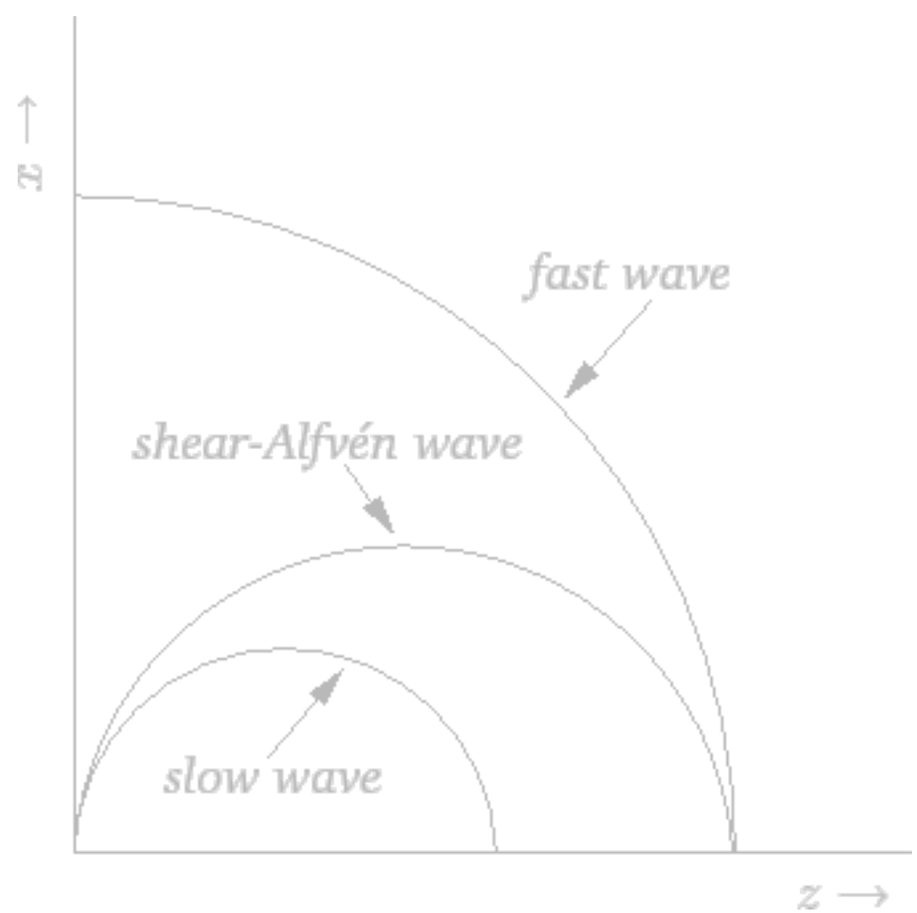
Topic 1

How to Write a 1-D MHD Code using Finite Difference?



Differential Equations in Space Plasmas

Fluid Description

$$\frac{\partial}{\partial t}(n_\alpha) + \nabla \cdot (n_\alpha \mathbf{u}_\alpha) = S_\alpha$$

Mass conservation

Momentum conservation

$$\frac{\partial}{\partial t}(n_\alpha \mathbf{u}_\alpha) + \nabla \cdot (n_\alpha \mathbf{u}_\alpha \mathbf{u}_\alpha + \mathbf{P}_\alpha) - \frac{n_\alpha q_\alpha}{m_\alpha}(\mathbf{E} + \mathbf{u}_\alpha \times \mathbf{B}) = \mathbf{R}_\alpha$$

Thermal Dynamics

$$\frac{\partial p_\alpha}{\partial t} + \mathbf{u}_\alpha \cdot \nabla p_\alpha + \gamma p_\alpha \nabla \cdot \mathbf{u}_\alpha = Q_\alpha$$

Maxwell's equation

$$\frac{\partial \mathbf{B}}{\partial \mathbf{t}} + \nabla \times \mathbf{E} = 0$$

$$\frac{\partial}{\partial t} : \text{ time derivative}$$

$$\nabla : \text{ spatial derivative}$$

$$= \frac{\partial}{\partial x}\hat{\mathbf{x}} + \frac{\partial}{\partial y}\hat{\mathbf{y}} + \frac{\partial}{\partial z}\hat{\mathbf{z}}$$

So the first key element of computation space plasma physics is to approximate these derivatives

Kinetic Description

Boltzmann equation

$$\frac{\partial f_s}{\partial t} + \nabla_{\mathbf{x}} \cdot (\mathbf{v} f_s) + \nabla_{\mathbf{v}} \cdot (\mathbf{F}_s f_s) = \left(\frac{\delta f_s}{\partial t}\right)_c$$

$$\mathbf{F}_s = q_s / m_s (\mathbf{E} + \mathbf{v} \times \mathbf{B})$$

Lorentz Force

$$\frac{\partial \mathbf{B}}{\partial \mathbf{t}} + \nabla \times \mathbf{E} = 0$$

Maxwell's equations

$$\mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{B} = -\mu_0 \sum_s q_s \int_{-\infty}^{+\infty} v f_s d^3 v$$

Particle Description

$$m_s n_s \frac{d\mathbf{v}_s}{dt} = q n_s (\mathbf{E} + \mathbf{v}_\mathbf{s} \times \mathbf{B})$$

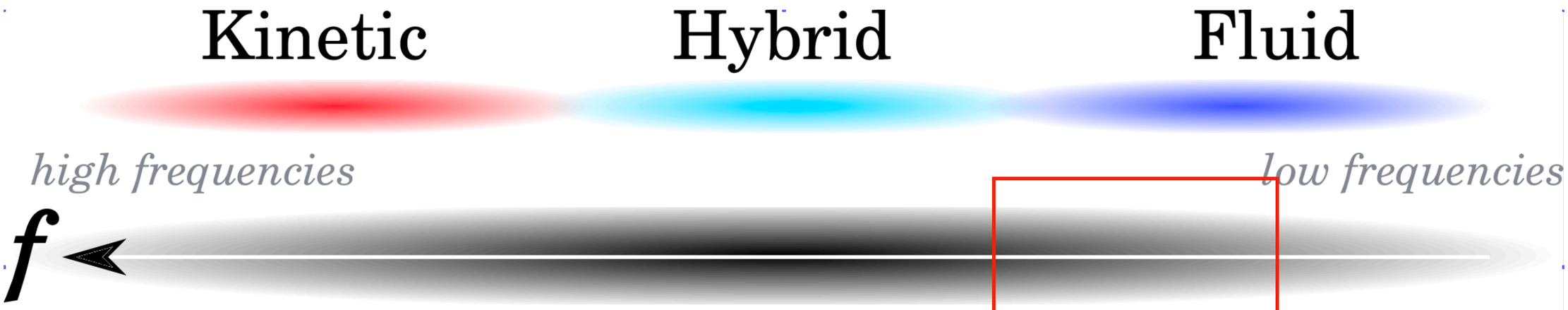
Equation of motion

$$\frac{\partial \mathbf{B}}{\partial \mathbf{t}} + \nabla \times \mathbf{E} = 0$$

Maxwell's equations

$$\mu_0 \epsilon_0 \frac{\partial \mathbf{E}}{\partial t} - \nabla \times \mathbf{B} = -\mu_0 \sum_s q_s n_s \mathbf{v}_s$$

Spectrum of Simulating Space Plasmas



	ω_{pe}	ω_{pi}	ω_{ci}	ω_a	ω_{ei}
Frequency	electron plasma	ion plasma	ion gyrofrequency	Alfvén	electron-ion collision
Timescale	$\leq 10^{-12}$ s	$\sim 10^{-10} - 10^{-8}$ s		$\sim 10^{-6} - 10^{-3}$ s	10^{1} s
Simulation Methods	Conventional Explicit Particle Codes	Ion-scale Particle Codes, Hybrid Codes		MHD Explicit	MHD Implicit (resistive)

TOPIC 1-2

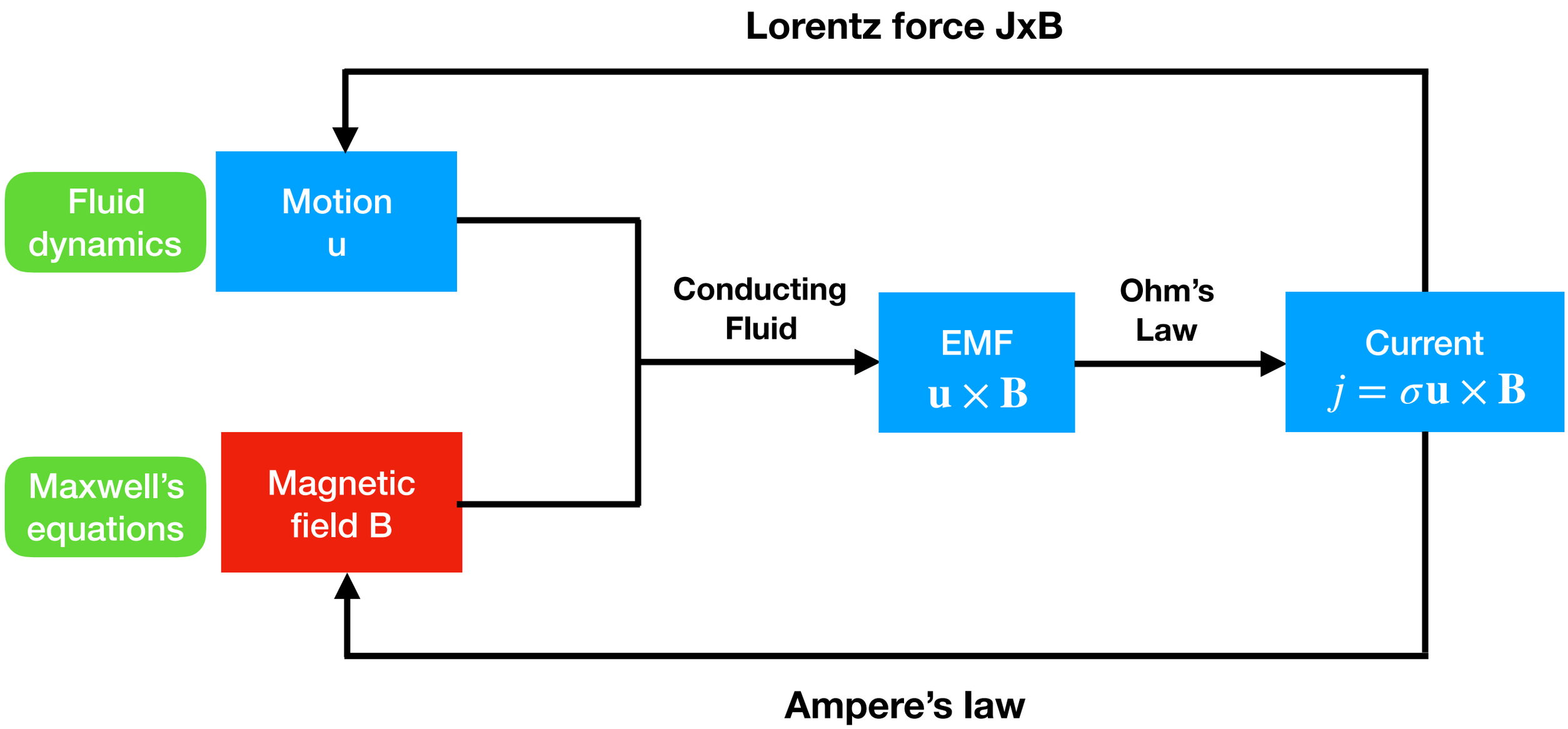
Vlasov codes

Guiding Center
Gryokinetic
Implicit particle codes

Physics

What is MHD?

Dynamics in MHD: coupling between conducting fluid and magnetic field



The nature of MHD: conservation laws

The MHD equations to solve

Unit Systems

SI Units

Mass Equation

$$\frac{\partial}{\partial t}\rho = -\nabla \cdot \rho \mathbf{u}$$

Velocity Equation

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{J} \times \mathbf{B}$$

Pressure Equation

$$p = p_g \left(\frac{\rho}{\rho_0}\right)^\gamma$$

Faraday's Law

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

Ohm's Law

$$\mathbf{E} = -\mathbf{u} \times \mathbf{B}$$

Ampere's Law

$$\mathbf{J} = \frac{1}{\mu_0} \nabla \times \mathbf{B}$$

CGS Units

Mass Equation

$$\frac{\partial}{\partial t}\rho = -\nabla \cdot \rho \mathbf{u}$$

Velocity Equation

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \frac{1}{c} \mathbf{J} \times \mathbf{B}$$

Pressure Equation

$$p = p_g \left(\frac{\rho}{\rho_0}\right)^\gamma$$

Faraday's Law

$$\frac{\partial \mathbf{B}}{\partial t} = -c \nabla \times \mathbf{E}$$

Ohm's Law

$$\mathbf{E} = -\frac{1}{c} \mathbf{u} \times \mathbf{B}$$

Ampere's Law

$$\mathbf{J} = \frac{c}{4\pi} \nabla \times \mathbf{B}$$

Depending on the the choice of the unit systems, there are different physical constants floating around - awkward

The full set of Normalized MHD equations

Dimensionless Equations

Mass Equation

$$\frac{\partial}{\partial \tilde{t}}\tilde{\rho} = -\tilde{\nabla}\cdot(\tilde{\rho}\tilde{\mathbf{u}})$$

Velocity Equation

$$\tilde{\rho}\frac{D\tilde{\mathbf{u}}}{D\tilde{t}} = -\tilde{\nabla}\tilde{p} + \tilde{\mathbf{J}}\times\tilde{\mathbf{B}}$$

Pressure Equation

$$\tilde{p} = \frac{\beta}{2}\tilde{\rho}^{\gamma}$$

Faraday's Law

$$\frac{\partial \tilde{\mathbf{B}}}{\partial \tilde{t}} = -\tilde{\nabla}\times\tilde{\mathbf{E}}$$

Ohm's Law

$$\tilde{\mathbf{E}} = -\tilde{\mathbf{u}}\times\tilde{\mathbf{B}}$$

Ampere's Law

$$\tilde{\mathbf{J}} = \tilde{\nabla}\times\tilde{\mathbf{B}}$$

Normalization Relations

$$L_0 = T_0 U_0 \qquad U_0 = B_0/\sqrt{\mu_0\rho_0}$$

$$E_0 = U_0 B_0 \qquad J_0 = \frac{B_0}{\mu_0 L_0}$$

Interpretations:

- this set of normalized equation is called "dimensionless", e.g., u = 1 means the magnitude of the velocity is one Alfvén speed - very convenient in understanding the solutions
- We ignore the ***tilde*** in the following analysis and use the following set of MHD equations:

$$\frac{\partial}{\partial t}\rho = -\nabla\cdot\rho\mathbf{u}$$

$$\rho\frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{J}\times\mathbf{B}$$

$$p = \frac{\beta_0}{2}\rho^{\gamma}$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla\times\mathbf{E}$$

$$\mathbf{E} = -\mathbf{u}\times\mathbf{B}$$

$$\mathbf{J} = \nabla\times\mathbf{B}$$

Dimensionless Ideal MHD equations

This is the set of equations solved in mhd.m

1-D MHD equations

Plasma variables

$$\rho, u_x, u_y, p$$

Field variables

$$B_y, E_z, J_z$$

Put the equations together:

$$\frac{\partial}{\partial t}\rho = -\nabla \cdot \rho \mathbf{u}$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{J} \times \mathbf{B}$$

$$p = \frac{\beta_0}{2}\rho^\gamma$$

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E}$$

$$\mathbf{E} = -\mathbf{u} \times \mathbf{B}$$

$$\mathbf{J} = \nabla \times \mathbf{B}$$

$$\frac{\partial}{\partial t}\rho = -\frac{\partial}{\partial x}(\rho u_x)$$

$$\frac{\partial u_x}{\partial t} = -u_x \frac{\partial u_x}{\partial x} + \frac{1}{\rho}\left(-\frac{\partial p}{\partial x} - J_z B_y\right)$$

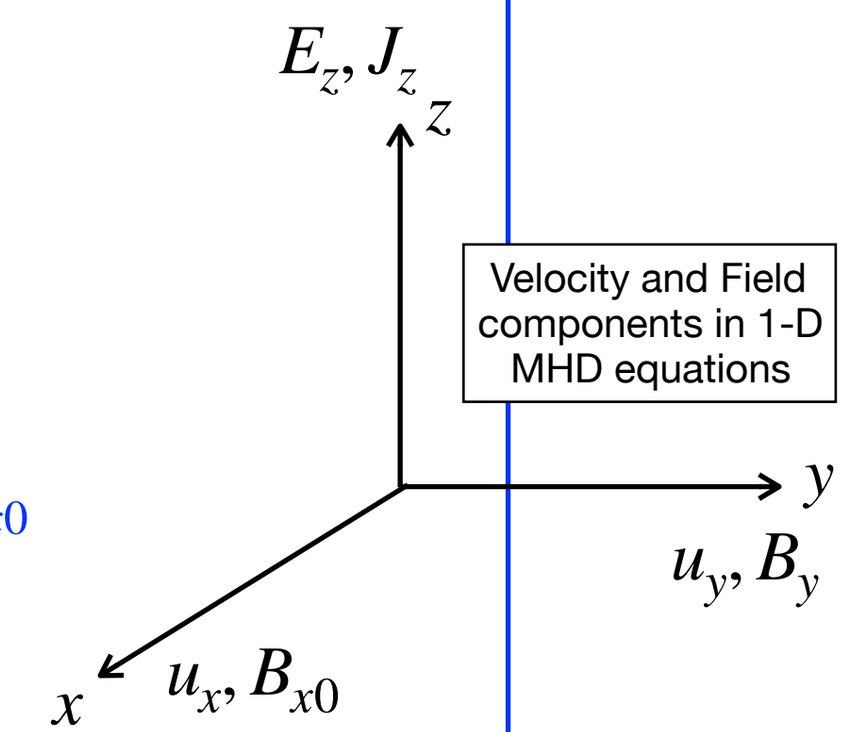
$$\frac{\partial u_y}{\partial t} = -u_x \frac{\partial u_y}{\partial x} + \frac{1}{\rho} J_z B_{x0}$$

$$p = \frac{\beta_0}{2}\rho^\gamma$$

$$\frac{\partial B_y}{\partial t} = \frac{\partial E_z}{\partial x}$$

$$E_z = -u_x B_y + u_y B_{x0}$$

$$J_z = \frac{\partial B_y}{\partial x}$$



Velocity and Field components in 1-D MHD equations

Physics

Waves in the 1-D MHD equations

First, for a linear polarized traveling wave, assume all perturbed terms vary with x and t like

$$\sim e^{i(kx-\omega t)}$$

The spatial and time derivatives become simply algebra calculations: $\frac{\partial}{\partial x} = ik \quad \frac{\partial}{\partial t} = -i\omega$

Now let's linearize the MHD equations and see what's the relationship between omega and k

Assuming $\rho = \rho_0,\ u_x = 0,\ u_y = \delta u_y\ B_y = \delta B_y$

$$\mathbf{J} = \nabla \times \mathbf{B} \longrightarrow J_z = \frac{\partial B_y}{\partial x} \longrightarrow \delta J_z = ik\delta B_y$$

$$\mathbf{E} = -\mathbf{u} \times \mathbf{B} \longrightarrow E_z = -u_x B_y + u_y B_{x0} \longrightarrow \delta E_z = \delta u_y B_0$$

$$\rho \frac{D\mathbf{u}}{Dt} = -\nabla p + \mathbf{J} \times \mathbf{B} \longrightarrow -i\omega\rho_0 \delta u_y = \delta J_z B_0$$

$$\frac{D\mathbf{u}}{Dt} \approx \frac{\partial \mathbf{u}}{\partial t}$$

Linearize

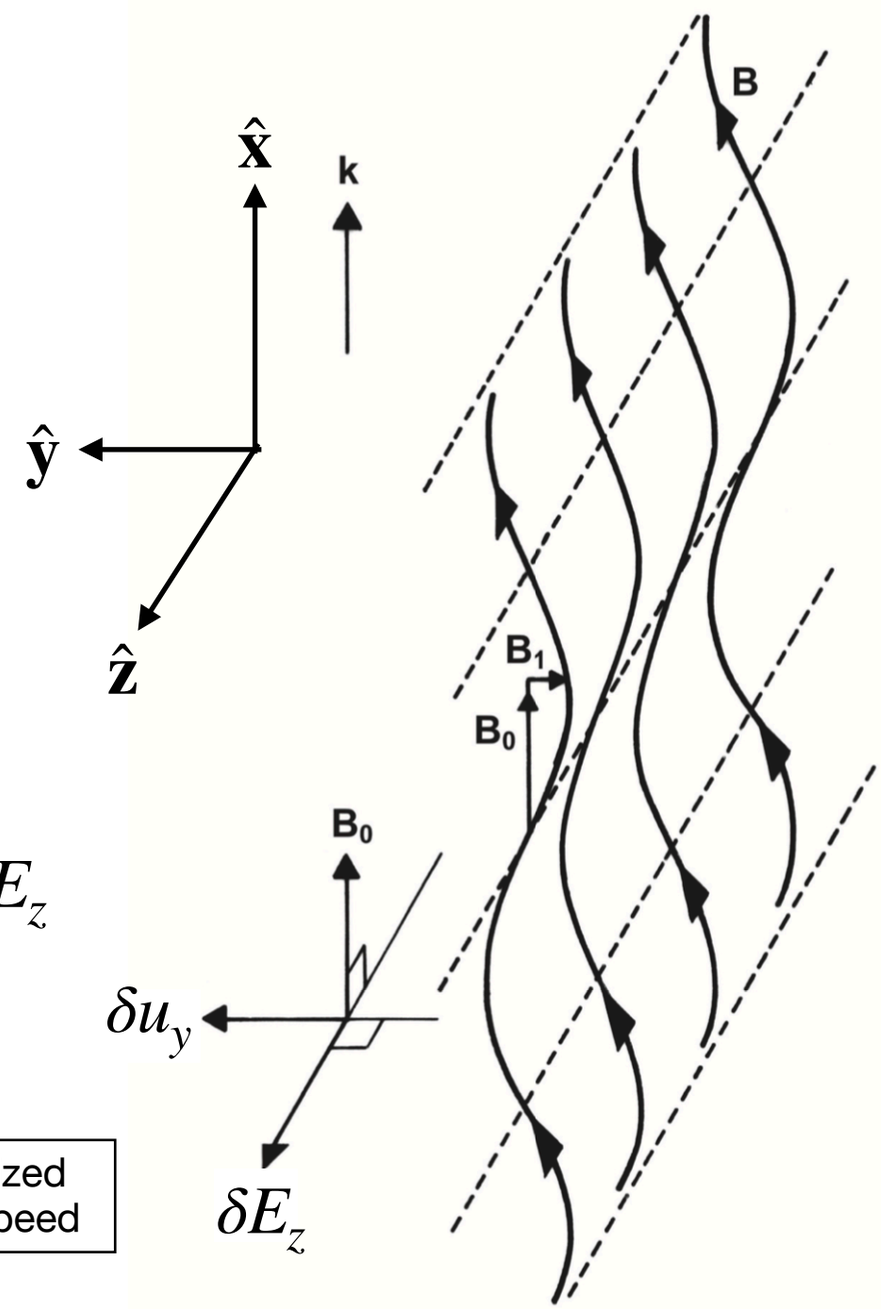
$p \approx 0$

Cold plasma Approximation

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \longrightarrow -i\omega\delta B_y = ik\delta E_z$$

$$\boxed{\left(\frac{\omega}{k}\right)^2 = \frac{B_0^2}{\rho_0} = 1}$$

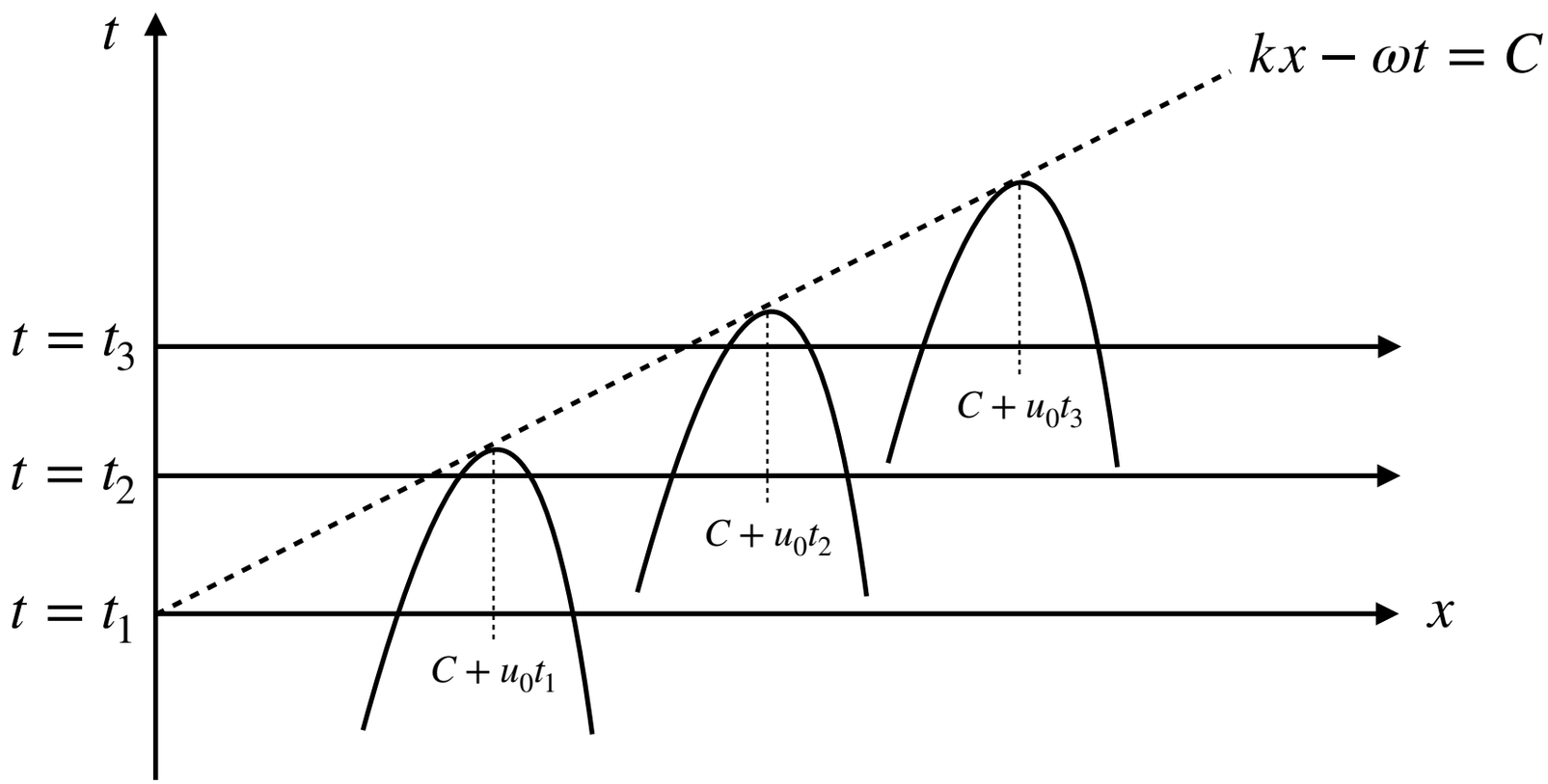
Normalized Alfvén speed



Traveling versus Standing Waves

In the Alfvén wave solution, note that $\dfrac{\omega}{k} = \pm 1$

- If k > 0, we get a wave traveling the +x direction - easy to see that from the phase of the wave: $(kx - \omega t)$ needs to keep constant
- If k < 0, we get a wave traveling the -x direction



To keep $kx - \omega t$ constant: $\quad kx - \omega t = \text{const} \longrightarrow \dfrac{dx}{dt} = \dfrac{\omega}{k} = \pm 1$

Wave traveling in both directions!

Traveling versus Standing Waves

Now add a wave traveling to the right with amplitude δu_{y0} and a wave traveling to the left with the same amplitude

$$\delta u_y = \delta u_{y0}\left[\underbrace{e^{i(kx-\omega t)}}_{\text{Right Wave}} + \underbrace{e^{i(-kx-\omega t)}}_{\text{Left Wave}}\right]$$

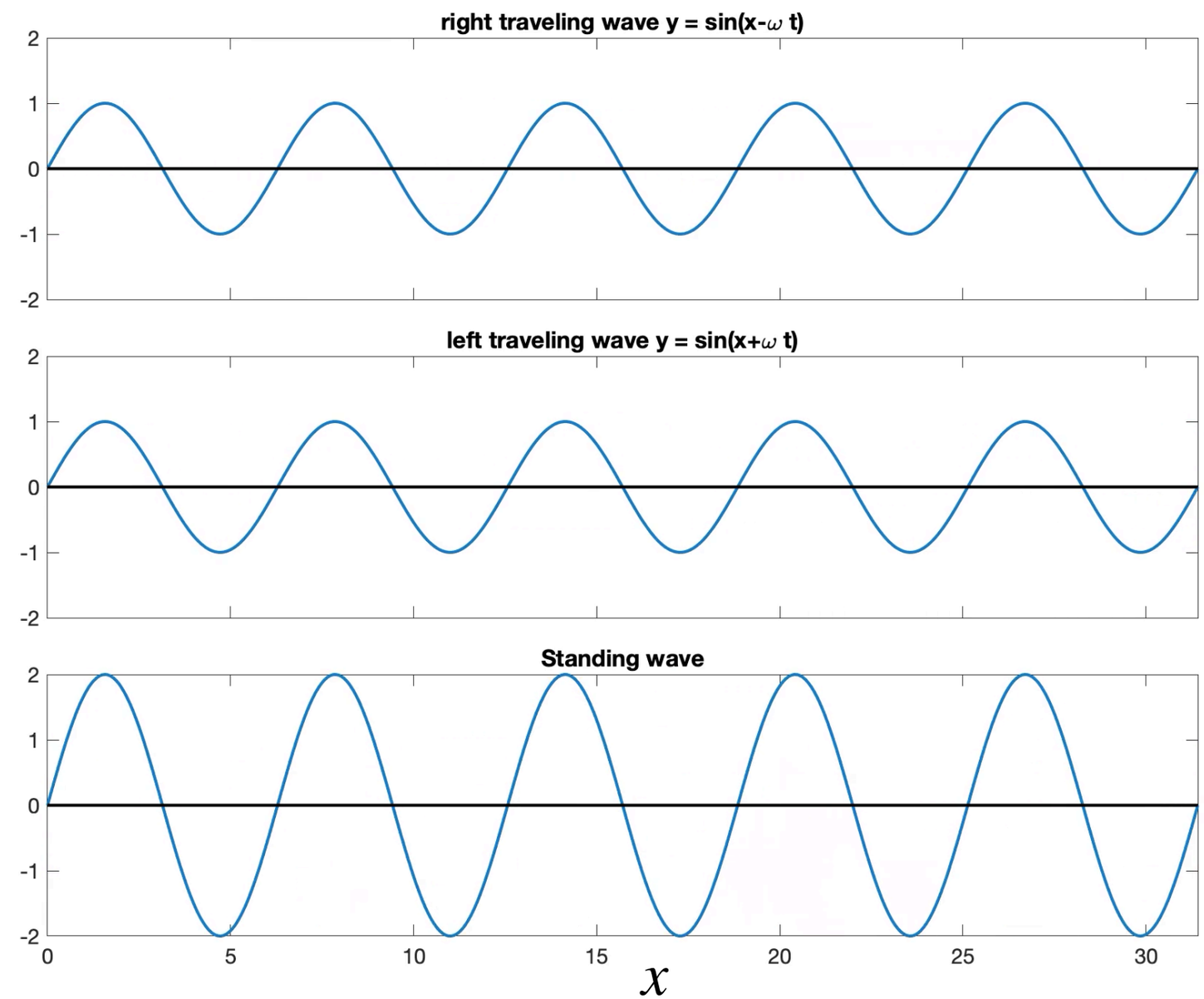
The total wave becomes

$$\delta u_y = \delta u_{y0} e^{-i\omega t}\left[e^{ikx} + e^{-ikx}\right]$$

$$= 2\delta u_{y0} e^{-i\omega t}\cos(kx)$$

Interpretations:

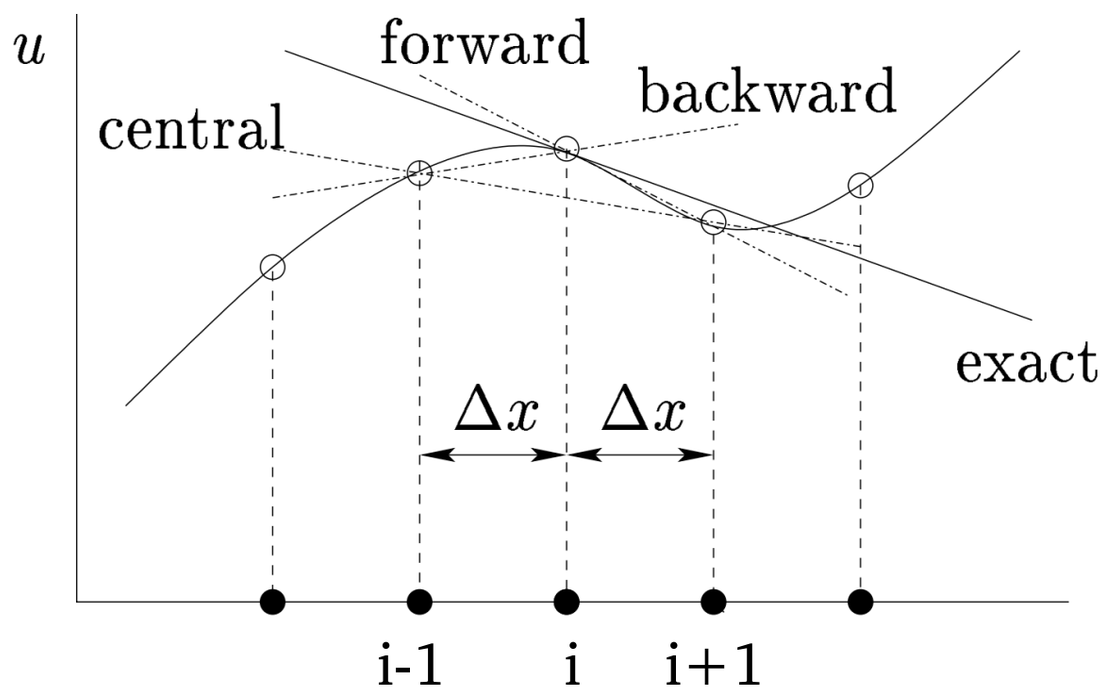
- In this case we end up with a "standing wave" oscillating at each grid point
- Information no longer travel - just oscillations



How to approximate partial derivatives

We've already learned the finite difference method

Geometric interpretation



$$\left(\frac{\partial u}{\partial t}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x} \quad \text{Forward difference}$$

$$\left(\frac{\partial u}{\partial t}\right)_i \approx \frac{u_i - u_{i-1}}{\Delta x} \quad \text{Backward difference}$$

$$\left(\frac{\partial u}{\partial t}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x} \quad \text{Central difference}$$

This is based on the Taylor series expansion $\quad u(x) = \sum_{n=0}^{\infty} \frac{(x - x_i)^n}{n!} \left(\frac{\partial^n u}{\partial x^n}\right)_i$

Let's try two expansions around x_i

T1: $$u_{i+1} = u_i + \Delta x \left(\frac{\partial u}{\partial x}\right)_i + \frac{1}{2}\Delta x^2 \left(\frac{\partial^2 u}{\partial x^2}\right)_i + \frac{1}{6}\Delta x^3 \left(\frac{\partial^3 u}{\partial x^3}\right)_i + \ldots$$

T2: $$u_{i-1} = u_i - \Delta x \left(\frac{\partial u}{\partial x}\right)_i + \frac{1}{2}\Delta x^2 \left(\frac{\partial^2 u}{\partial x^2}\right)_i - \frac{1}{6}\Delta x^3 \left(\frac{\partial^3 u}{\partial x^3}\right)_i + \ldots$$

Recall the Linear Advection Equation

$$\left(\frac{\partial u}{\partial t}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x}$$

Forward difference

$$\frac{\partial Q}{\partial t} + u_0 \frac{\partial Q}{\partial x} = 0$$

Forward difference

Forward difference

Euler Time-Stepping

$$\left.\frac{\partial Q}{\partial t}\right|_{t=t_n} = \frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \mathcal{O}(\Delta t)$$

$$\left.\frac{\partial Q}{\partial x}\right|_{x=x_i} = \frac{Q_{i+1}^n - Q_i^n}{\Delta x} + \mathcal{O}(\Delta x)$$

Combine the two numerical derivatives

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} = u_0 \frac{Q_{i+1}^n - Q_i^n}{\Delta x} + \mathcal{O}(\Delta x) + \mathcal{O}(\Delta t) \longrightarrow$$

Forward Euler method

$$\boxed{Q_i^{n+1} \approx \underbrace{Q_i^n + \frac{u_0 \Delta t}{\Delta x}(Q_{i+1}^n - Q_i^n)}_{\text{Values on step n (known)}}}$$

If we know all the Q values at time t = t_n, then we can calculate the Q values at t = t_n+1. This is known as an explicit scheme of first-order accuracy

Can we do this to our MHD equations?

A first-order Scheme

1-D MHD equations

$$\frac{\partial}{\partial t}\rho = -\frac{\partial}{\partial x}(\rho u_x)$$

$$\frac{\partial u_x}{\partial t} = -u_x\frac{\partial u_x}{\partial x} + \frac{1}{\rho}\left(-\frac{\partial p}{\partial x} - J_z B_y\right)$$

$$\frac{\partial u_y}{\partial t} = -u_x\frac{\partial u_y}{\partial x} + \frac{1}{\rho}J_z B_{x0}$$

$$\frac{\partial B_y}{\partial t} = \frac{\partial E_z}{\partial x}$$

$$E_z = -u_x B_y + u_y B_{x0}$$

$$J_z = \frac{\partial B_y}{\partial x}$$

$$p = \frac{\beta_0}{2}\rho^\gamma$$

Finite-Difference Approximations

$$\rho_i^{n+1} \approx \rho_i^n + \frac{\Delta t}{\Delta x}(u_{x,i+1}^n \rho_{i+1}^n - u_{x,i}^n \rho_i^n)$$

$$u_{x,i}^{n+1} \approx u_{x,i}^n - u_{x,i}\frac{\Delta t}{\Delta x}(u_{x,i+1}^n - u_{x,i}^n) + \frac{\Delta t}{\rho_i^n}\left[\frac{1}{\Delta x}\left(p_{i+1}^n - p_i\right) - J_{z,i}^n B_{y,i}^n\right]$$

$$u_{y,i}^{n+1} \approx u_{y,i}^n - u_{x,i}\frac{\Delta t}{\Delta x}(u_{y,i+1}^n - u_{y,i}^n) + \frac{\Delta t}{\rho_i^n}J_{z,i}^n B_{x0}$$

$$B_y^{n+1} \approx B_y^n + \frac{\Delta t}{\Delta x}(E_{z,i+1}^n - E_{z,i}^n)$$

$$E_{z,i}^n = -u_{x,i}^n B_{y,i}^n + u_{y,i}^n B_{x0}$$

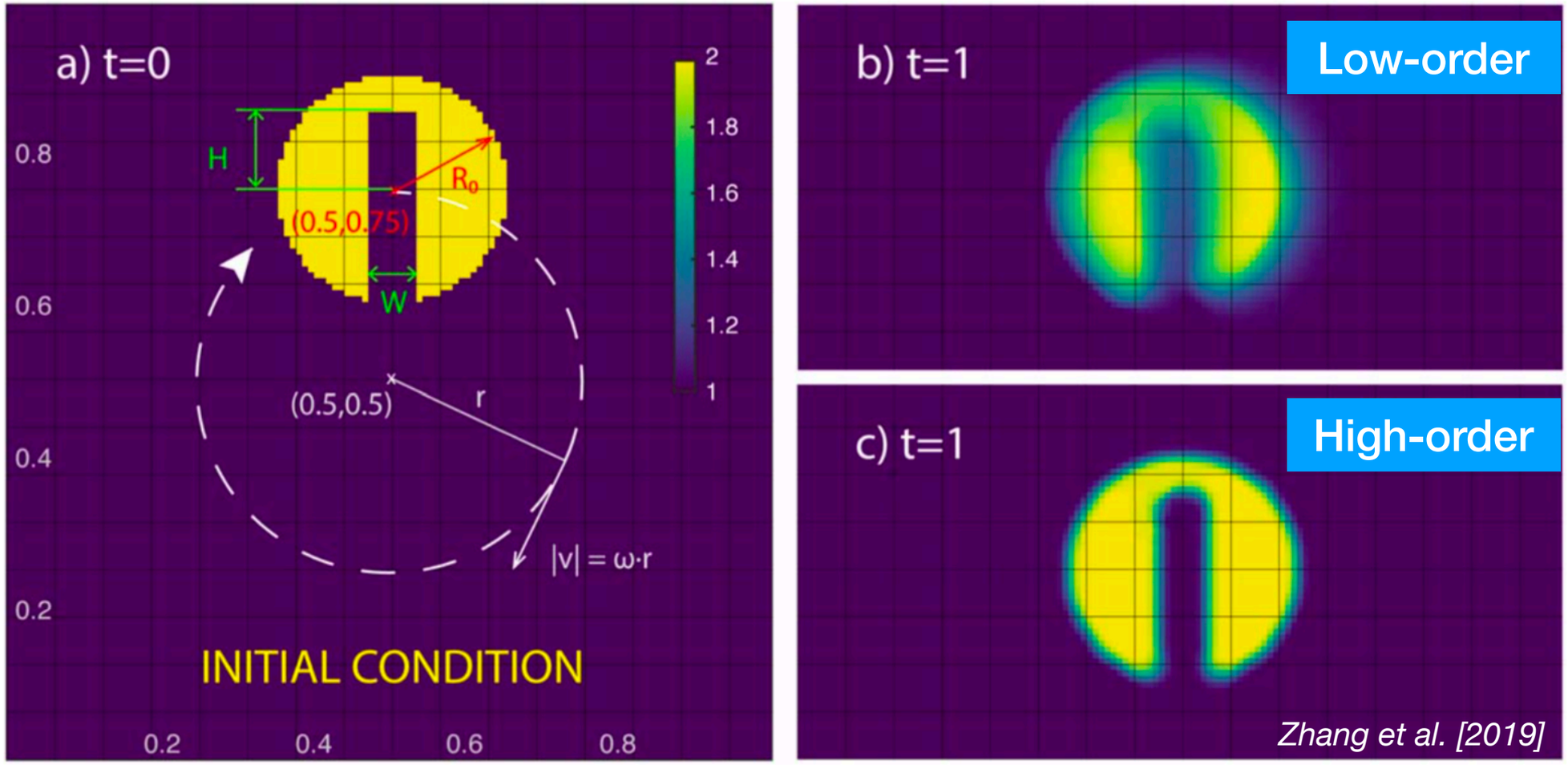
$$J_{z,i}^n = \frac{1}{\Delta x}\left(B_{y,i+1}^n - B_{y,i}^n\right)$$

$$p_{z,i}^n = \frac{\beta_0}{2}(\rho_i^n)^\gamma$$

Why not using First-order schemes

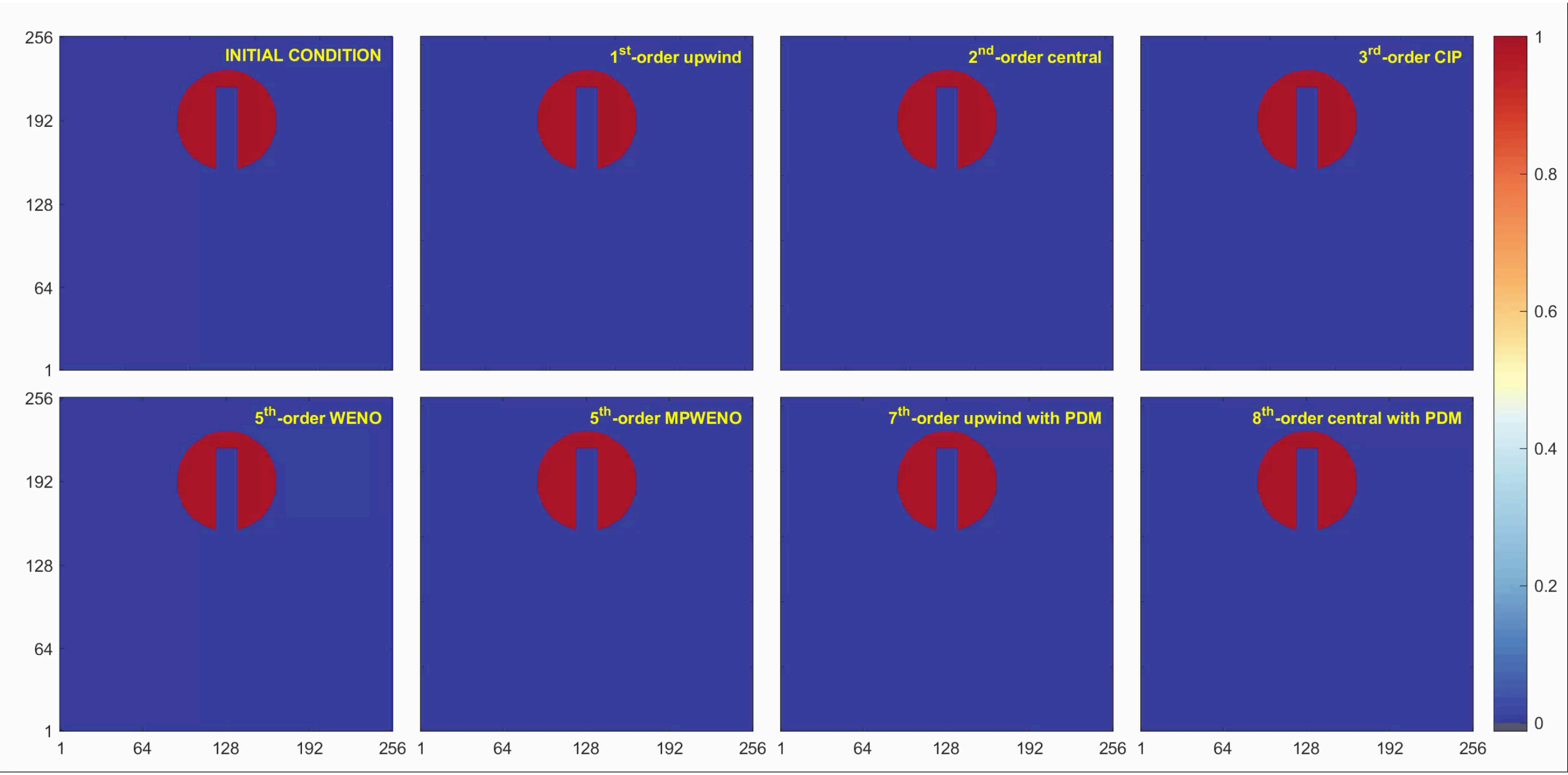
Pros:
- Simple, straightforward to implement
- Usually quite stable

Cons:
- Not accurate - only first order approximations
- Converges at a very slow rate - linearly with grid size



Zhang et al. [2019]

Advection w/ different schemes

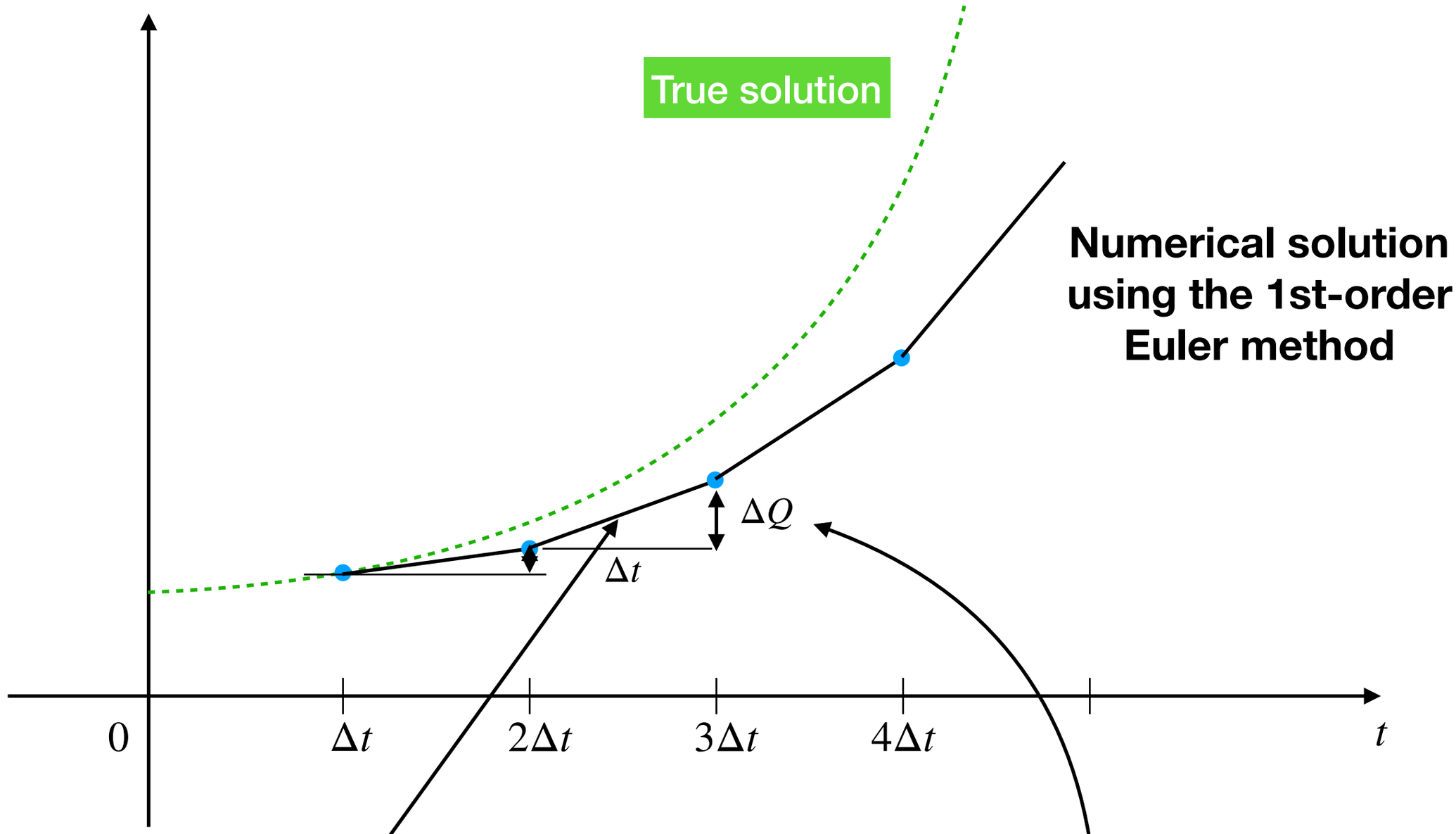


Movie credit: ZZ Li, Earth and Space Sciences@USTC

Note: this is a quite extreme case since the initial density profile is NOT smooth

Requires techniques in TOPIC 3 for proper calculations

Why First-order time stepping sucks



$$\frac{\partial Q}{\partial t} = \frac{Q_i^{n+1} - Q_i^n}{\Delta t} = \boxed{u_0 \frac{\partial Q}{\partial x}} \longrightarrow Q_i^{n+1} = Q_i^n + \boxed{\Delta t \cdot u_0 \frac{\partial Q}{\partial x}}$$

So we need at least second-order schemes

For spatial derivative terms, it is straightforward to extend to a second-order scheme using central difference, e.g.,

$$\frac{\partial}{\partial t}\rho = \underbrace{-\frac{\partial}{\partial x}(\rho u_x)}_{S(\rho)}$$

Spatial operations only

$$\Longrightarrow$$

$$\rho_i^{n+1} \approx \rho_i^n - \frac{\Delta t}{\Delta x}(u_{x,i+1}^n \rho_{i+1}^n - u_{x,i}^n \rho_i^n)$$

1st-order Scheme

$$\rho_i^{n+1} \approx \rho_i^n - \frac{\Delta t}{2\Delta x}(u_{x,i+1}^n \rho_{i+1}^n - u_{x,i-1}^n \rho_{i-1}^n)$$

2nd-order Scheme

For the time derivative, it is not a good practice to use a central difference as the default second-order method, i.e.,

$$\frac{\rho_i^{n+1} - \rho_i^{n-1}}{2\Delta t} = \frac{1}{2\Delta x}(u_{x,i+1}^n \rho_{i+1}^n - u_{x,i-1}^n \rho_{i-1}^n) \longrightarrow \rho_i^{n+1} = \rho_i^{n-1} - \frac{\Delta t}{\Delta x}(u_{x,i+1}^n \rho_{i+1}^n - u_{x,i-1}^n \rho_{i-1}^n)$$

The problem is basically due to the odd-even decoupling:

Simple Leapfrog time stepping algorithms

Let's write the 1st order PDE in terms of f, f here could be mass, velocity, magnetic field:

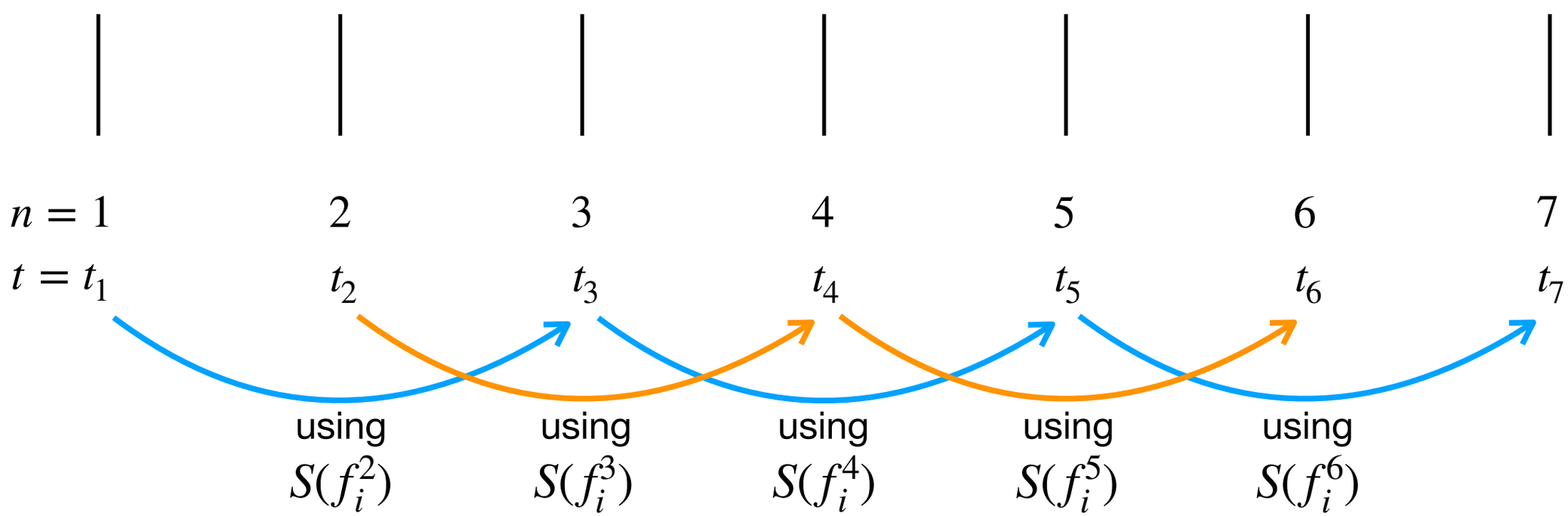
$$\frac{\partial f}{\partial t} = S(f)$$

Here S is the operators on the RHS of the mass, velocity, magnetic equations

To use a second-order central difference for the time derivative, we get

$$f_i^{n+1} = f_i^{n-1} + 2\Delta t S(f_i^n)$$

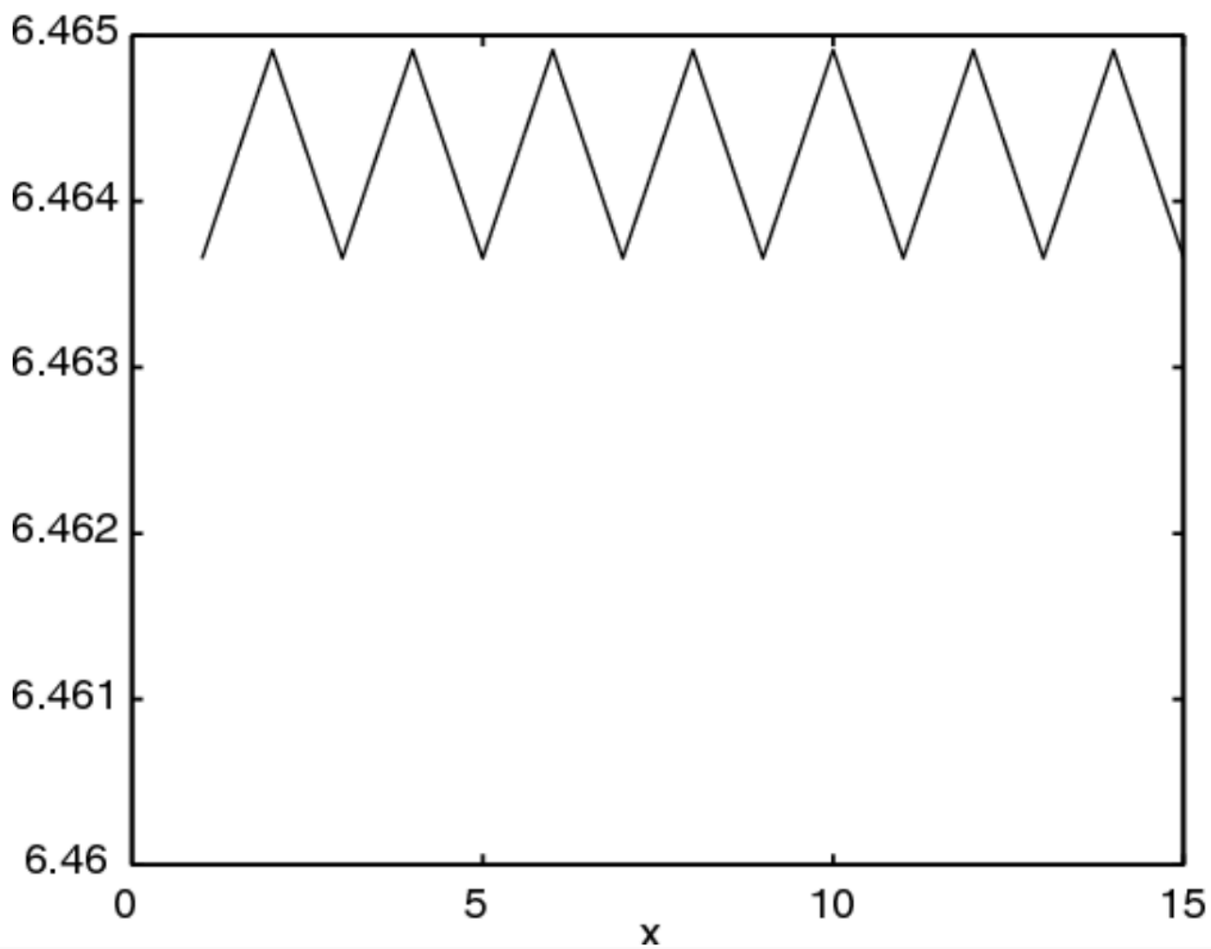
Which means the equations are solved in a "leapfrog" style as follows:



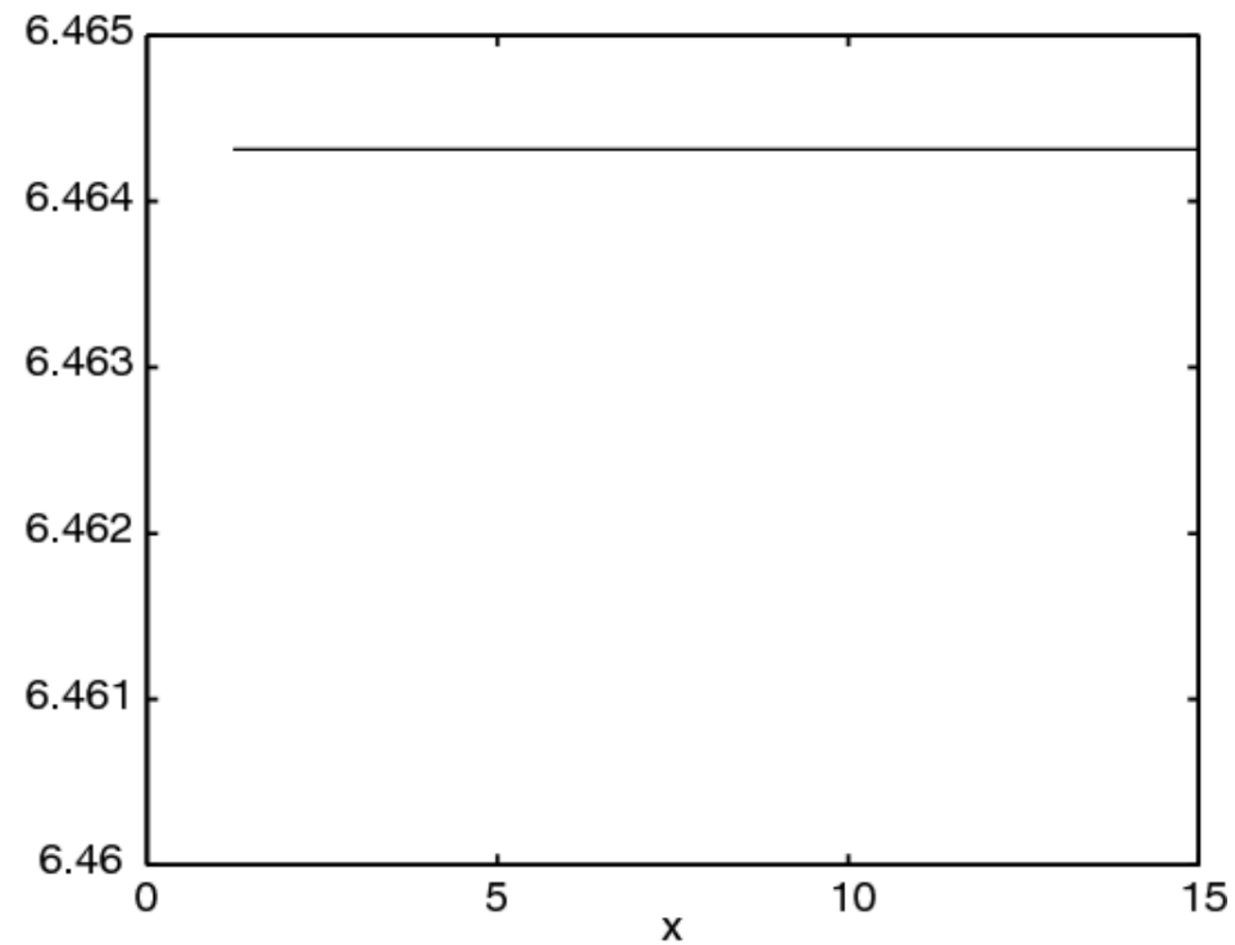
Consequence: odd and even steps decouple and they can drift apart significantly!

Odd-Even Decoupling

Numerical Solution



True Solution



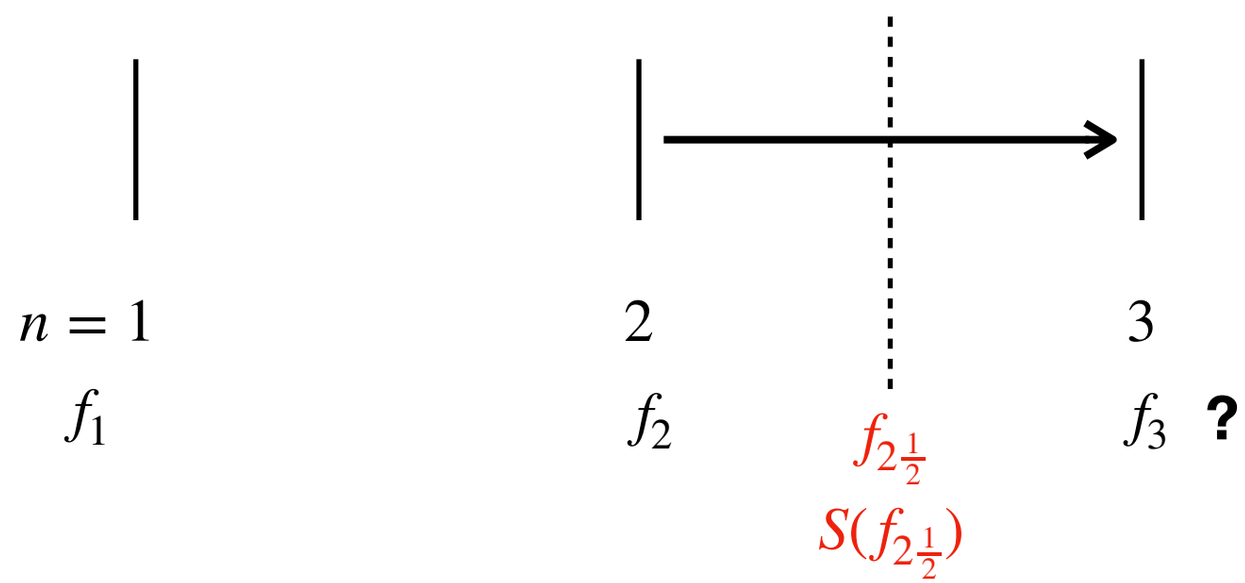
So the “odd” grid cells form a solution that drifts away from the “even” grid cells - each sets of solutions are good, but combined together you get oscillations (ugly)

Leapfrog Trapezoidal time stepping

There are many choices of getting a viable second order time stepping method, we will use a kind of “predictor - corrector” scheme named “*Leapfrog Trapezoidal*” (LT) method in the following discussions, which uses two sup-steps for the time evolution

Assume that f is known at t1 and t2, corresponding to n=1 and n=2, we want to calculate f at t3 using the differential equation

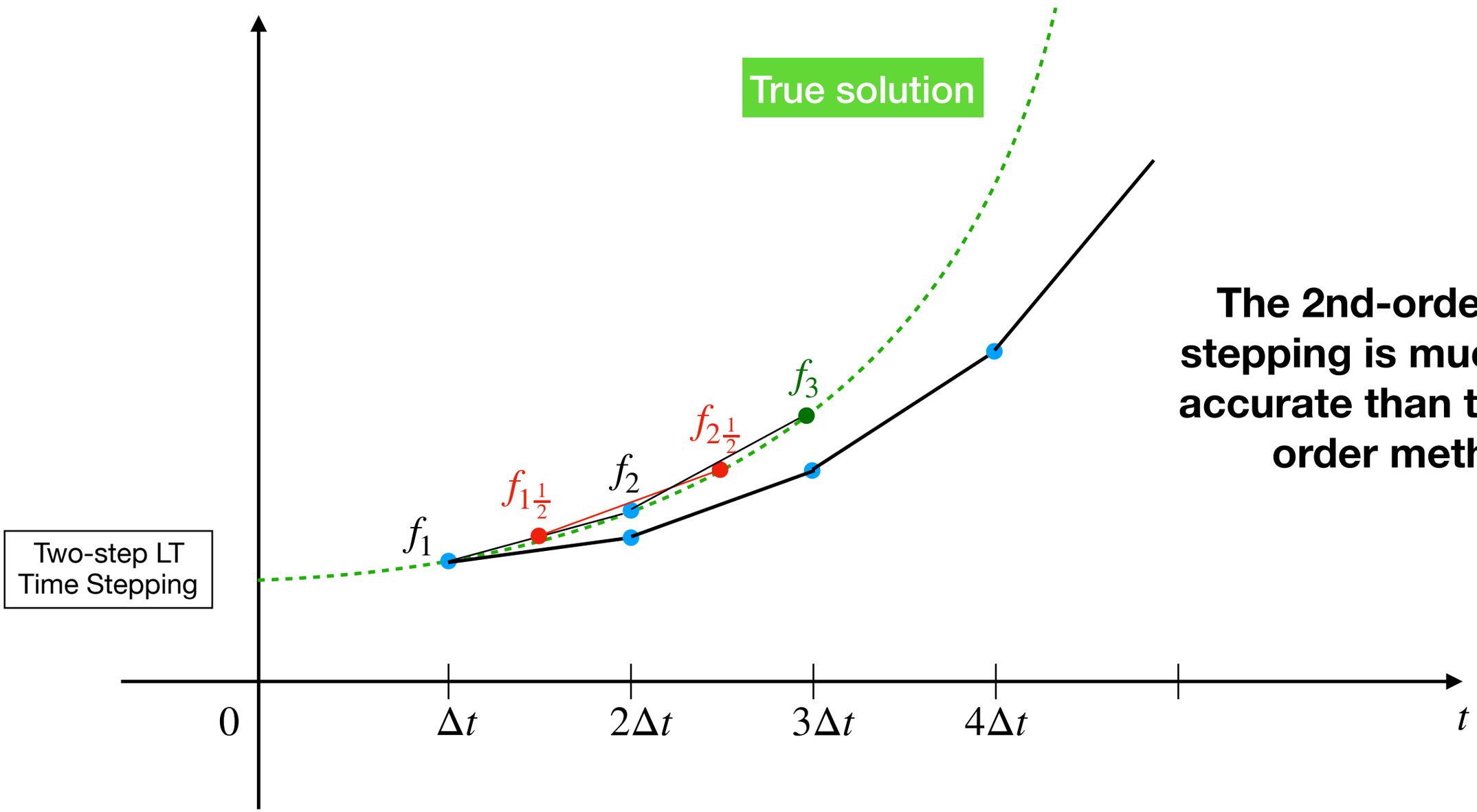
$$\frac{\partial f}{\partial t} = S(f)$$



- Going from f2 to f3 using f2 and S(f2) gives the first order time stepping
- If we know f2.5 and S(f2.5), we get something like a central difference scheme for f3:

$$\frac{f_i^{n+1} - f_i^n}{\Delta t} = S(f_i^{n+\frac{1}{2}}) \quad \Longrightarrow \quad f_i^{n+1} = f_i^n + \Delta t S(f_i^{n+\frac{1}{2}})$$

Interpretation of the LT method



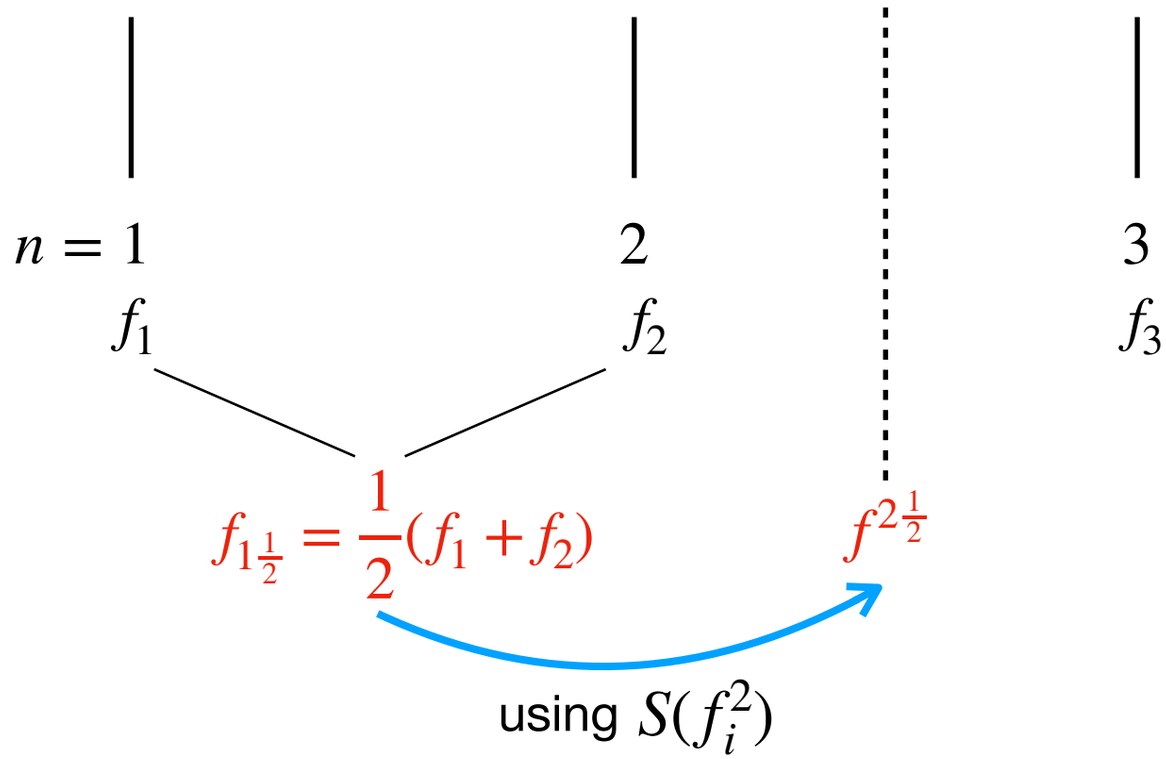
The 2nd-order time stepping is much more accurate than the first-order method

$$f^{n+\frac{1}{2}} = \frac{1}{2}(f^n + f^{n-1}) + \Delta t \cdot S(f^n)$$

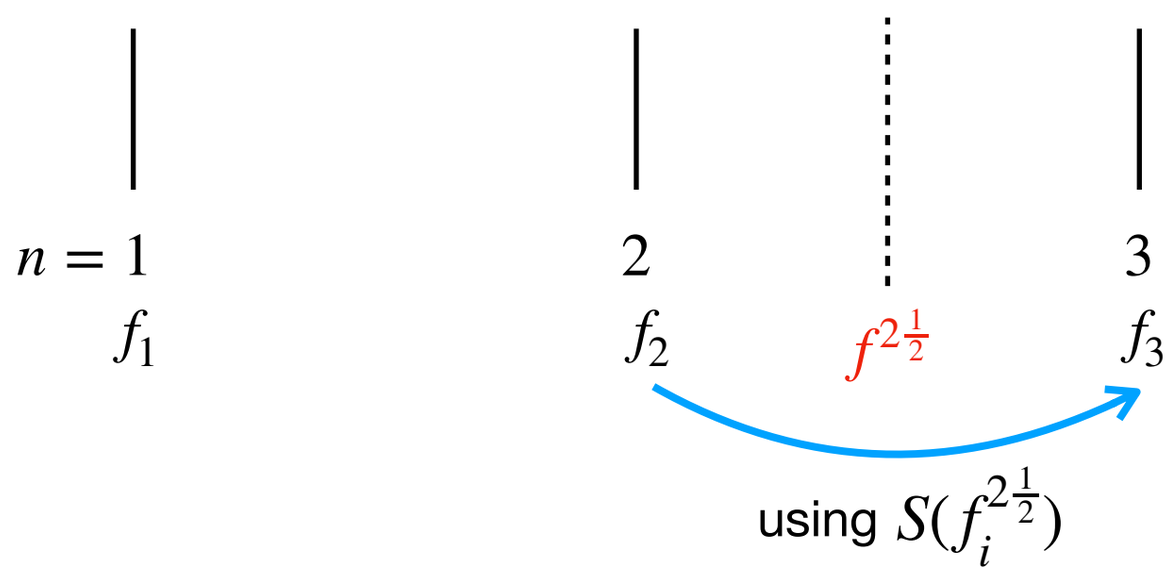
$$f^{n+1} = f^n + \Delta t \cdot S(f^{n+\frac{1}{2}})$$

Leapfrog Trapezoidal time stepping

The calculation of f2.5 is called the ***predictor*** step:



Going from f2 to f3 is called the ***corrector*** step



Leapfrog Trapezoidal time stepping

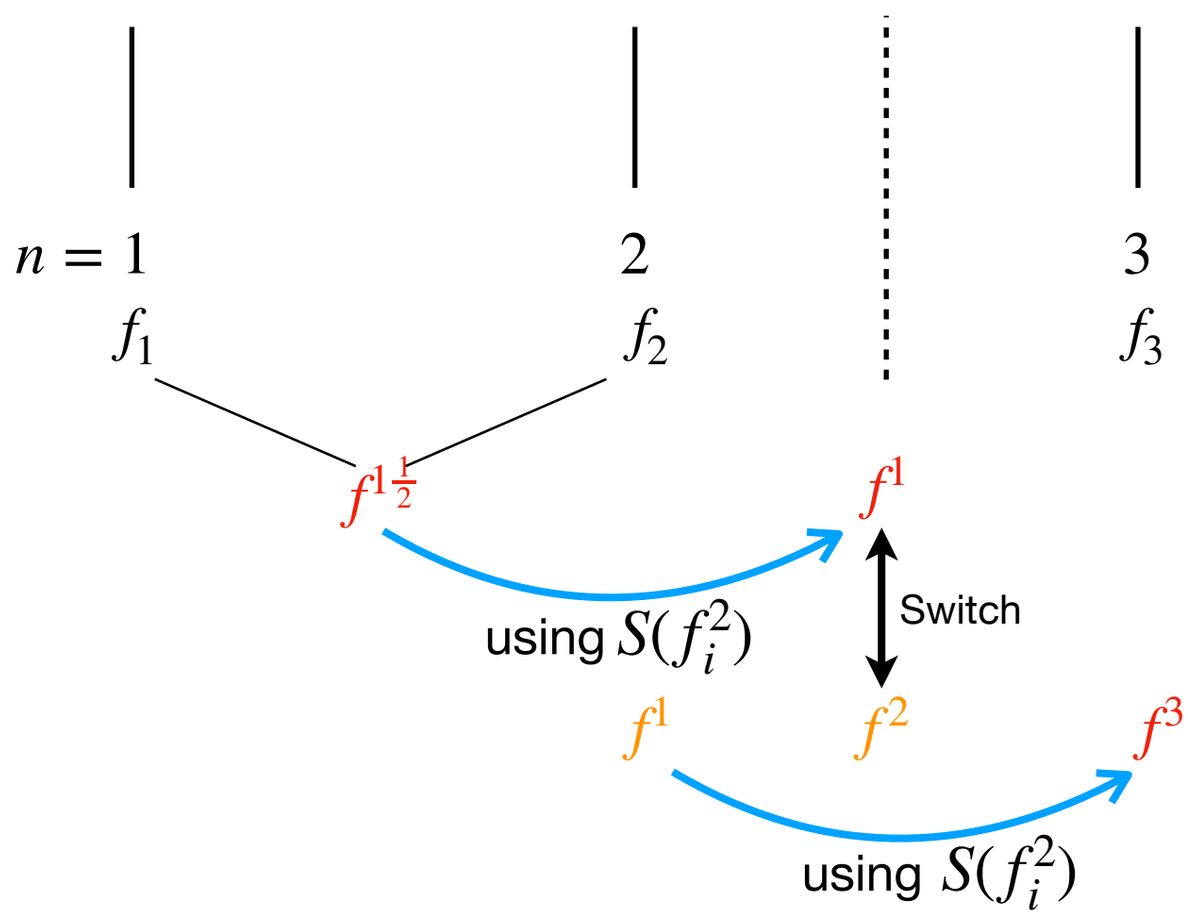
The LT algorithm for time stepping goes like

Two-step LT Time Stepping

$$f^{n+\frac{1}{2}} = \frac{1}{2}(f^n + f^{n-1}) + \Delta t \cdot S(f^n)$$
$$f^{n+1} = f^n + \Delta t \cdot S(f^{n+\frac{1}{2}})$$

$n = 2,3,4,...$

In general we need to store the variables (arrays) at each new time step (and each half time step like f2.5), it could be a waste of resource while the scale of the simulation is enormous. So we do something like this to get around:



Algorithm used in code

STEP 1: $f^1 = \frac{1}{2}(f^1 + f^2) + \Delta t \cdot S(f^2)$

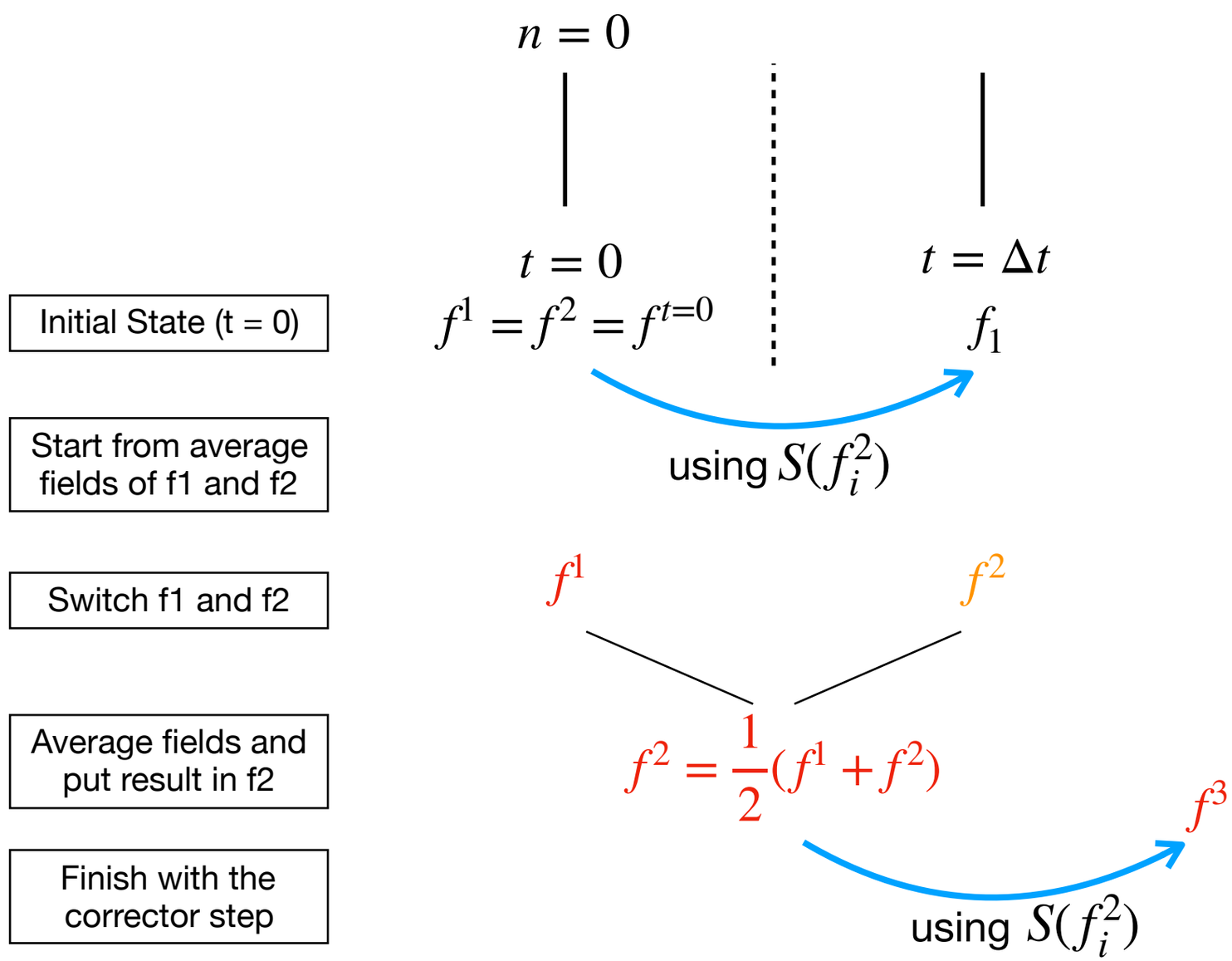
STEP 2: switch f1 and f2

STEP 3: $f^2 = f^1 + \Delta t \cdot S(f^2)$

Dealing with the first time step

For the first time step, we don't have the plasma and field variables at two different times t1 and t2 as illustrated before.

Instead, all the initial plasma and field variables are specified at the same time t=0:



Why is LT second-order accurate?

The answer is again in the ***Taylor's series expansions*** of f and S(f)

First it is very straightforward to show the Leapfrog scheme is 2nd-order accurate in time:

$$f_i^{n+1} = f_i^{n-1} + 2\Delta t S(f_i^n) + \mathcal{O}(\Delta t^2)$$

To evaluate the accuracy of the Leapfrog trapezoidal algorithm, expand

$$S(f) \approx S_0 + S'\Delta f + \frac{1}{2}S''\Delta f^2 + \frac{1}{6}S'''\Delta f^3$$

Where $S_0 = S(f^0)$ $\quad S' = \left.\frac{\partial S}{\partial f}\right|_{f^0}$ $\quad S'' = \left.\frac{\partial^2 S}{\partial f^2}\right|_{f^0}$ and $\Delta f = f - f_0$

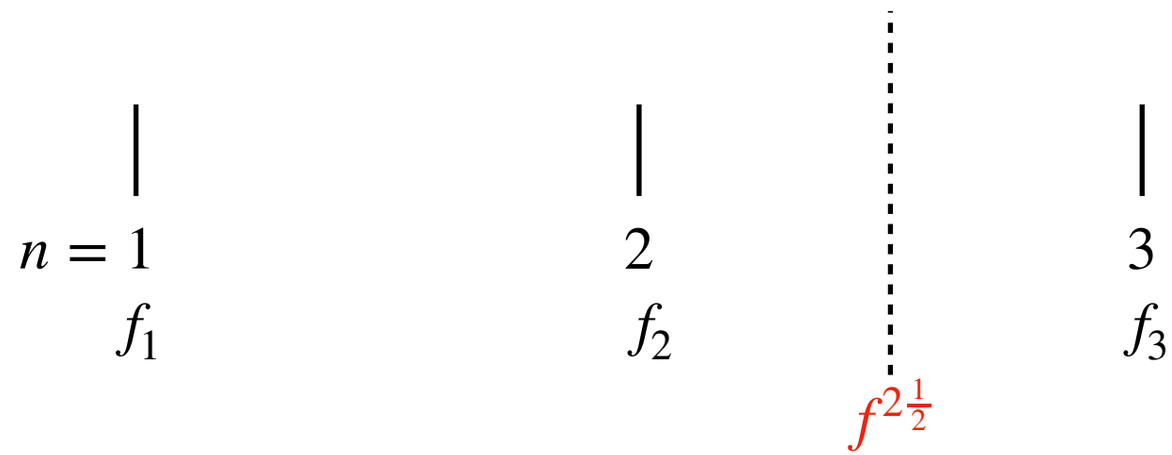
Considering that errors in Δf will be proportional to some power of delta t, the biggest problem in the evaluation of S (largest error) will come from the S'df term

So let's assume S goes like $S(f) \approx S_0 + S'\Delta f$

Now the problem becomes, if we know f1 and f2 exactly, what is the error in f3 using the LT algorithm for time evolution?

We need to use ***Taylor's series expansions*** for f(t) now

Why is LT second-order accurate?



use ***Taylor's series expansions*** for f(t):

$$f^1 \approx f^2 - f^{2'}\Delta t + \frac{1}{2}f^{2''}\Delta t^2 - \frac{1}{6}f^{2'''}\Delta t^3 \qquad f^{2\frac{1}{2}} \approx f^2 + f^{2'}\frac{\Delta t}{2} + \frac{1}{2}f^{2''}\left(\frac{\Delta t}{2}\right)^2 + \frac{1}{6}f^{2'''}\left(\frac{\Delta t}{2}\right)^3$$

Now get $f^* = \frac{1}{2}(f^1 + f^2) + \Delta t \cdot S^2$

$$= \frac{1}{2}(f^2 + \underbrace{f^2 - f^{2'}\Delta t + \frac{1}{2}f^{2''}\Delta t^2 - \frac{1}{6}f^{2'''}\Delta t^3 + \dots}_{f^1}) + \Delta t \cdot \underbrace{S(f^2)}_{f^{2'}}$$

Because we know f2 exactly!

$$= f^2 + f^{2'}\frac{\Delta t}{2} + \frac{1}{4}f^{2''}\Delta t^2 + \dots$$

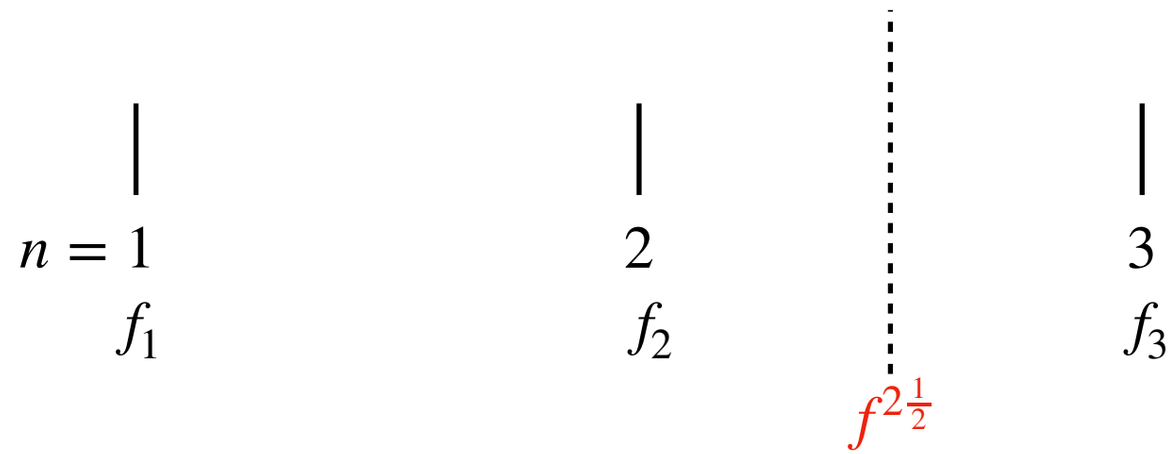
compare to

$$f^{2\frac{1}{2}} \approx f^2 + f^{2'}\frac{\Delta t}{2} + \frac{1}{8}f^{2''}\Delta t^2 + \dots$$

Now we can use Taylor series expansions to evaluate the accuracy of the solution at f3

We see that $\boxed{f^* = f^{2\frac{1}{2}} + \frac{1}{8}f^{2''}\Delta t^2 + \dots}$

Why is LT second-order accurate?



Now do a Taylor expansion of f about f2.5:

$$f^2 = f^{2\frac{1}{2}} - f^{2\frac{1}{2}'}\frac{\Delta t}{2} + \frac{1}{2}f^{2\frac{1}{2}''}\left(\frac{\Delta t}{2}\right)^2 - \frac{1}{6}f^{2\frac{1}{2}'''}\left(\frac{\Delta t}{2}\right)^3 + \dots$$

$$f^3 = f^{2\frac{1}{2}} + f^{2\frac{1}{2}'}\frac{\Delta t}{2} + \frac{1}{2}f^{2\frac{1}{2}''}\left(\frac{\Delta t}{2}\right)^2 + \frac{1}{6}f^{2\frac{1}{2}'''}\left(\frac{\Delta t}{2}\right)^3 + \dots$$

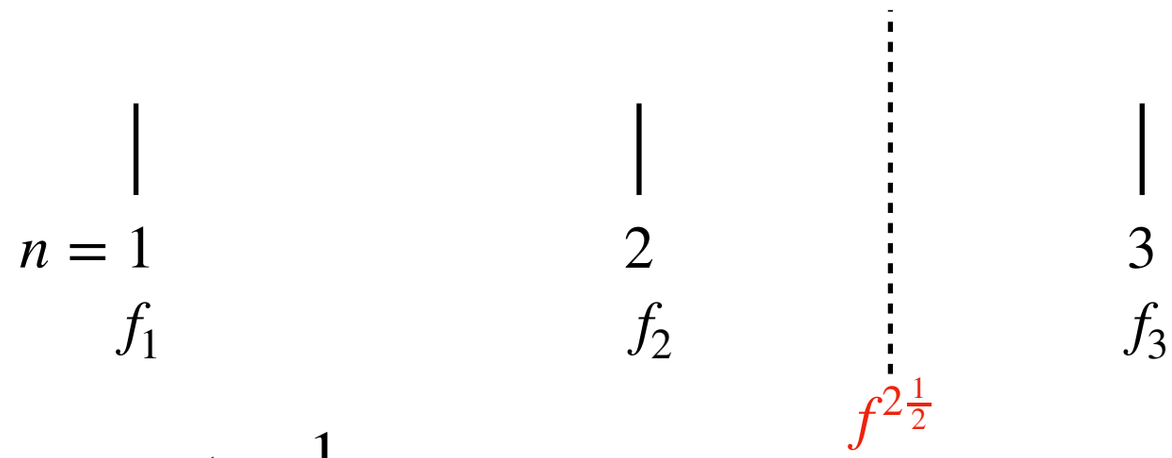
Subtract the two series, we get

$$f^3 - f^2 = f^{2\frac{1}{2}'}\Delta t + \frac{1}{24}f^{2\frac{1}{2}'''}\Delta t^3 + \dots$$

If we know f2.5 exactly, then we also know S(f2.5) exactly and the error in f3 will be delta t^2 because

$$\frac{f^3 - f^2}{\Delta t} = f^{2\frac{1}{2}'} + \frac{1}{24}f^{2\frac{1}{2}'''}\Delta t^2 + \dots \longrightarrow \left.\frac{\partial f}{\partial t}\right|_{2\frac{1}{2}} = S(f^{2\frac{1}{2}}) + \mathcal{O}(\Delta t^2)$$

Why is LT second-order accurate?



But we only know $f^* = f^{2\frac{1}{2}} + \frac{1}{8} f^{2''} \Delta t^2 + \ldots$

Now use $\frac{f^3 - f^2}{\Delta t} = S(f^*) \approx S(f^{2\frac{1}{2}}) + S'(f^* - f^{2\frac{1}{2}}) = S(f^{2\frac{1}{2}}) + S' \frac{1}{8} f^{2''} \Delta t^2 + \ldots$

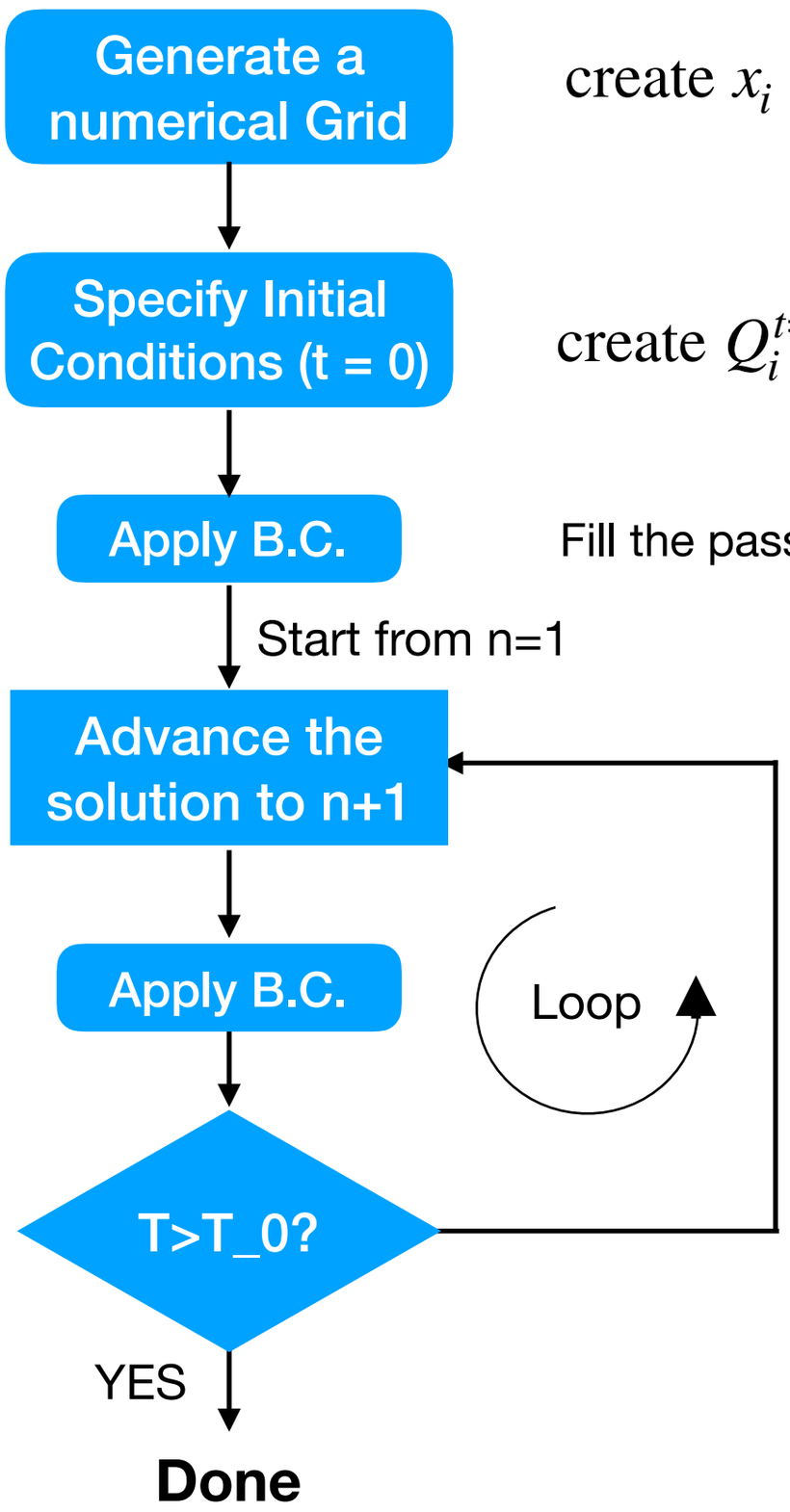
Which means the Leapfrog trapezoidal time stepping is written as

$$\frac{f^3 - f^2}{\Delta t} = S(f^{2\frac{1}{2}}) + \underbrace{S' \frac{1}{8} f^{2''} \Delta t^2}_{\text{Leading error}} + \ldots$$

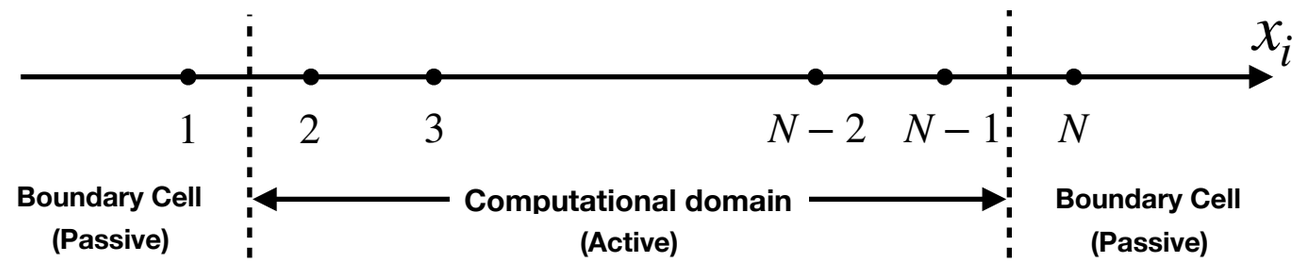
Compared to the central difference around f2.5:

$$\frac{f^3 - f^2}{\Delta t} = f^{2\frac{1}{2}'} + \underbrace{\frac{1}{24} f^{2\frac{1}{2}'''} \Delta t^2}_{\text{Order of error}} + \ldots$$

Build the code w/ predictor-corrector steps



create x_i



create $Q_i^{t=0}$ $\quad$ e.g., $\rho_i^{t=0} = \text{something}$

Fill the passive cells for derivatives at the end points

$$f^{n+\frac{1}{2}} = \frac{1}{2}(f^n + f^{n-1}) + \Delta t \cdot S(f^n)$$
$$f^{n+1} = f^n + \Delta t \cdot S(f^{n+\frac{1}{2}})$$

Two-step LT Time Stepping

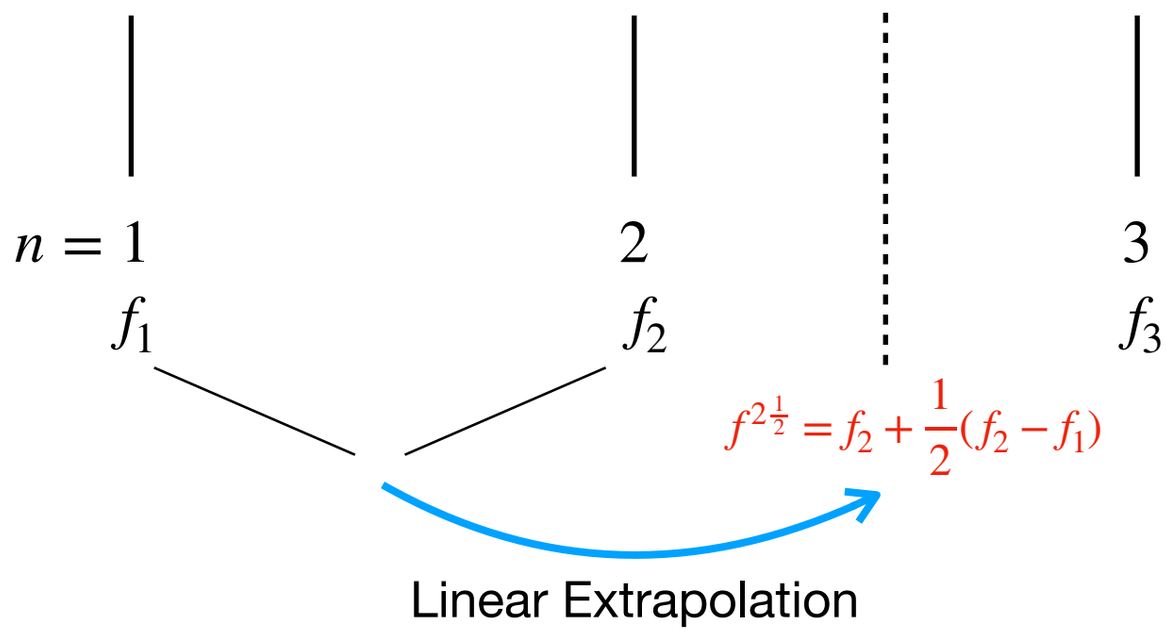
Fill the passive cells for derivatives at the end points

Determine whether reaching the end time (or other criteria)

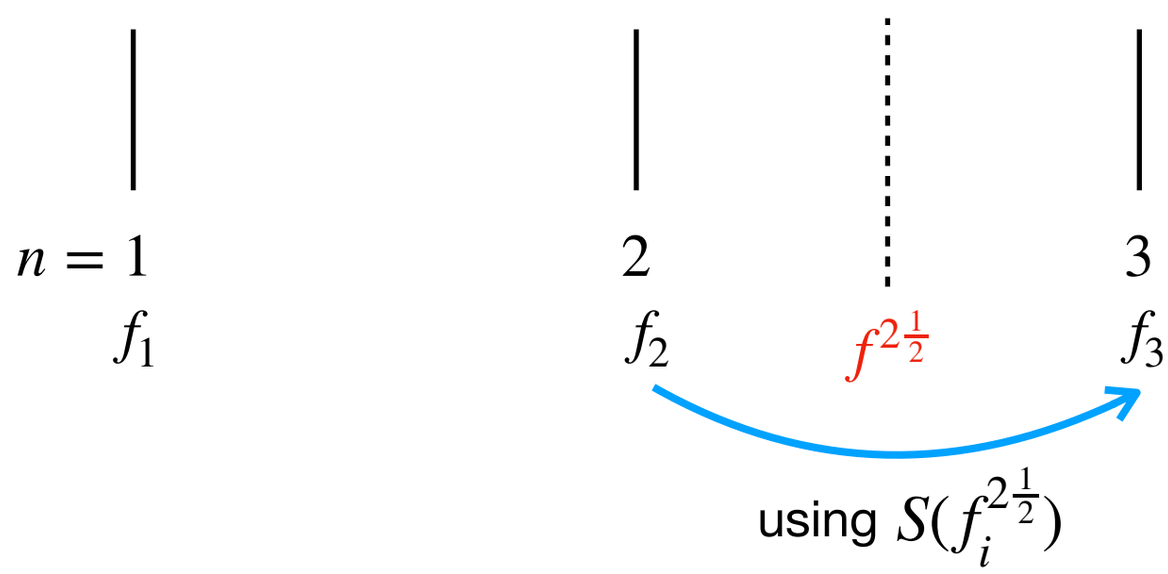
One-Step predictor-corrector method

Adams-Bashforth Method

The calculation of f2.5 is the ***predictor*** step - which is a linear extrapolation



Going from f2 to f3 is the same ***corrector*** step as leapfrog trapezoid method



Physics

Energy in an MHD code (kinetic)

Before talking about the boundary conditions, let's first take a look at the energy conservation properties of MHD codes (why is energy important?)

Start from the velocity equation:

$$\rho\frac{d\mathbf{u}}{dt} = -\nabla p + \mathbf{J}\times\mathbf{B} \xrightarrow{\cdot\,\mathbf{u}} \rho\frac{d}{dt}\left(\frac{1}{2}\mathbf{u}\cdot\mathbf{u}\right) = -\mathbf{u}\cdot\nabla p + \mathbf{u}\cdot\mathbf{J}\times\mathbf{B}$$

$$\frac{1}{2}\mathbf{u}\cdot\mathbf{u} \rightarrow \frac{u^2}{2}$$

Integrate over volume

$$\int_V \rho\frac{d}{dt}\left(\frac{1}{2}u^2\right)dV = \int_V \left(-\mathbf{u}\cdot\nabla p + \underline{\mathbf{u}\cdot\mathbf{J}\times\mathbf{B}}\right)dV$$

$$\mathbf{u}\cdot\mathbf{J}\times\mathbf{B} \rightarrow \mathbf{J}\cdot(\mathbf{B}\times\mathbf{u}) = \mathbf{J}\cdot\mathbf{E}$$

Use integrate by part:

$$\int_V \frac{\partial(\rho f)}{\partial t}dV = \int_V \left(\rho\frac{\partial f}{\partial t} + f\frac{\partial\rho}{\partial t}\right)dV \xrightarrow{\frac{\partial}{\partial t}\rho = -\nabla\cdot\rho\mathbf{u}} = \int_V \left(\rho\frac{\partial f}{\partial t} - f\nabla\cdot\rho\mathbf{u}\right)dV$$

$$\xrightarrow{\nabla\cdot(f\mathbf{A}) = f\nabla\cdot\mathbf{A} + A\cdot\nabla f} = \int_V \left[\rho\frac{\partial f}{\partial t} - \nabla\cdot\rho\mathbf{u}f + \rho\mathbf{u}\cdot\nabla f\right]dV = \int_V \left[\rho\frac{df}{dt} - \nabla\cdot(\rho\mathbf{u}f)\right]dV$$

So

$$\boxed{\int_V \frac{d}{dt}\left(\frac{1}{2}\rho u^2\right)dV = \int_V \left[-\nabla\cdot\left(\frac{1}{2}\rho u^2\mathbf{u}\right) - \mathbf{u}\cdot\nabla p + \mathbf{J}\cdot\mathbf{E}\right]dV}$$

Kinetic Energy Equation

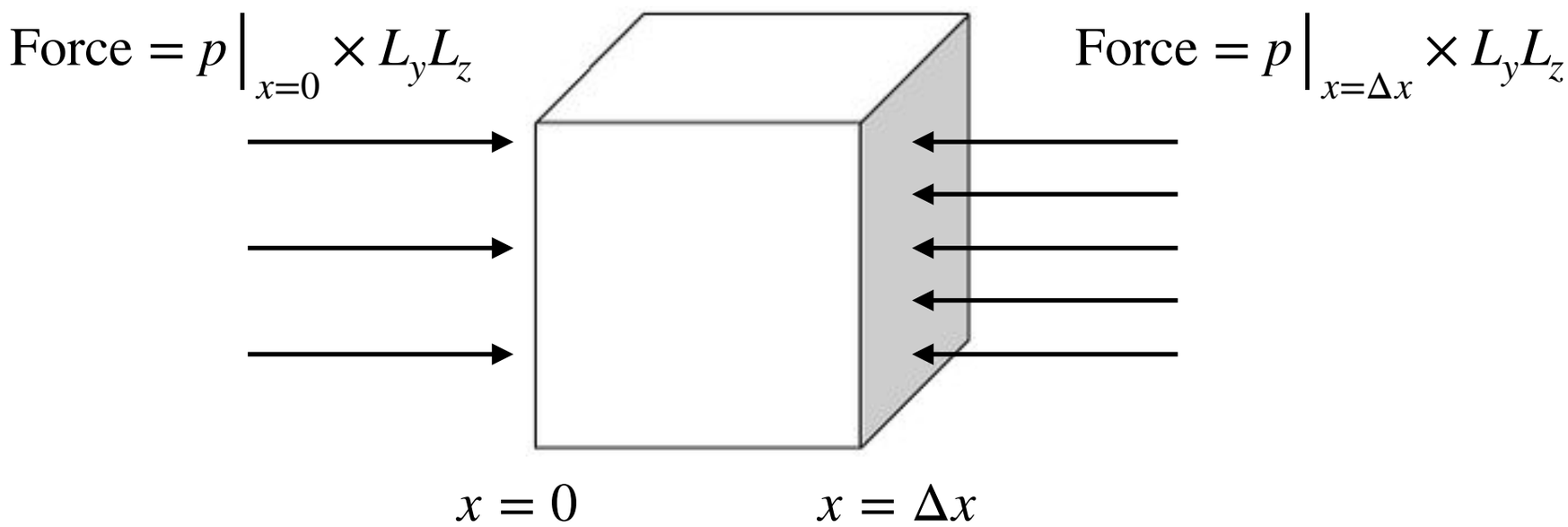
Physics

$-\nabla p$ Interpretation

Based on the definition of pressure $p = \dfrac{\text{Force}}{\text{Area}}$

Consider the force exerted by the external plasma on a volume element $V = \Delta x \times L_y \times L_z$

$$Vol = \Delta x L_y L_z$$



$$\text{Total Force} = \left(p\big|_{x=0} - p\big|_{x=\Delta x}\right) \times L_y L_z \xrightarrow{\text{Force per volume}} \frac{\left(p\big|_{x=0} - p\big|_{x=\Delta x}\right) \times L_y L_z}{\Delta x L_y L_z} \xrightarrow{V \to 0} \sim -\frac{\partial p}{\partial x}\hat{\mathbf{x}} = -\nabla p$$

Then $-\mathbf{u} \cdot \nabla p = \dfrac{d\mathbf{x}_{\text{fluid}}}{dt} \cdot \dfrac{\text{Force}}{\text{volume}} \sim \dfrac{\text{Work}}{\text{time} \cdot \text{volume}}$ Acting on the fluid element

This is why the $-\mathbf{u} \cdot \nabla p$ term converts kinetic energy to internal energy, or vice versa

Physics

Energy in an MHD code (internal)

Start from the adiabatic pressure equation:

$$p = \frac{\beta_0}{2}\rho^\gamma \xrightarrow{\frac{\partial}{\partial t}} \frac{\partial p}{\partial t} = \frac{\beta_0}{2}\gamma\rho^{\gamma-1}\frac{\partial \rho}{\partial t} \xrightarrow{\frac{\partial}{\partial t}\rho = -\nabla\cdot\rho\mathbf{u}} \frac{\partial p}{\partial t} = -\frac{\beta_0}{2}\gamma\rho^{\gamma-1}\nabla\cdot\rho\mathbf{u}$$

Integrate over volume

$$\int_V \frac{\partial p}{\partial t}dV = \int_v \left[-\frac{\beta_0}{2}\gamma\rho^{\gamma-1}\nabla\cdot(\rho\mathbf{u})\right]dV \xrightarrow{\nabla\cdot(f\mathbf{A}) = f\nabla\cdot\mathbf{A} + A\cdot\nabla f}$$

$$= \int_v \left[-\nabla\cdot(\frac{\beta_0}{2}\gamma\rho^{\gamma-1}\rho\mathbf{u}) + \rho\mathbf{u}\cdot\nabla(\frac{\beta_0}{2}\gamma\rho^{\gamma-1})\right]dV$$

$$= \int_v \left[-\nabla\cdot(\frac{\beta_0}{2}\gamma\rho^{\gamma}\mathbf{u}) + \underline{\rho\frac{\beta_0}{2}\gamma(\gamma-1)\rho^{\gamma-2}\mathbf{u}\cdot\nabla\rho}\right]dV$$

On the other hand

$$p = \frac{\beta_0}{2}\rho^\gamma \xrightarrow{\nabla} \nabla p = \nabla\left(\frac{\beta_0}{2}\rho^\gamma\right) = \frac{\beta_0}{2}\gamma\rho^{\gamma-1}\nabla\rho$$

$$\rho e \equiv \frac{p}{\gamma-1}$$ Internal Energy

We get

$$\int_V \frac{\partial p}{\partial t}dV = \int_V \left[-\nabla\cdot(\gamma p\mathbf{u}) + (\gamma-1)\mathbf{u}\cdot\nabla p\right]dV$$

Or the internal energy equation:

$$\boxed{\int_V \frac{\partial}{\partial t}\left(\frac{p}{\gamma-1}\right)dV = \int_V \left[-\nabla\cdot(\frac{\gamma p}{\gamma-1}\mathbf{u}) + \mathbf{u}\cdot\nabla p\right]dV}$$

Physics

Energy in an MHD code (magnetic)

The magnetic energy equation is basically the Poynting theorem applied to the plasma:

$$\frac{\partial \mathbf{B}}{\partial t} = -\nabla \times \mathbf{E} \xrightarrow{\cdot\,\mathbf{B}} \mathbf{B}\cdot\frac{\partial \mathbf{B}}{\partial t} = -\mathbf{B}\cdot\nabla\times\mathbf{E} \xrightarrow{\nabla\cdot(\mathbf{E}\times\mathbf{B}) = \mathbf{B}\cdot\nabla\times\mathbf{E} - \mathbf{E}\cdot\nabla\times\mathbf{B}}$$

$$\longrightarrow \frac{\partial}{\partial t}\left(\frac{1}{2}B^2\right) = -\mathbf{E}\cdot\nabla\times\mathbf{B} - \nabla\cdot(\mathbf{E}\times\mathbf{B}) \xrightarrow[\mathbf{J} = \nabla\times\mathbf{B}]{\text{Ampere's law}}$$

$$= -\mathbf{E}\cdot\mathbf{J} - \nabla\cdot(\mathbf{E}\times\mathbf{B})$$

So the volume-integrated magnetic energy equation is written as

Magnetic Energy

$$\int_V \frac{\partial}{\partial t}\left(\frac{1}{2}B^2\right) dV = \int_V \left[-\nabla\cdot(\mathbf{E}\times\mathbf{B}) - \mathbf{E}\cdot\mathbf{J}\right] dV$$

Converts magnetic energy to/from kinetic energy

And

Kinetic Energy

$$\int_V \frac{d}{dt}\left(\frac{1}{2}\rho u^2\right) dV = \int_V \left[-\nabla\cdot\left(\frac{1}{2}\rho u^2\mathbf{u}\right) - \mathbf{u}\cdot\nabla p + \mathbf{J}\cdot\mathbf{E}\right] dV$$

Converts internal energy to/from kinetic energy

Internal Energy

$$\int_V \frac{\partial}{\partial t}\left(\frac{p}{\gamma-1}\right) dV = \int_V \left[-\nabla\cdot\left(\frac{\gamma p}{\gamma-1}\mathbf{u}\right) + \mathbf{u}\cdot\nabla p\right] dV$$

Physics

Energy in an MHD code (total)

Adding the three energy equations together

$$\int_V \frac{d}{dt}\left(\frac{1}{2}\rho u^2 + \frac{p}{\gamma-1} + \frac{1}{2}B^2\right) dV = \int_V \left[-\nabla\cdot\left(\frac{1}{2}\rho u^2\mathbf{u}\right) - \nabla\cdot\left(\frac{\gamma p}{\gamma-1}\mathbf{u}\right) - \nabla\cdot(\mathbf{E}\times\mathbf{B})\right] dV$$

Total Energy

$$= -\int_V \nabla\cdot\left[\left(\frac{1}{2}\rho u^2\mathbf{u}\right) + \left(\frac{\gamma p}{\gamma-1}\mathbf{u}\right) + (\mathbf{E}\times\mathbf{B})\right] dV$$

$$\xrightarrow[\int_V \nabla\cdot A dV = \oint_S \mathbf{A}\cdot d\mathbf{S}]{\text{Divergence Theorem}} \quad = -\oint_S d\mathbf{S}\cdot\left[\left(\frac{1}{2}\rho u^2\mathbf{u}\right) + \left(\frac{\gamma p}{\gamma-1}\mathbf{u}\right) + (\mathbf{E}\times\mathbf{B})\right]$$

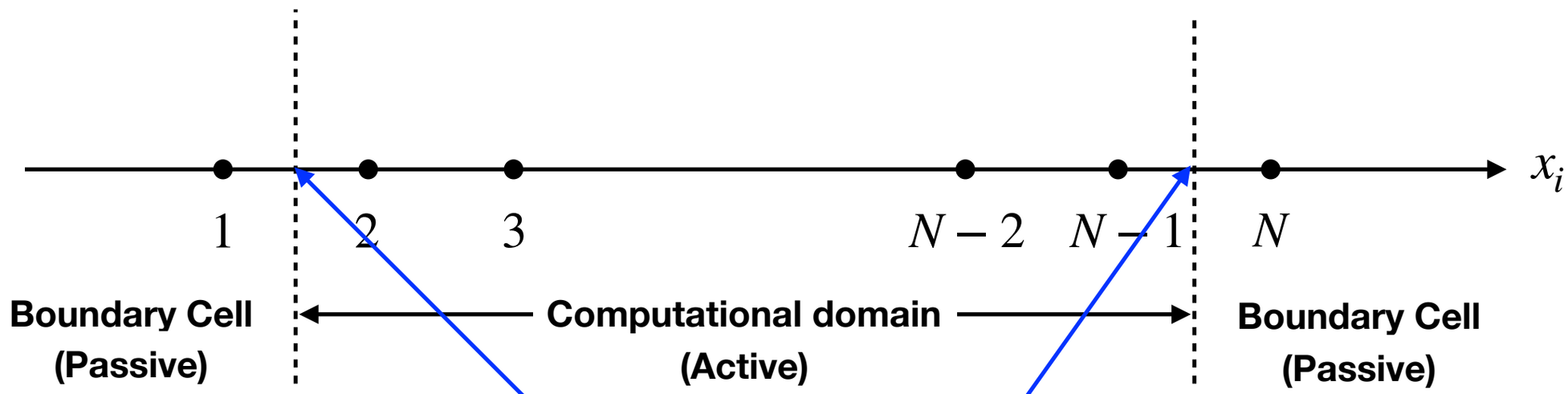
Given certain boundary conditions the surface integral $\oint_S$ may go to zero, leading to total **energy conservation**

$$\mathscr{E}_T = \frac{1}{2}\rho u^2 + \frac{p}{\gamma-1} + \frac{1}{2}B^2$$ Total Energy

And the following term is called the total energy flux

$$\mathbf{F}_{\mathscr{E}} = \frac{1}{2}\rho u^2\mathbf{u} + \frac{\gamma p}{\gamma-1}\mathbf{u} + \mathbf{E}\times\mathbf{B}$$ Total Energy Flux

Energy Conservation in 1-D MHD code



$$\frac{d}{dt}\int_V \mathscr{E}_T dV = -\oint_S d\mathbf{S}\cdot\boxed{\left(\frac{1}{2}\rho u^2\mathbf{u} + \frac{\gamma p}{\gamma-1}\mathbf{u} + \mathbf{E}\times\mathbf{B}\right)}$$

Total Energy Flux

If the total energy flux is zero on the boundary grid cells, then the total energy is conserved

$$\frac{d}{dt}\int_V \mathscr{E}_T dV = 0$$

This is called energy-conserving boundary conditions for MHD

Energy Conservation in 1-D MHD code

Useful Notes

The derivation of energy equations (conservation) include the following equations

$$\nabla \cdot \left(\frac{1}{2} \rho u^2 \mathbf{u} \right) = \rho \mathbf{u} \cdot \nabla \left(\frac{1}{2} u^2 \right) + \frac{1}{2} u^2 \nabla \cdot (\rho \mathbf{u})$$

$$\nabla \cdot \left(\frac{\beta_0}{2} \gamma \rho^\gamma \mathbf{u} \right) = \rho \mathbf{u} \cdot \nabla \left(\frac{\beta_0}{2} \gamma \rho^{\gamma-1} \right) + \frac{\beta_0}{2} \gamma \rho^{\gamma-1} \nabla \cdot (\rho \mathbf{u})$$

And

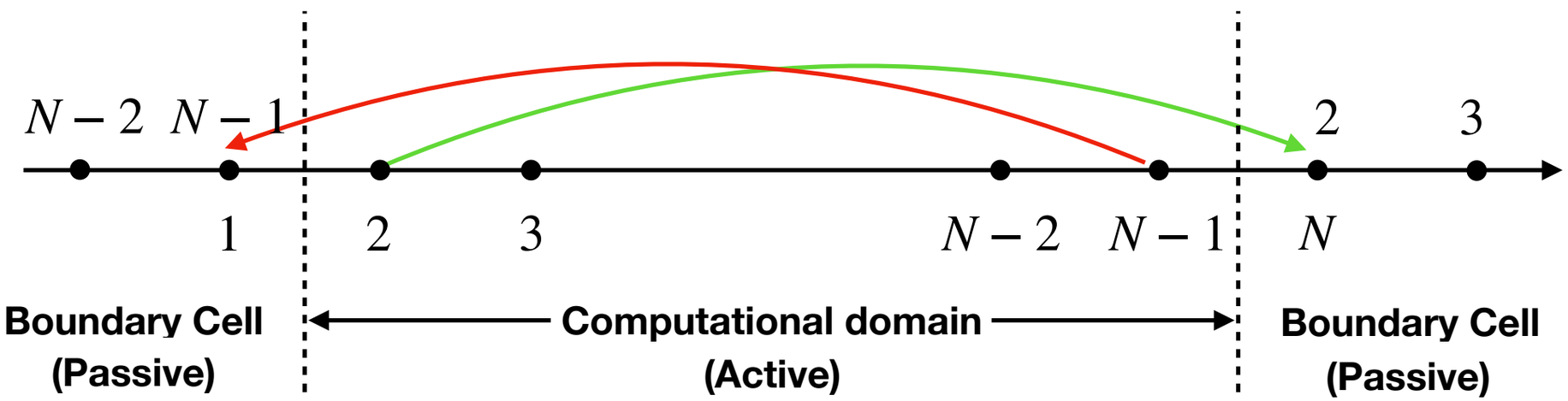
$$\nabla \cdot (\mathbf{E} \times \mathbf{B}) = \mathbf{B} \cdot \nabla \times \mathbf{E} - \mathbf{E} \cdot \nabla \times \mathbf{B}$$

These equations are of the form $\frac{\partial}{\partial x}(ab) = b \frac{\partial a}{\partial x} + a \frac{\partial b}{\partial x}$

This relation is not exactly in the finite difference form, and it is only accurate to the extent that the functions a and b are smooth on the grid (scale lengths >> delta x)

Boundary Conditions in 1-D MHD code

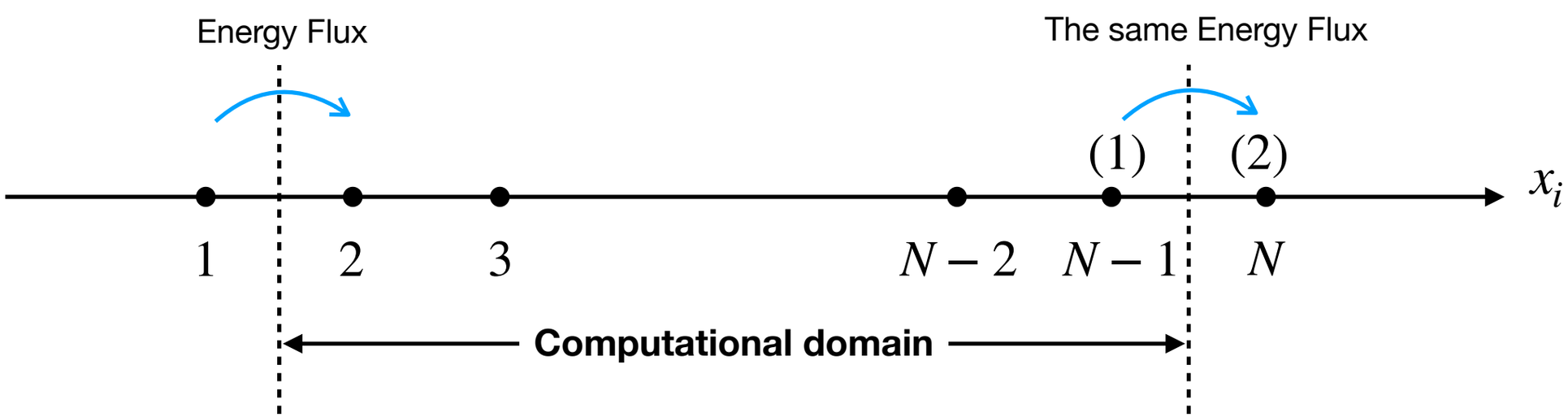
Recall the periodic boundary conditions we've discussed before (which conserves energy):



In the Code

$$Q(1) = Q(N-1)$$
$$Q(N) = Q(2)$$

So in terms of energy conservation, periodic boundary condition ensures the energy fluxes going through the left and right boundary are exactly equal:



So loss of energy out right side is ***compensated*** by gain at left side - energy unchanged

Boundary Conditions in 1-D MHD code

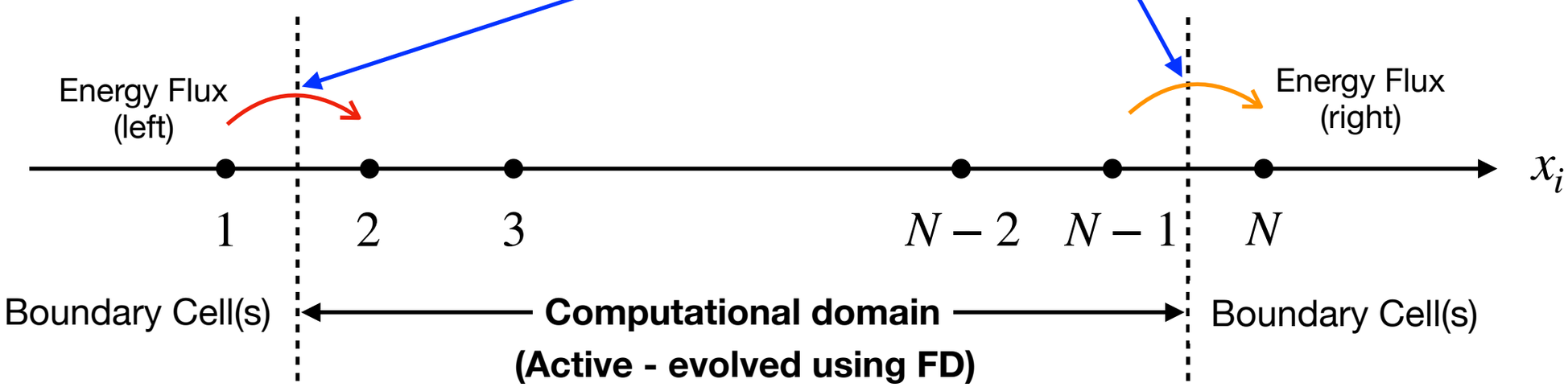
Another type of boundary conditions that conserves total energy is called "***Hard Wall***"

Recall the energy flux on the boundary is calculated as

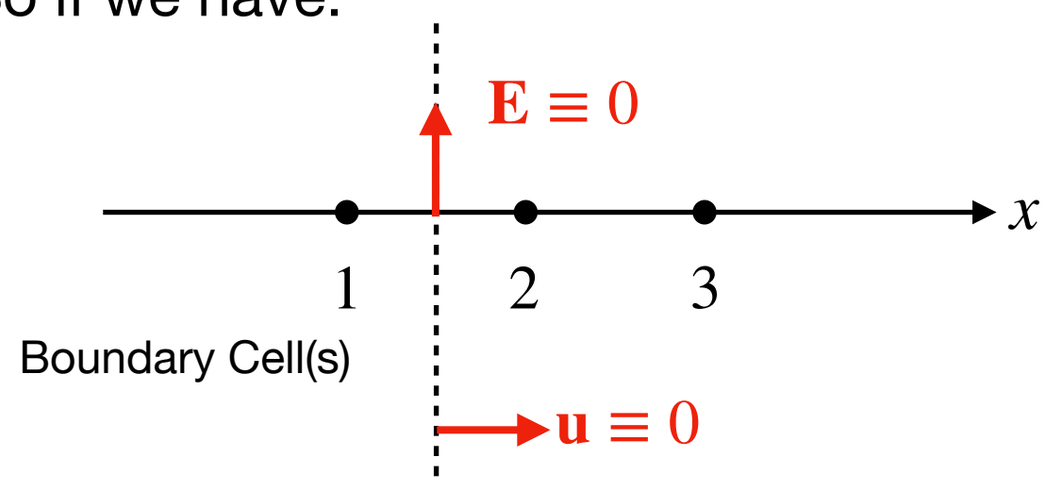
$$\frac{d}{dt}\int_V \mathscr{E}_T dV = -\oint_S d\mathbf{S}\cdot\boxed{\left[\left(\frac{1}{2}\rho u^2 + \frac{\gamma p}{\gamma - 1}\right)\mathbf{u} + \mathbf{E}\times\mathbf{B}\right]}$$

Total Energy Flux Through boundaries

0



So if we have:

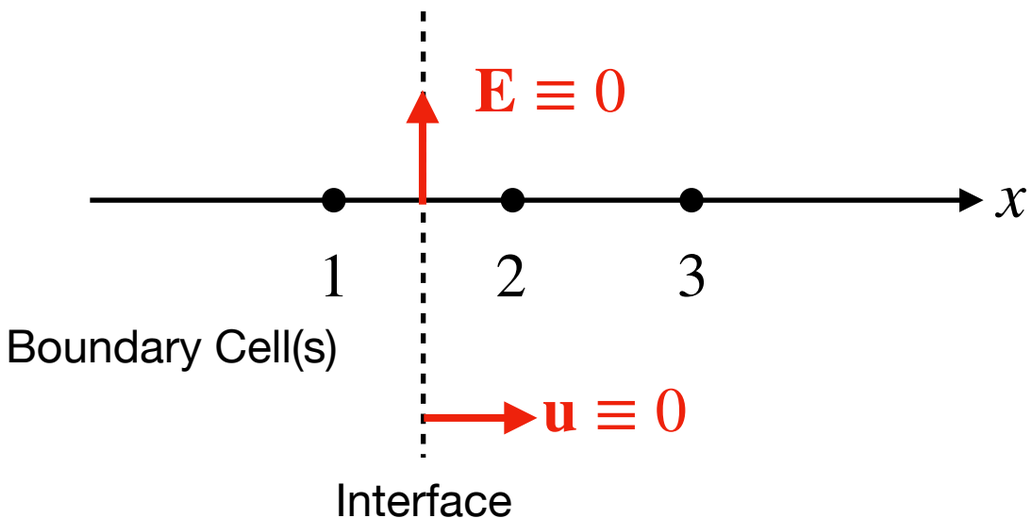


$$\left[\left(\frac{1}{2}\rho u^2 + \frac{\gamma p}{\gamma - 1}\right)\mathbf{u} + \mathbf{E}\times\mathbf{B}\right]_{1\frac{1}{2}} \equiv 0$$

Total energy flux is zero on boundary

Energy-Conserving Boundary in MHD code

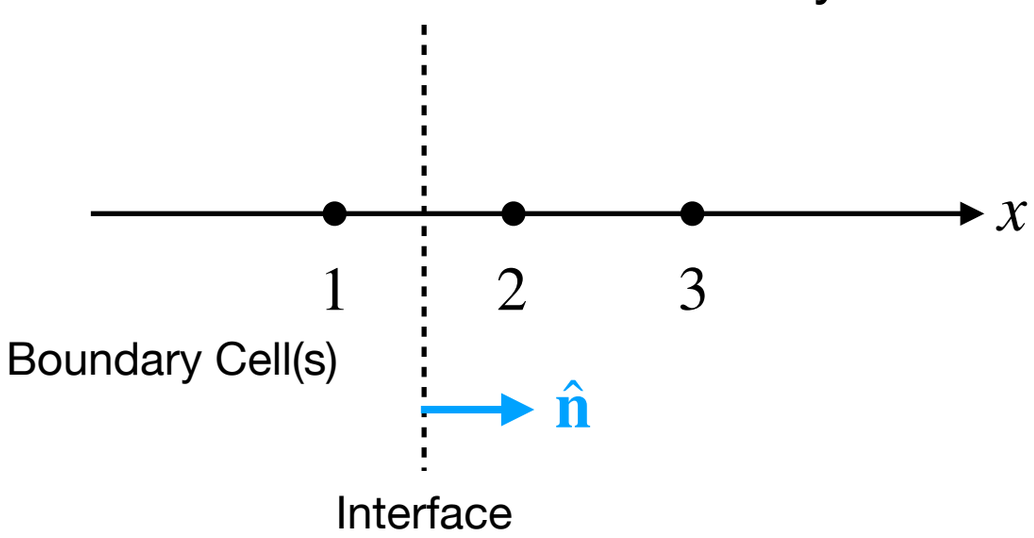
The **u**=0 and **E**=0 boundary conditions does conserve the total energy in an MHD code, but they are quite extreme. For multi-dimensional problems, there are more generalized boundary conditions for the same purpose. Starting from the rate of total energy loss on the boundary:



$$\left[\left(\frac{1}{2}\rho u^2 + \frac{\gamma p}{\gamma - 1}\right)\mathbf{u} + \mathbf{E}\times\mathbf{B}\right]_{1\frac{1}{2}} \equiv 0$$

Total energy flux is zero on boundary

The requirement is no energy flux through the interface (boundary), which means no energy is allowed to across the simulation "wall". Therefore we have two constraints on the velocity and electric field at the boundary:



Hard wall: $\underset{\text{normal}}{u_n} = \mathbf{u}\cdot\hat{\mathbf{n}} \equiv 0$

Conducting wall: $\underset{\text{tangential}}{E_t} = \mathbf{E} - \mathbf{E}\cdot\hat{\mathbf{n}} \equiv 0$

Note that $\hat{\mathbf{n}}\cdot(\mathbf{E}\times\mathbf{B}) = (\hat{\mathbf{n}}\cdot\mathbf{E})\times\mathbf{B} \equiv 0$

Means no Poynting flux across conducting wall

What does “Hard-wall" boundary mean?

The “hard-wall” boundary puts a constraint on the normal component of the velocity, I.e., there’s no flow velocity across the boundary interface (wall). Thus the mass flux through the interface is also expected to be zero - the total mass within the computation domain is conserved (to the scheme accuracy)

Let’s take a look at the mass equation

$$\frac{\partial}{\partial t}\rho = -\nabla\cdot\rho\mathbf{u} \xrightarrow{\int_V dV} \int_V \left(\frac{\partial}{\partial t}\rho = -\nabla\cdot\rho\mathbf{u}\right) dV$$

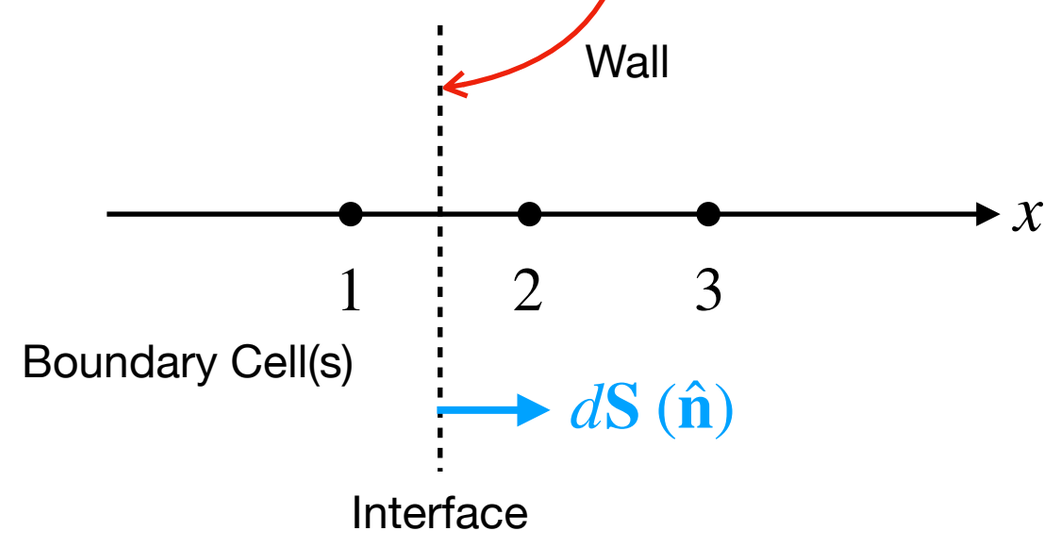
$$\longrightarrow \frac{\partial}{\partial t}\int_V \rho dV = -\int_V \nabla\cdot\rho\mathbf{u} dV \longrightarrow \frac{\partial}{\partial t}\int_V \rho dV = -\oint_{\boxed{S}} \rho\mathbf{u}\cdot d\mathbf{S}$$

So if $\mathbf{u}\cdot d\mathbf{S} \equiv 0$

We have

$$\oint_S \rho\mathbf{u}\cdot d\mathbf{S} \equiv 0 \longrightarrow \frac{\partial}{\partial t}\int_V \rho dV = 0$$

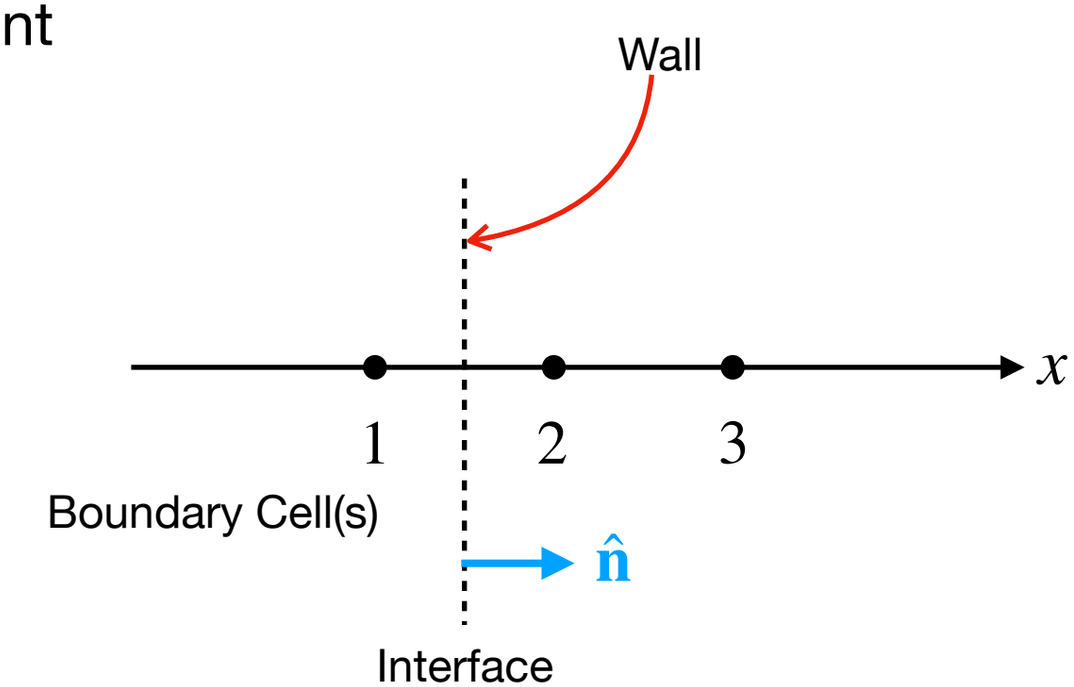
total mass unchanged



How to implement hard-wall boundary?

Hard-wall condition requires the normal component of the velocity across the wall is zero

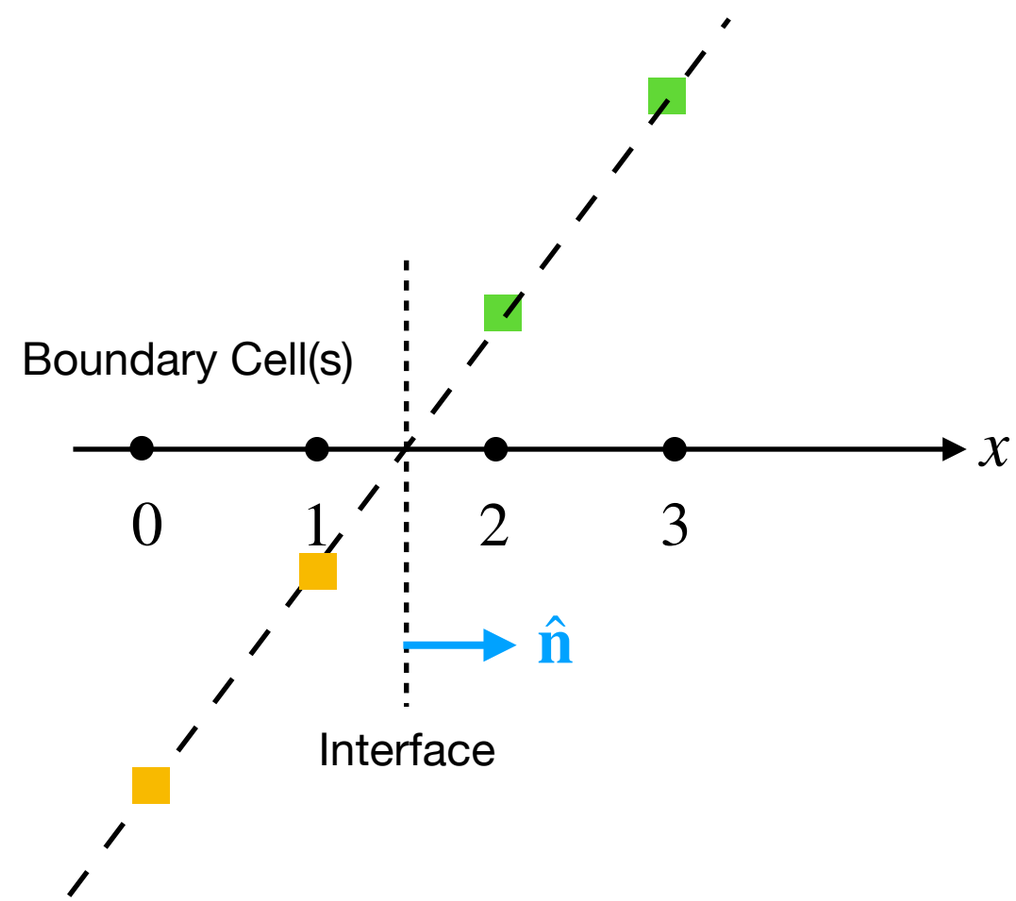
Hard wall: $\underset{\text{normal}}{u_n} = \mathbf{u} \cdot \hat{\mathbf{n}} \equiv 0$



In the code, velocities are specified at grid points 1,2,3,… N. How's interface velocity calculated?

It's done based on the boundary cell(s):

If we set the boundary cells with values that are anti-symmetric with the active computational cells right next to the boundary, the "interpolated" value on the interface is going to be exactly zero (why?)



This is also called **"anti-symmetric" boundary conditions. In the code, simply do:**

$$u_x(1) = -u_x(2)$$

$$u_x(N) = -u_x(N-1)$$