fast the damping occurs.

1. Implement the equations of motion. Check that one dimensional constant states work, and that small perturbations propagate sensibly.
2. Attempt the [Orszag-Tang vortex test](#). With a simple Python code it may be difficult to get a reasonable solution in the time available. At this point, you may wish to move on to more efficient codes.