Numerical Simulation Methods in Geophysics, Part 12: Advection problems and the Finite Volume Method

1. MGPY+MGIN

thomas.guenther@geophysik.tu-freiberg.de

Recap

1. The Finite Difference (FD) method
   - Poisson equation in 1D, look into 2D/3D
   - diffusion equation in 1D, time-stepping
2. Solving the hyperbolic (acoustic) wave equation in 1D
3. The Finite Element (FE) method
   - Poisson and diffusion equation in 1D
   - (complex) Helmholtz equation in 2D for EM problems
   - solving EM problems and computational aspects

Solution in pyGIMLi

step by step

```
import pygimli.solver as ps
A = ps.createStiffnessMatrix(mesh, a)
M = ps.createMassMatrix(mesh, b)
f = ps.createLoadVector(mesh, f)
ps.assembleNeumannBC(b, boundaries)
ps.assembleDirichletBC(A, b)
ps.assembleRobinBC(A, boundaries, b)
u = ps.linSolve(A, b)
```

or shortly

```
bc = {'Dirichlet': {4: 1.0, 3: 0.0},
      'Neumann': {2: 1.0}}
u = ps.solveFiniteElements(mesh, a, b, f,
bc=bc,
                                   t=times, c)
u = pygimli.physics.seismics.solvePressureWave(
    mesh, velocities, times, sourcePos,
uSource)
```

$$c\frac{\partial u}{\partial t} = \mathbf{\nabla}\cdot(a\mathbf{\nabla}u) + bu + f(\mathbf{r},t)$$

Advection

Volume elements move with velocity $\mathbf{v}$ and take (advectio=move with)

$$\frac{\partial T}{\partial t} \Rightarrow \frac{\partial T}{\partial t} + \frac{\mathrm{d}z}{\mathrm{d}t}\frac{\partial T}{\partial z}$$

in 3D $\mathbf{v} \cdot \boldsymbol{\nabla} T \Rightarrow$ advection-dispersion equation

$$\frac{\partial T}{\partial t} = \boldsymbol{\nabla} \cdot a \boldsymbol{\nabla} T + Q - \mathbf{v} \cdot \boldsymbol{\nabla} T$$

Advection-dispersion equation (general)

$$\frac{\partial u}{\partial t} + \mathbf{v} \cdot \boldsymbol{\nabla} u = \boldsymbol{\nabla} \cdot (a \boldsymbol{\nabla} u) + bu + f(\mathbf{r}, t)$$

$\mathbf{v} \cdot \boldsymbol{\nabla} u$ (Advection), $\boldsymbol{\nabla} \cdot (a \boldsymbol{\nabla} u)$ (Dispersion=Diffusion)

Solve, e.g., by `pyGIMLi.solver.solveFiniteVolume` (Link)

```
bc = dict(Dirichlet={4: 1.0, 3: 0.0}, Neumann={2: 1.0})
u = ps.solveFiniteVolume(mesh, a, b, f, bc=bc, t=times, c)
```

Computational fluid dynamics

Magnetohydrodynamic induction equation

$$\frac{\partial \mathbf{B}}{\partial t} = \boldsymbol{\nabla} \times \frac{1}{\mu\sigma}\boldsymbol{\nabla} \times \mathbf{B} + \boldsymbol{\nabla} \times (\mathbf{v} \times \mathbf{B})$$

Forces on fluid (induction, Coriolis, pressure, gravity, Lorentz)

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho(\mathbf{v} \cdot \boldsymbol{\nabla})\mathbf{v} + \boldsymbol{\nabla} p = -2\omega \times \mathbf{v} + \alpha T\mathbf{g} + \mathbf{j} \times \mathbf{B}/\rho$$

FE solution for inhomogeneous viscosity

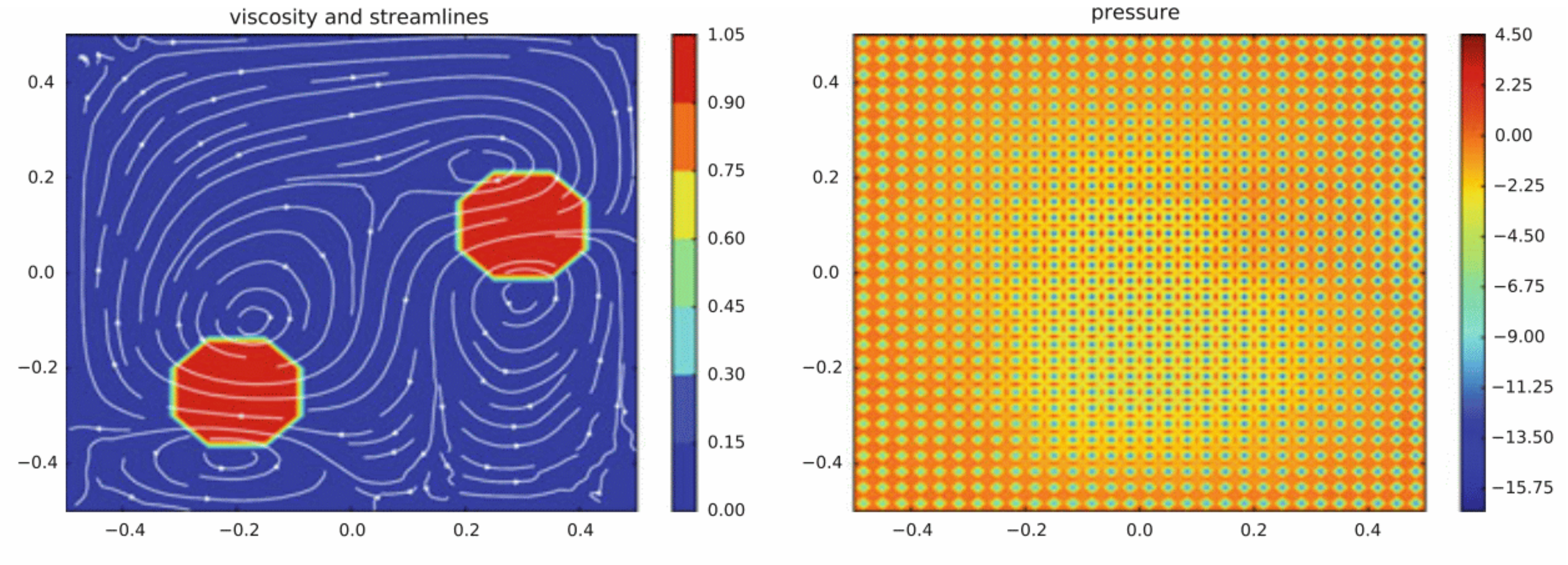


Fig. 10.1 *Left* velocity field determined with a full momentum+pressure solver for the same two particles in a box, using the full Stokes operator. Here, the solution is the same as the one in Fig. 9.5 except near the walls, due to the different Boundary conditions (free slip in the other case, no-slip in this case). *Right* pressure field for the same case. One observes that an instability arises. It arises because velocity and pressure are both solved on the same nodes

Simple 1D advection problem

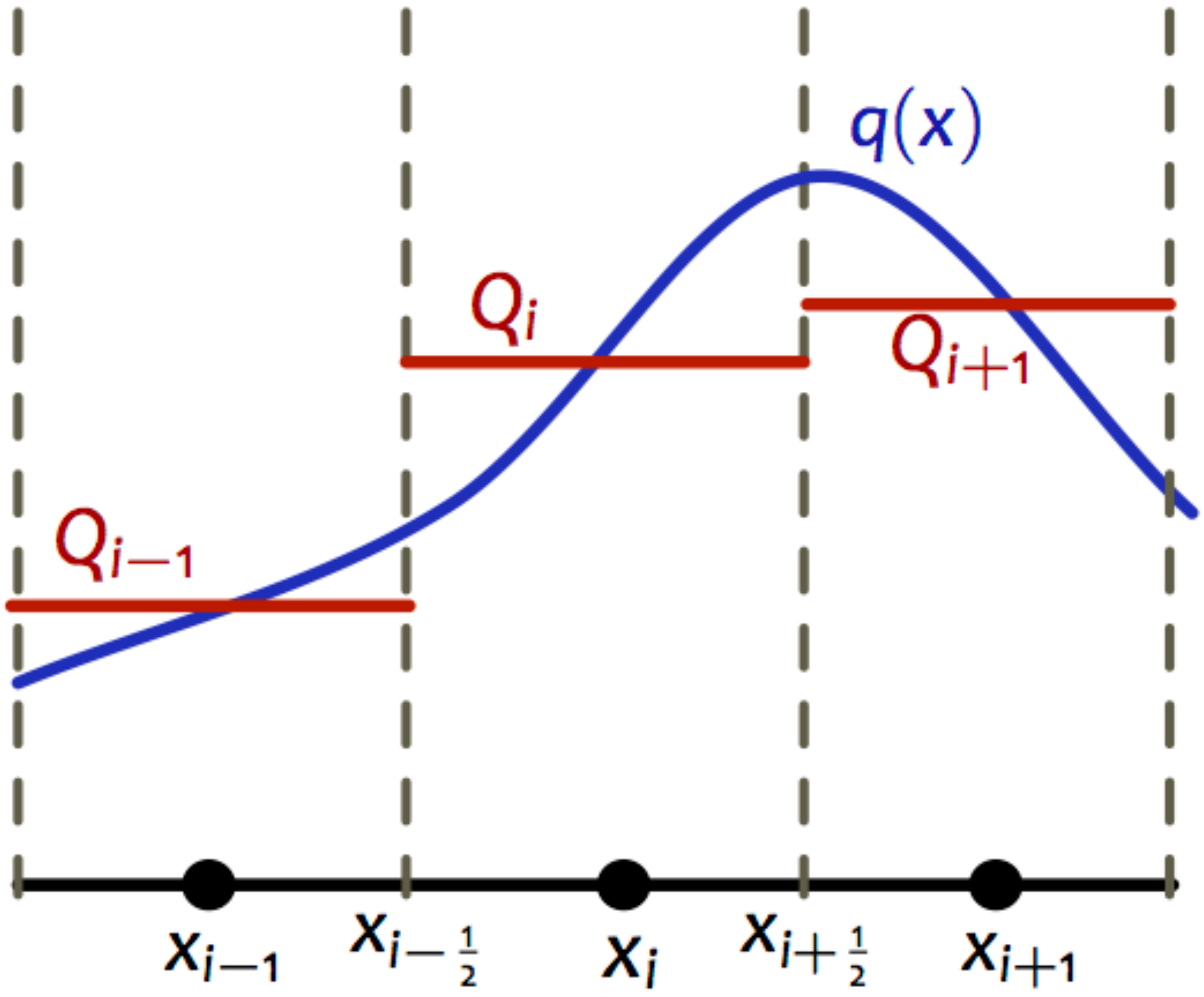
$$\frac{\partial u}{\partial t} + \frac{\partial v(u)}{\partial x} = 0$$

Solution u_i at node i represents average value over cell

$$\overline{u}_i(t) = \frac{1}{x_{i+1/2} - x_{i-1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} u(x,t)\mathrm{d}x$$

$$\int_{t_1}^{t_2} \frac{\partial u}{\partial t} = u(x,t_2) - u(x,t_1) = -\int_{t_1}^{t_2} \frac{\partial v(u)}{\partial x}$$

FV



Simple 1D advection problem

$$u(x,t_2) = u(x,t_1) - \int_{t_1}^{t_2} \frac{\partial v(u)}{\partial x}$$

$$\overline{u}(t_2) = \frac{1}{x_{i+1/2} - x_{i-1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \left(u(x,t_1) - \int_{t_1}^{t_2} \frac{\partial v(u)}{\partial x} \right)$$

$$\overline{u}(t_2) = \overline{u}(t_1) - \frac{1}{x_{i+1/2} - x_{i-1/2}} \left(\int_{t_1}^{t_2} v_{i+1/2} \mathrm{d}t - \int_{t_1}^{t_2} v_{i-1/2} \mathrm{d}t \right)$$

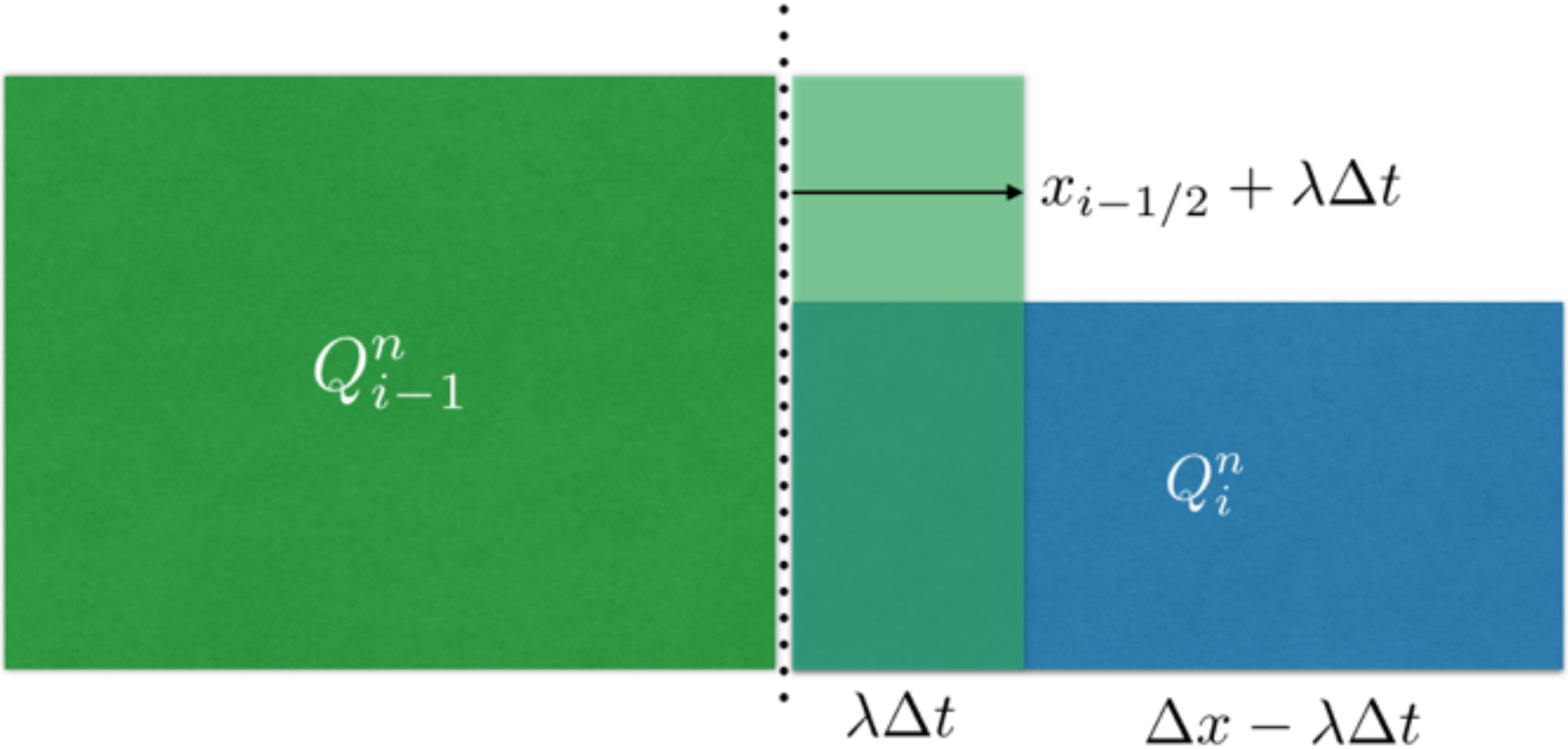
Result

$$\frac{\partial \overline{u}_i}{\partial t} + \frac{1}{\Delta x_i}\left(v_{i+1/2} - v_{i-1/2}\right) = 0$$

Note

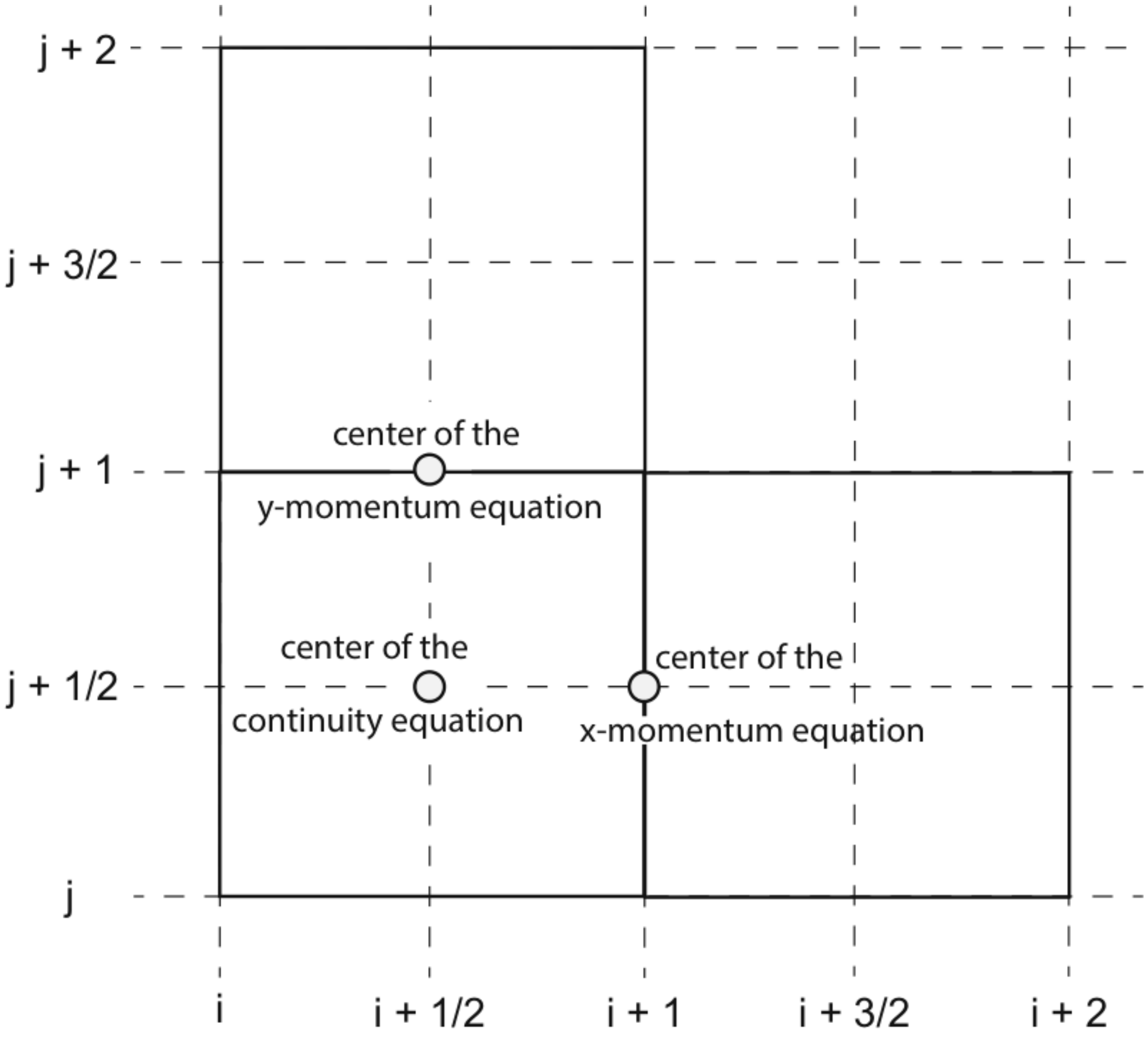
Finite volume schemes are conservative as cell averages change through the edge fluxes. In other words, one cell's loss is always another cell's gain!

Visualization



Similarity to staggered (E, B) grid methods

Fig. 10.2 Sketch of the of the indices and of the positions where the continuity, the x-momentum and the y-momentum equations are calculated. The mid indexes ($\frac{1}{2}$ and $\frac{3}{2}$) indicate the center between two edges of the volume. By calculating the finite volume at the center of each side one is also implicitamente integrating the momentum equations on each side. It is possible to show that this choice makes this algorithm second-order accurate.



Conservation law (3D) problem

$$\frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{v}(u) = 0$$

volume integral over cell, using Gauss' law

$$\int_{\Omega_i} \frac{\partial \mathbf{u}}{\partial t} \mathrm{d}\Omega + \int_{\Omega_i} \boldsymbol{\nabla} \cdot \mathbf{v}(u) \mathrm{d}\Omega = 0 = \int_{\Omega_i} \frac{\partial \mathbf{u}}{\partial t} \mathrm{d}\Omega + \int_{\Gamma_i} \mathbf{v}(u) \cdot \mathbf{n} \, \mathrm{d}\Gamma$$

$$\Rightarrow \frac{\partial \overline{u}}{\partial t} + \frac{1}{V_i} \int_{\Gamma_i} \mathbf{v}(u) \cdot \mathbf{n} \, \mathrm{d}\Gamma = 0$$