Conservation of stuff

1. Continous equations

For example: momentum conservation in volume $\mathcal{V}$

$$\underbrace{\frac{d}{dt}\int_{\mathcal{V}} \rho \vec{u}\, d\mathcal{V}}_{\text{rate of change of momentum in volume } \mathcal{V}} + \underbrace{\int_S \rho \vec{u}\, \vec{u}\cdot\hat{n}\, dA}_{\text{flux of momentum on surface } S} + \underbrace{\int_S -\tau\cdot\hat{n}\, dA}_{\text{surface seresses}} = 0$$

divergence theorem

$$\int_{\mathcal{V}} \left[\frac{\partial}{\partial t}\rho\vec{u} + \nabla\cdot\left(\rho\vec{u}\vec{u} - \tau\right)\right] d\mathcal{V} = 0$$

$$\Downarrow$$

$$\frac{\partial \rho\vec{u}}{\partial t} + \nabla\cdot\rho\vec{u}\vec{u} = -\nabla\cdot\tau$$

differencial form

We can also derive conservation laws for other quantities

Kinetic energy conservation equation: $\vec{u}\cdot\left(\underbrace{\frac{\partial \rho\vec{u}}{\partial t}} + \nabla\cdot\rho\vec{u}\vec{u} = -\nabla\cdot\tau\right)$

$$\Downarrow$$

$$\frac{\partial}{\partial t}\frac{1}{2}\rho\vec{u}\cdot\vec{u} + \dots$$

Enserophy conservation equation: $\nabla\times\vec{u}\cdot\left(\ \times(\text{momentum equation})\right)$

Key Point: in the continuous "world" I can write conservation laws of any quantity I can think of (some are more useful than others)

In contrast: in the discrete "world" i.e. in our CFD calculation conservation is not guaranteed

Important term

- Only specially constructed methods guarantee conservation
  → these methods or schemes are called conservative
- Conservative methods conserve only the quantities we solve for.

  e.g. if a method numerically integrates mass and momentum, a conservative method with conserve only mass and momentum to machine precision and nothing more (not kinetic energy, not enstrophy, etc...)
- If a CFDer wants to conserve additional quantities then a special method is required

Remember:

- No conservation means a quantity "leaks" from the interior of the computational domain
  e.g. somehow mass or momentum is added or depleted in the interior of the computational domain

Conservation properties are important in CFD

Discretization forms of convection term

- Divergence $\quad \dfrac{\partial}{\partial x_j} u_j u_i$
- Advective $\quad u_j \dfrac{\partial u_i}{\partial x_j}$
- Skew-symmetric $\quad \dfrac{1}{2} \dfrac{\partial}{\partial x_j} (u_j u_i) + \dfrac{1}{2} u_j \dfrac{\partial u_i}{\partial x_j}$
- Rotational $\quad u_j \left(\dfrac{\partial u_i}{\partial x_j} - \dfrac{\partial u_j}{\partial x_i} \right) + \dfrac{1}{2} \dfrac{\partial u_j u_i}{\partial x_j}$

- All forms are equivalent in differential form
- But, have different properties in discrete form

- Please see details in Morinish et al. (1998) in "Additional Material" on HuskyCT
- Check out Table 7

Verification of Euler equations time integration
(Part 2 of Project)

We will exploid that for our discretization

→ Kinetic energy remains constant* (not really)

→ Scalar variance remains constant* (not really)

⇒ these imply that the simulation cannot "blow up" because energy in always bounded by the initial condition

Verification Test 1 (easy)

- Initialize u, v, θ with random numbers but apply u, v BC
  u and v periodic left/right & $v=0$ top/bottom walls
- Perform 100 time steps (any Δt as long the integration is stable)
- Success if
  a. Flow is divergence-free
  b. u, v, θ remain finite

Verification Test 2 (advanced)

1. a. Initialize u, v, θ with random numbers
   b. Apply BCs
   c. Make flow divergence free

2. Calculate initial

   only physical domain
   NO GHOST cells

   Kinetic energy $K_0 = \sum_i \sum_j (u_{ij}^2 + v_{ij}^2)$

   Temperature "energy" $V_0 = \sum_i \sum_j \theta_{ij}^2$

3. Integrate to time T_{end} with constant Δt

   pick a time

4. Calculate final

   Kinetic energy $K_e = \sum_i \sum_j (u_{ij}^2 + v_{ij}^2)$

   Temperature "energy" $V_e = \sum_i \sum_j \theta_{ij}^2$

Ideally $K_e = K_0$ and $V_e = V_0$ since the method conserves K and V by constraction

- However, Runge-Kutta is not exact time integration and introduces a Δt^3 error
- The error is dissipative, because of numerical stability

  Thus: $K_e < K_0$ and $V_e < V_0$

To verify the method:

- Fix (Do NOT change) Δx, Δy and T_e
- Do steps 1–4 with different Δt
- For each Δt calculate errors:

$$E_K(\Delta t) = K_0 - K_e$$

$$E_V(\Delta t) = V_0 - V_e$$

Do NOT take absolute value here

- Show that $E_K \sim \Delta t^3$ and $E_V \sim \Delta t^3$