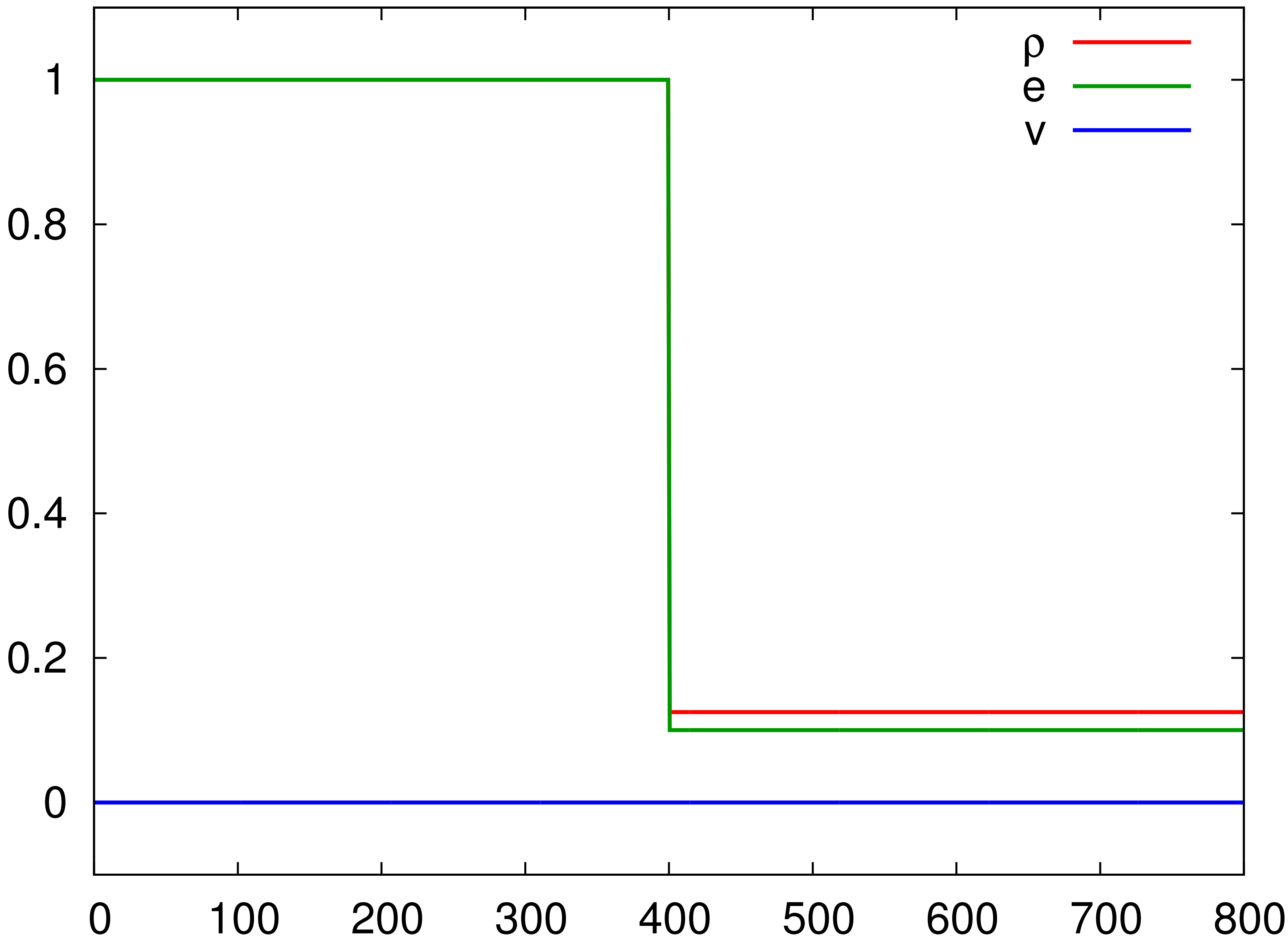
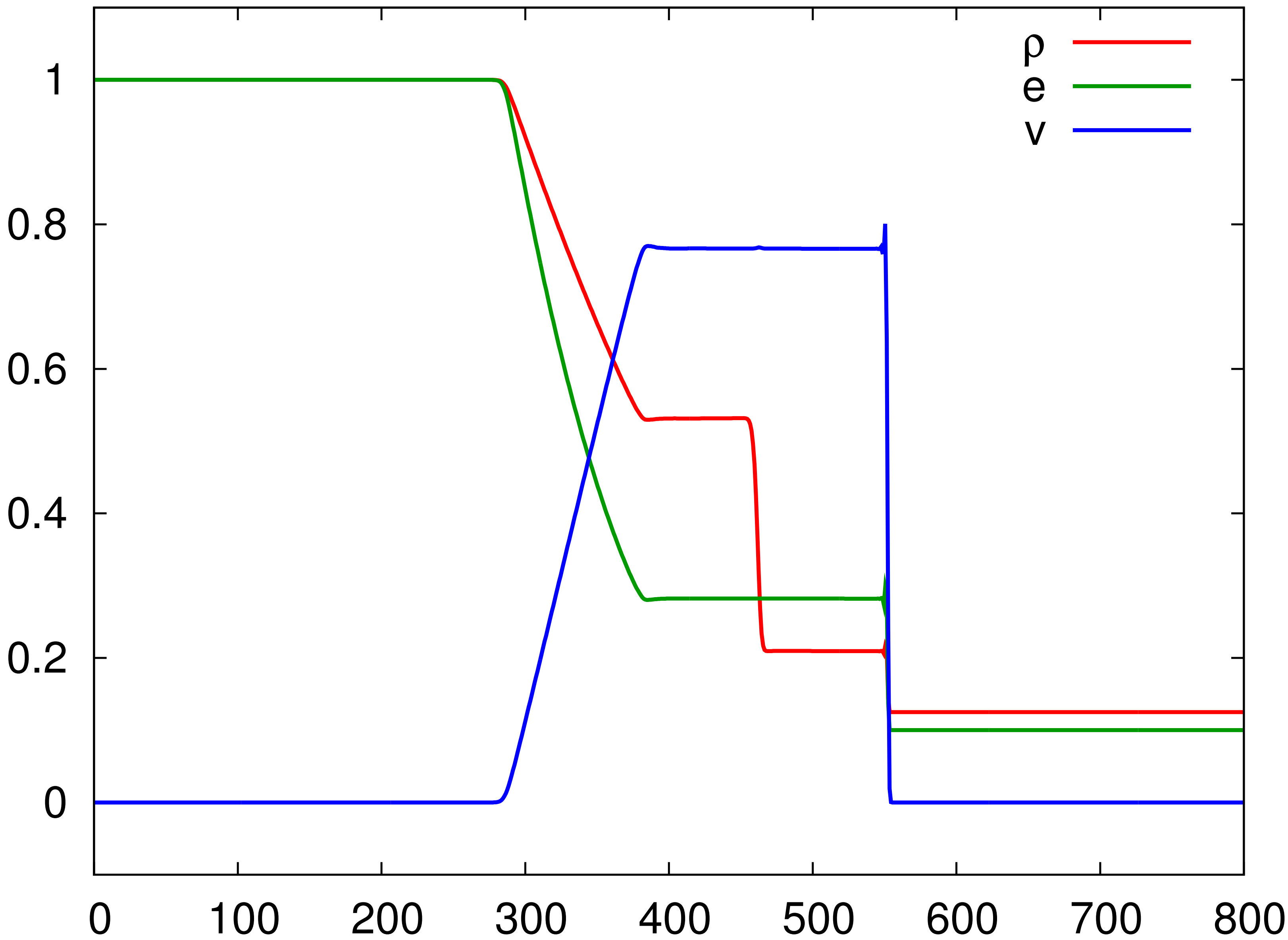
Hydrodynamic simulations of the ISM 2

Richard Wünsch (**richard@wunsch.cz**)
Astronomický ústav AV ČR, Boční II 1401, Praha 4

Outline:

1. Basic numerical schemes:
   Lax scheme, numerical viscosity, Courant-Friedrichs-Lewy condition
2. Introduction to ZEUS
3. Homework: Kelvin-Helmholtz instability with ZEUS


ρ
e
v
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800


ρ
e
v
1
0.8
0.6
0.4
0.2
0
0
100
200
300
400
500
600
700
800

Finite Volume Methods - Conservative formulation

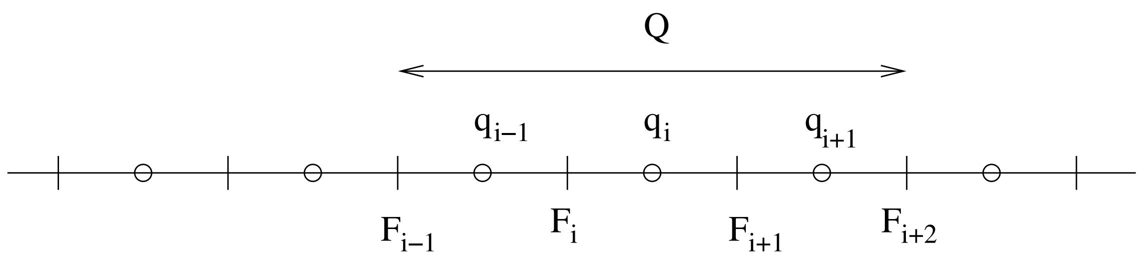
$$\frac{\partial}{\partial t}\left(\begin{array}{c}\rho\\ \rho v_i\\ e\end{array}\right)+\frac{\partial}{\partial x_j}\left(\begin{array}{c}\rho v_j\\ \rho v_j v_i\\ e v_j\end{array}\right)=\left(\begin{array}{c}0\\ -\frac{\partial p}{\partial x_i}-\rho\frac{\partial \Phi}{\partial x_i}\\ -\frac{\partial (p v_j)}{\partial x_j}-\rho v_j\frac{\partial \Phi}{\partial x_j}\end{array}\right)$$

in general:

$$\frac{\partial q}{\partial t}+\frac{\partial F}{\partial x}=\Xi$$

Integrate over some volume, use divergence theorem:

$$\frac{\partial}{\partial t}\int_V q d^3V+\int_{\partial S} F d^2S=\int_V \Xi d^3V$$

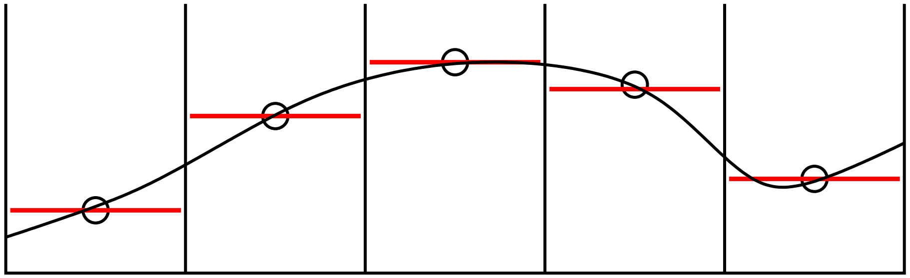


disregarding Ξ for a moment:

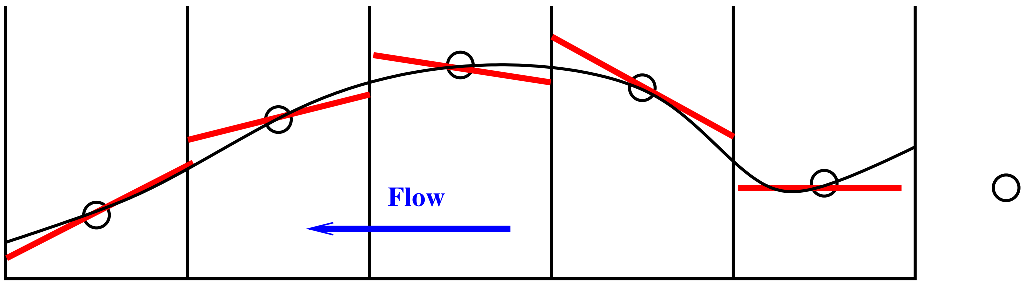
$$\frac{\partial q_i}{\partial t}=F_i-F_{i+1}, \text{must work also for Q}: \frac{\partial Q}{\partial t}=\frac{\partial q_{i-1}}{\partial t}+\frac{\partial q_i}{\partial t}+\frac{\partial q_{i+1}}{\partial t}=F_{i-1}-F_{i+2}$$

Interpolation order

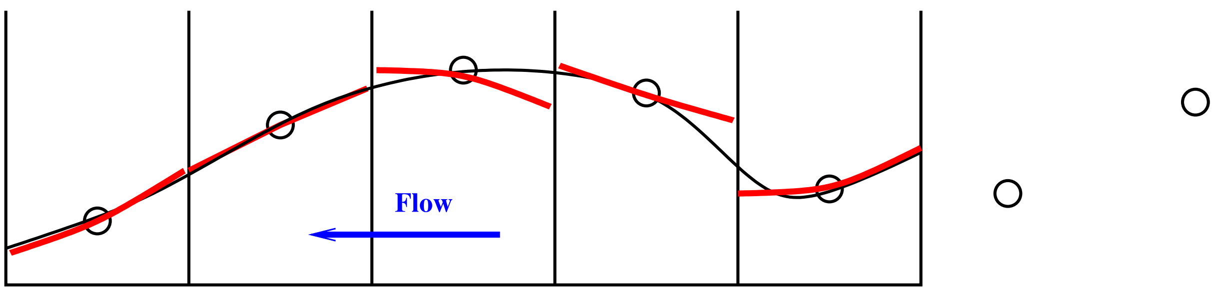
- 1st order, donnor-cell



- 2st order, piecewise linear



- 3st order, piecewise parabolic advection (PPA)



Monotonicity

- van Leer (1977), to prevent spurious maxima and minima

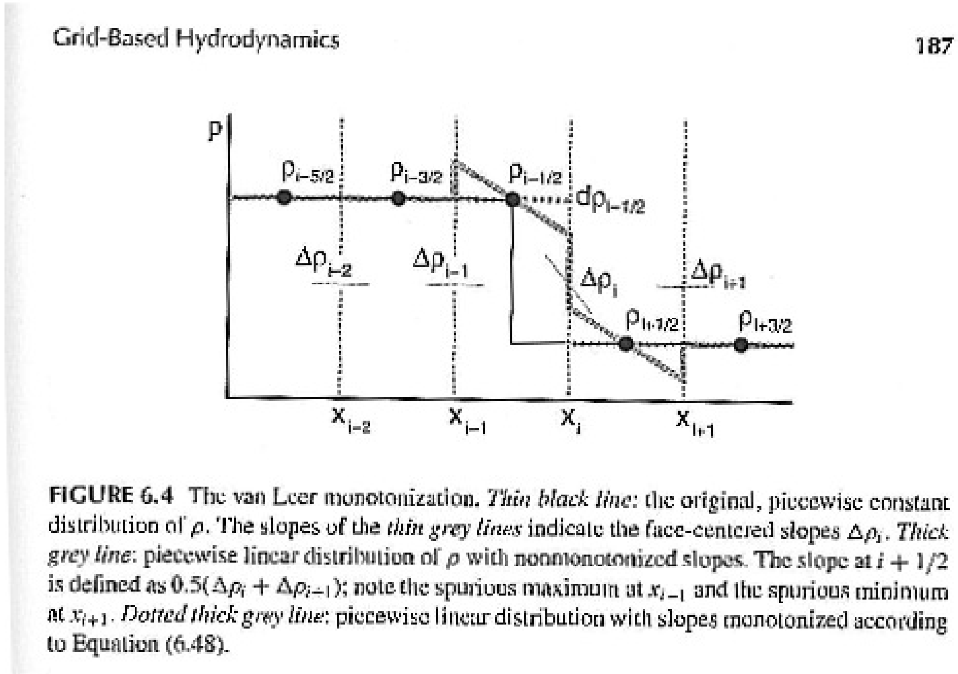

Grid-Based Hydrodynamics 187

FIGURE 6.4 The van Leer monotonization. *Thin black line*: the original, piecewise constant distribution of ρ. The slopes of the *thin grey lines* indicate the face-centered slopes $\Delta\rho_i$. *Thick grey line*: piecewise linear distribution of ρ with nonmonotonized slopes. The slope at $i + 1/2$ is defined as $0.5(\Delta\rho_i + \Delta\rho_{i-1})$; note the spurious maximum at x_{i-1} and the spurious minimum at x_{i+1}. *Dotted thick grey line*: piecewise linear distribution with slopes monotonized according to Equation (6.48).

- $\rho_i(w) = \rho_i + w d\rho_i$, where w is norm. distance from the cell centre
- the monotonized slope $d\rho_i$ is:

$$d\rho_i = \begin{cases} \frac{2\Delta\rho_i\Delta\rho_{i+1}}{\Delta\rho_i+\Delta\rho_{i+1}} & \text{if } \Delta\rho_i\Delta\rho_{i+1} > 0 \\ 0 & \text{otherwise} \end{cases}$$

where $\Delta\rho_i = \rho_{i+1} - \rho_i$

Artificial viscosity

- necessary to dissipate a fraction of kinetic energy at shocks
- otherwise numerical instability appears
- von Neumann & Richtmeyer (1950):

$$\frac{v_i^{n+1} - v_i^n}{\Delta t} = -\frac{2(q_i^n - q_{i-1}^n)}{\Delta x(\rho_i^n + \rho_{i-1}^n)}$$

$$q_i = \begin{cases} C\rho_i(v_{i+1} - v_i)^2 & \text{if } (v_{i+1} - v_i) < 0 \\ 0 & \text{otherwise} \end{cases}$$

- sensitive only to compression
- satisfies conservation laws
- C - dimensionless constant, how many grid cells the shock will be spread out

ZEUS installation

- Mike Norman, Jim Stone, David Clarke
- official version 3.4.2 available at Laboratory for computational Astrophysics, University of California, San Diego
- D. Clarke version 3.5 (includes OpenMP)
- "easy" to install package:
  `http://galaxy.ig.cas.cz/~richard/teaching/NAST021/`
- prerequisities:
  - UNIX type OS (tested with Linux)
  - Fortran77 and C compilers (tested with gfortran and gcc)
  - csh and ex (comes with vi text editor)
  - recommended: gnuplot, hdf4 libraries
- download Zdistr-gfortran.tgz and follow instructions in the README file
- compile the code and run the example shkset_XYZ
- use gnuplot script "fig.gp" to plot the results

The ZEUS code

- Operator splitting:
  $\frac{dy}{dt} = \mathcal{L}(y) = \mathcal{L}_1(y) + \mathcal{L}_2(y) + \ldots$

$$(y^1 - y^0)/\Delta t = \mathcal{L}_1(y^0)$$
$$(y^2 - y^1)/\Delta t = \mathcal{L}_2(y^1)$$
$$\ldots$$

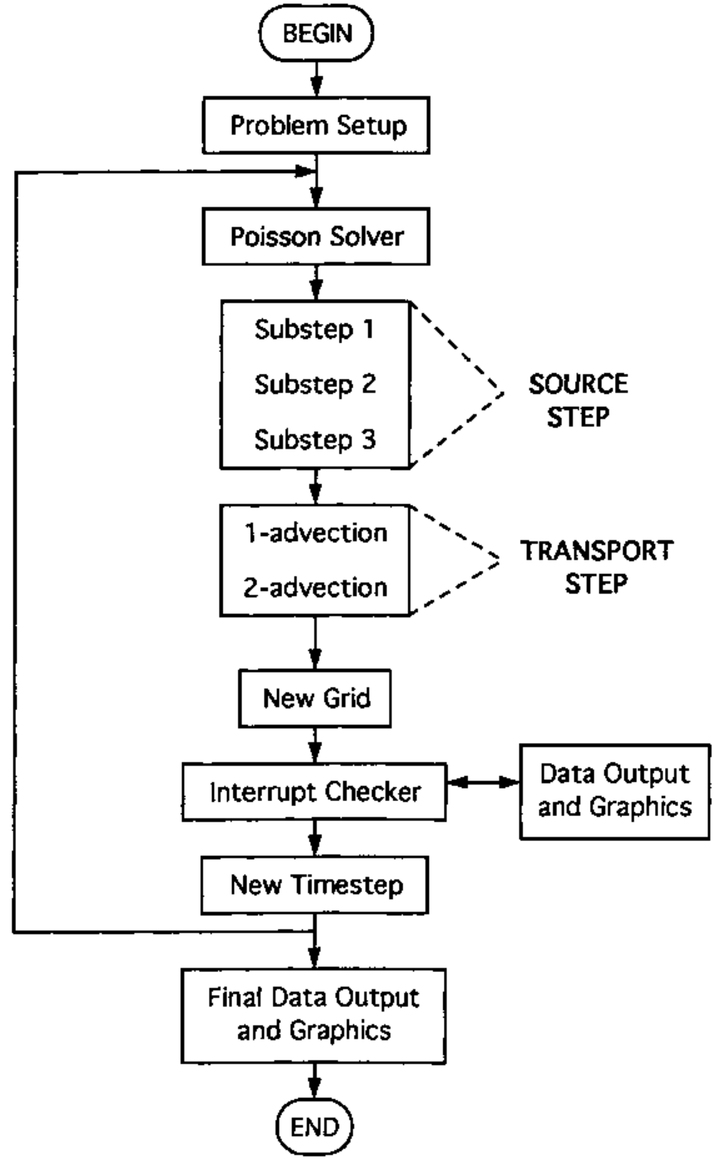
- Source step

$$\rho\frac{\partial v}{\partial t} = -\nabla P\Big|_{\mathrm{SS1}} - \rho\nabla\Phi\Big|_{\mathrm{SS1}} - \nabla\cdot Q\Big|_{\mathrm{SS2}}$$
$$\frac{\partial e}{\partial t} = -p\nabla\cdot v\Big|_{\mathrm{SS3}} - Q:\nabla v\Big|_{\mathrm{SS2}}$$

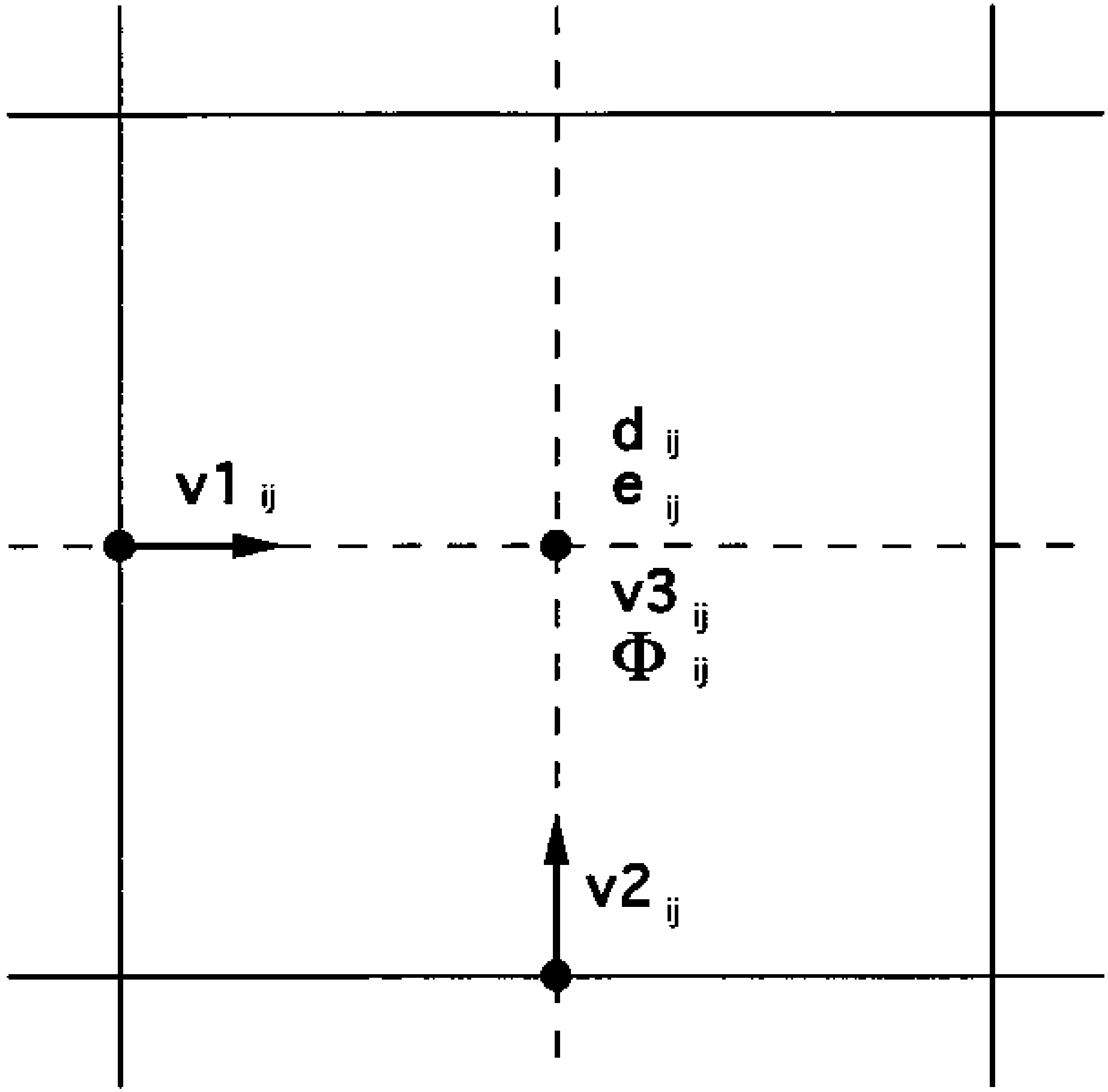
- Transport step

$$\frac{d}{dt}\int \rho dV = -\int_{\partial V} \rho(v - v_g)dS$$
$$\frac{d}{dt}\int \rho v dV = -\int_{\partial V} \rho v(v - v_g)dS$$
$$\frac{d}{dt}\int e dV = -\int_{\partial V} e(v - v_g)dS$$

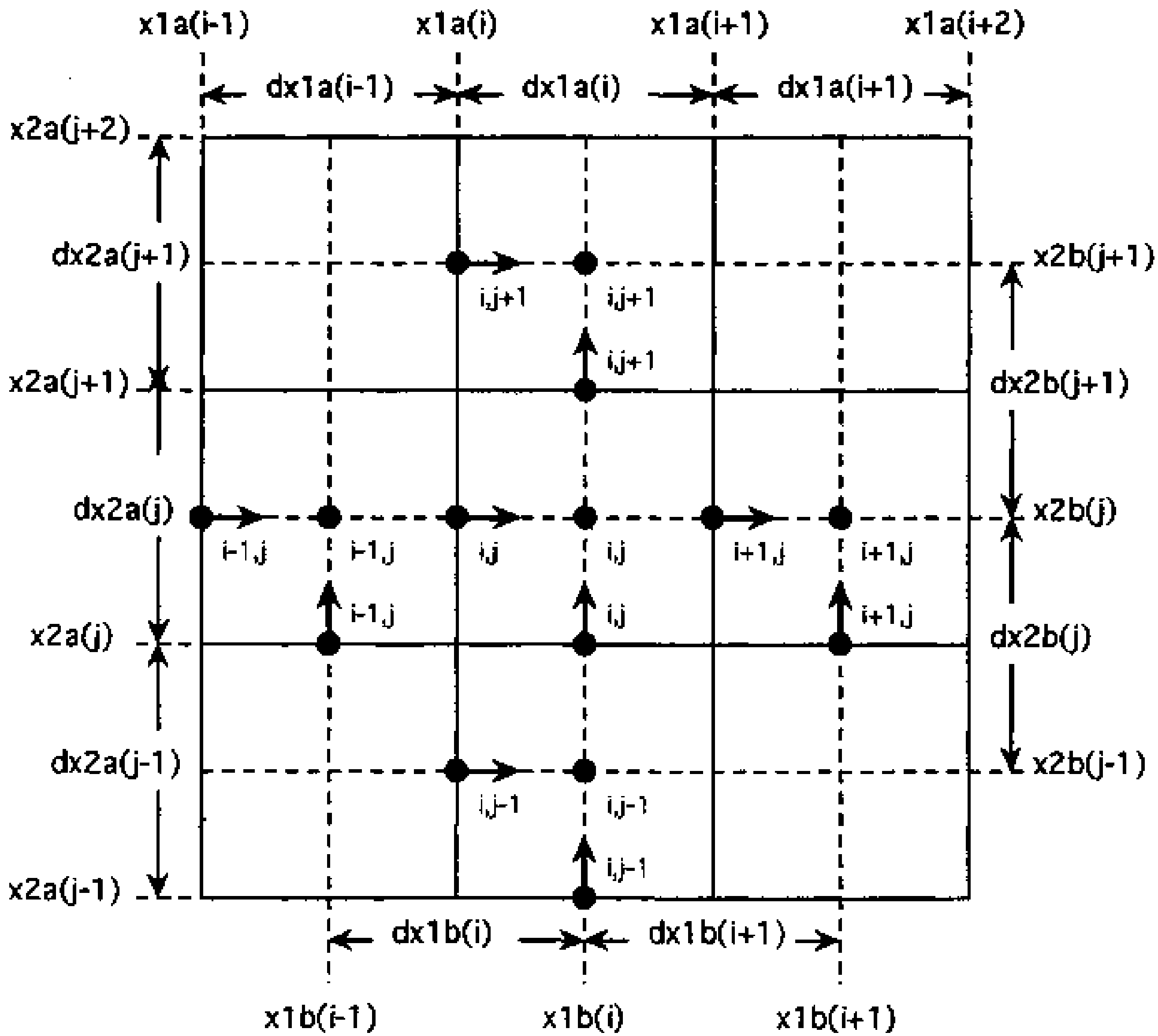
ZEUS - flowchart



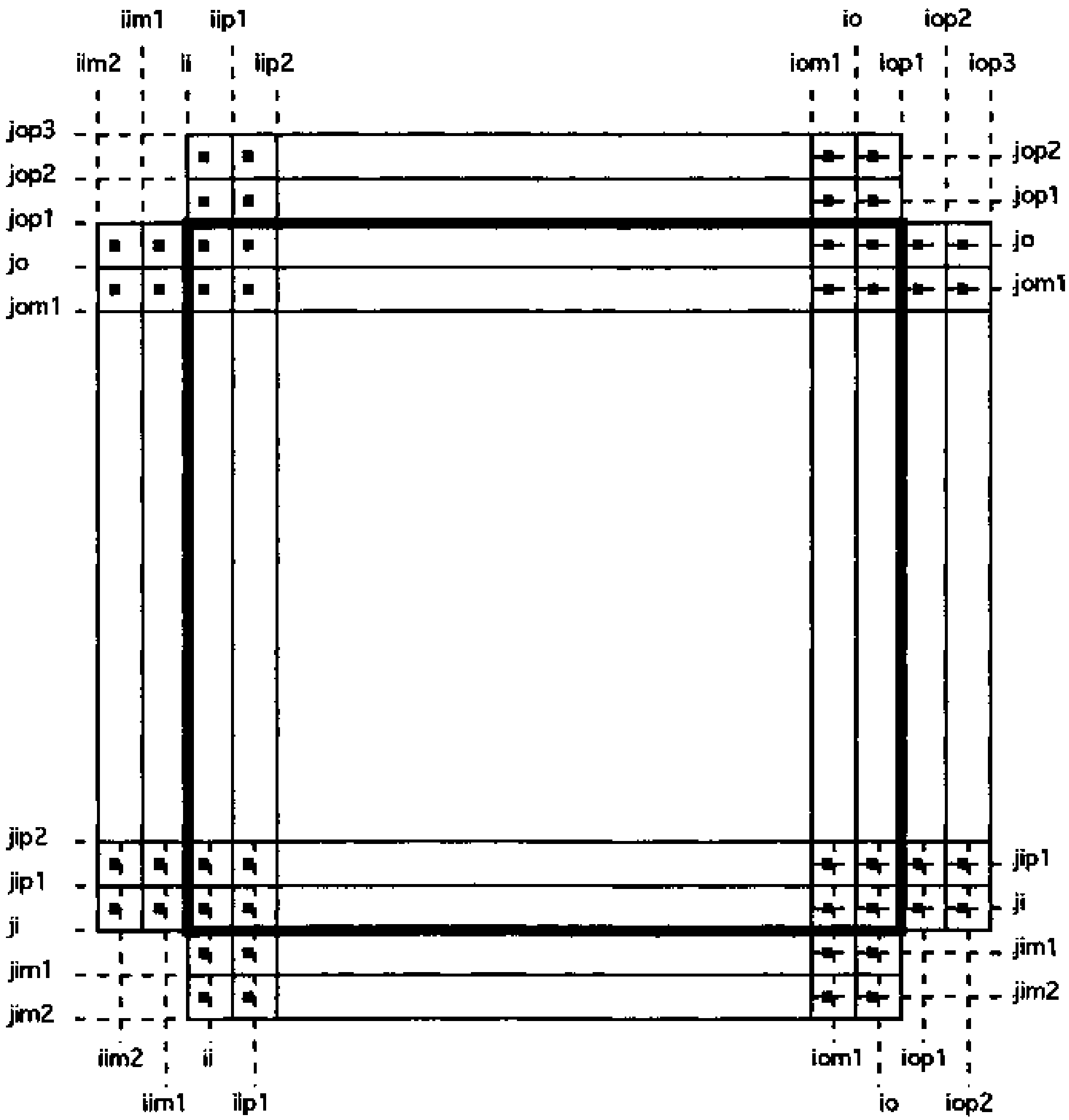
One ZEUS grid-cell



The ZEUS grid



Computational domain



ZEUS - How to setup a new problem

- Three steps: write a setup subroutine, compile and run
- Writing a setup subroutine
  - ▷ write it into the chengedeck in src/
  - ▷ see example chgz.cloud
  - ▷ read parameters from inzeus, set intitial values on the grid, call bndyflgs and bndyall
- Compilation:
  - ▷ preprocessing by EDITOR (xedit22)
  - ▷ configured by the zeus34.mac file
  - ▷ zeus34.mac generated by zeus34.s script from the problem definition file in examples/
- Running:
  - ▷ ./xzeus34 reads parameters from inzeus
  - ▷ inzeus generated also by zeus34.s script from the problem definition file in examples/

You typically need to do

- write chgz.myproblem (in src/ directory)
- write problem definition file: examples/myproblem
  (and mention your chgz.myproblem in it)
- run commands:

```
cd myproblem
.  ~/ZEUS3D/arch/setZeus3D.Linux.gfortran
~/ZEUS3D/bin/zeus34.s ~/ZEUS3D/examples/myproblem
./xzeus34
```

- Try to run the simulation of Kelvin-Helmholtz instability in 2D
  (it will be equivalent of the exam for my part of this course)