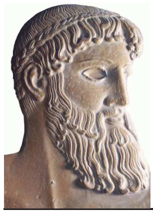
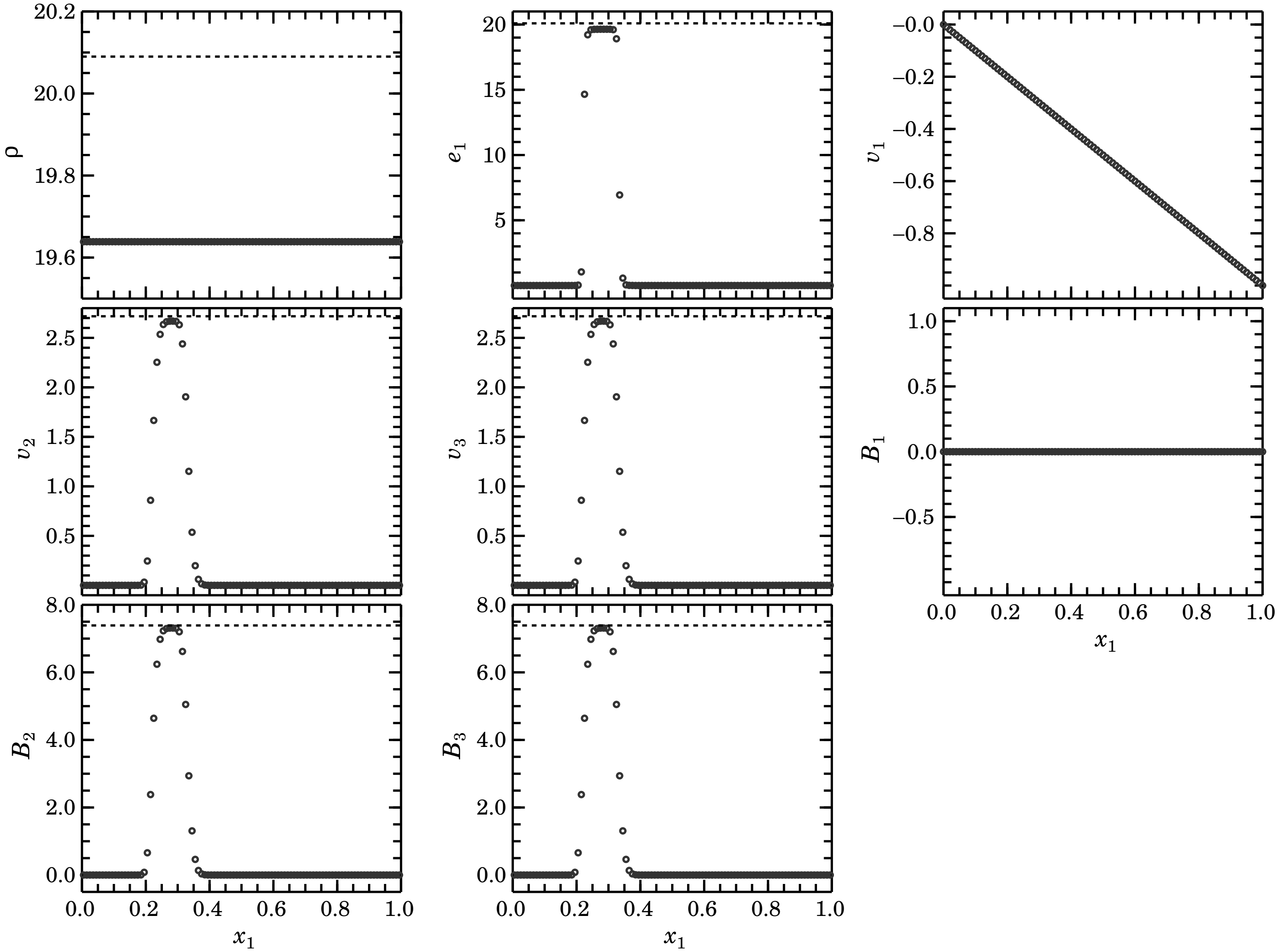


ZEUS-3D 1-D Gallery #3: Advection in Spherical Polar Coordinates



This is an advection problem[1] in spherical polar coordinates, where a square wave in each of e_1, v_2, v_3, B_2, and B_3 of amplitude 1 is initialised in $0.6 \leq x_1 \leq 0.9$ and allowed to migrate on a 1-D (radial) grid toward $r = x_1 = 0$ with an imposed velocity profile $v_1 = -x_1$. Density also evolves but as a uniform background quantity rather than a pulse which would interfere with the advection of velocity waves. Analytical expectations are ρ, $e_1 \sim e^{3t}$, $B_\perp \sim e^{2t}$, and $v_\perp \sim e^t$, with pulse widths narrowing as e^{-t}. Issues to be concerned with include monotonicity, widths and levels of pulses, and diffusion of discontinuities.

Open circles are the `dzeus36` solution using 100 zones, `CMoC`, `FIT`, no artificial viscosity, and `courno=0.5`. Third-order interpolation (`iords=3`) with discontinuity steepening (`istp=1`) is used for the scalars (e_1), and second order van Leer interpolation (`iord=2`) is used for the vector components, which is the highest order of interpolation compatible with `CMoC`. Dashed lines are the expected levels of the pulses at $t = 1$. Disagreement between numerical and analytical solution is most apparent on the density plot (note ordinate scale), and is due to second-order temporal discretisation errors which are most apparent in advection problems. The origin of these errors is discussed on the next page.

These solutions are virtually identical to those from all versions of *ZEUS* since `zeus04`.

[1]See the page for Cartesian advection for a working definition of *advection*.

Origin of second-order temporal discretisation errors

Consider the continuity equation,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \vec{v}\rho \;=\; 0. \tag{1}$$

For this 1-D (in r) advection problem, it is assumed $\rho = \rho(t)$ and $\vec{v} = -r\hat{e}_r$, the latter of which causes features to drift self-similarly toward the origin. Thus, in spherical polar coordinates, equation (1) reduces to:

$$\frac{\partial \rho(t)}{\partial t} \;=\; -\frac{1}{r^2}\frac{\partial r^2 v_r \rho(t)}{\partial r} \;=\; \rho(t)\frac{\partial r^3}{\partial r} \;=\; 3\rho(t) \quad \Rightarrow \quad \boxed{\rho(t) \;=\; \rho_0 e^{3t},} \tag{2}$$

is the analytical expectation for $\rho(t)$, as stated on the previous page. Thus, at time $t + \delta t$,

$$\rho(t+\delta t) \;=\; \rho_0 e^{3(t+\delta t)} \;=\; \rho_0 e^{3t} e^{3\delta t} \;=\; \rho(t) e^{3\delta t}$$

$$\Rightarrow \quad \boxed{\rho(t+\delta t) \;=\; \rho(t)\left(1 + 3\delta t + \tfrac{1}{2}(3\delta t)^2 + \cdots\right),} \tag{3}$$

making explicit the zeroth, first, and higher order terms of the advanced value for ρ.

ZEUS, like many explicit finite difference codes, does not use a true second-order accurate in time transport algorithm. For this problem, *ZEUS*' difference form of equation (1) reduces to:

$$\frac{\delta \rho}{\delta t} \;=\; \frac{\rho_i^{n+1} - \rho_i^n}{\delta t} \;=\; 3\rho_i^n \quad \Rightarrow \quad \boxed{\rho_i^{n+1} \;=\; \rho_i^n + 3\rho_i^n \delta t \;=\; \rho_i^n(1 + 3\delta t),} \tag{4}$$

where the super/subscripts, n and i, are the temporal and spatial indices respectively[2]. Since equation (4) includes only the zeroth- and first-order terms in equation (3), the transport algorithm in `dzeus36` is just first-order accurate in time for this particular problem (but up to third order accurate in space, as attested to by the sharpness of the discontinuities in the e_1 pulse on the previous page). It is the lack of the second order terms that gives rise to the second-order discretisation errors.

For m time steps after n, equation (4) evidently requires:

$$\rho_i^{n+m} \;=\; \rho_i^n (1 + 3\delta t)^m. \tag{5}$$

Thus, for $n = 0$, $\rho_i^0 = \rho_0 = 1$, $m = 198$ (number of time steps taken by this advection problem), and $t_{\max} = 1$, $\delta t = t_{\max}/m = 1/198$ and equation (5) can be evaluated to give:

$$\rho(1) \;=\; \rho_i^{198} \;=\; (1)\left(1 + \frac{3}{198}\right)^{198} \;=\; 19.64,$$

precisely the level of the open circles on the density plot. Using equation (2), the analytically expected value is $\rho(1) = (1)e^{3(1)} = 20.09$, the level of the dashed line. The 2.2% difference is then entirely attributable to second-order temporal discretisation errors.

A true second-order in time algorithm requires twice as much CPU and memory as a first-order algorithm. *Approximate* second order algorithms, like the one implemented in *ZEUS*, can get away with less, but the price paid is that in some problems—such as this advection problem—the algorithm reduces to first-order accuracy in time.

[2] That is, ρ_i^n represents the value of ρ in the i^{th} zone after n time steps.

One example of a true second-order algorithm is *extrapolation-integration*. Here, we *extrapolate* forward in time by just 1/2 of a time step using the first-order scheme in equation (5):

$$\rho_i^{n+1/2} \;=\; \rho_i^n \left(1 + 3\frac{\delta t}{2}\right),$$

and then use these half-time step values to *integrate* the solution to a full time step:

$$\frac{\delta\rho}{\delta t} \;=\; \frac{\rho_i^{n+1} - \rho_i^n}{\delta t} \;=\; 3\rho_i^{n+1/2} \quad\Rightarrow\quad \boxed{\rho_i^{n+1} \;=\; \rho_i^n + 3\rho_i^{n+1/2}\delta t \;=\; \rho_i^n \left(1 + 3\delta t + \tfrac{1}{2}(3\delta t)^2\right).} \tag{6}$$

Since equation (6) now includes the second-order term in equation (3), this algorithm is second-order accurate in time, and therefore would exhibit *third*-order discretisation errors. Thus, after m timesteps after n,

$$\rho_i^{n+m} \;=\; \rho_i^n \left(1 + 3\delta t + \tfrac{1}{2}(3\delta t)^2\right)^m \;=\; 20.0833,$$

for the values used above. By comparison, $e^3 = 20.0855$, and the now minuscule 0.011% error is why second-order algorithms are so desirable.