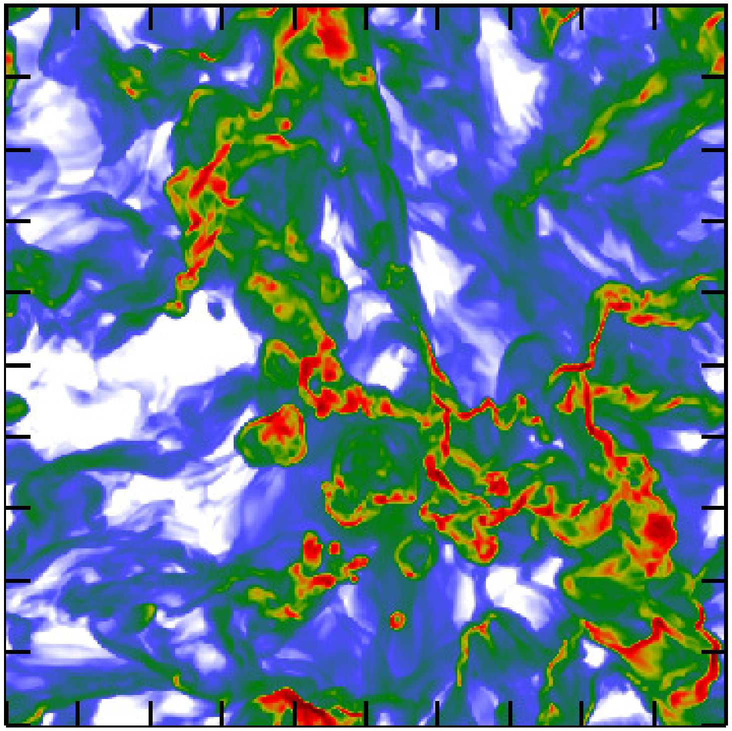
Hydrodynamic simulations of the ISM

Richard Wünsch (**richard@wunsch.cz**)

Astronomický ústav AV ČR, Boční II 1401, Praha 4

Aims:

1. Basics of theory of grid-based hydrodynamic codes
2. Practical experience with the simple hydrocode (ZEUS)



Numerical hydrodynamics in astrophysics

- Cosmology (e.g. Millenium simulation - Gadget2)
- Galaxies, Clusters of them (e.g. Ram pressure stripping)
- ISM in galaxies (e.g. Formation of GMC, ISM turbulence, Bubbles)
- Star formation (Collapse of GMC, formation of clusters)
- Stellar evolution (stellar interior and atmospheres, stellar winds, SN explosions)
- Formation of planets (protoplanetary discs)
- Accretion discs around compact objects (BH, NS, jets, . . .)
- . . .

Numerical methods

- Grid/AMR codes
  - Eulerian
  - traditional approach, HD eqs. discretized on a grid
  - Pros: better with shocks and HD instabilities, easier to paralelize, faster
  - Cons: anisothropy, artefacts along grid axes, computational domain
  - ZEUS, Flash, Ramses, Enzo, ...
- SPH codes
  - Lagrangian
  - evolved from Nbody codes, add smoothing length → pressure force
  - Pros: follow mass at grav. collapse, no problems with vacuum
  - Cons: cannot describe some HD phenomena, problems with discontinuities, never crash
  - Gadget, Seren, Vine, Gasoline, ...
- Semi-lagrangian codes
  - try to get rid of disadventages of both groups
  - Arepo, ...

Paralelization scheme

- Shared memory (OpenMP)
  - ▷ all processes have access to the same memory
  - ▷ easier to code, but limited number of cores
  - ▷ ZEUS, Seren, Vine . . .
- Distributed Memory (MPI)
  - ▷ each process has its own memory, communication through explicit messages
  - ▷ hard to code, limits on core number given only by $
  - ▷ Flash, Ramses, Gadget, Seren . . .
- GPU Computing
  - ▷ evolved from graphic cards
  - ▷ ~ 500 simple cores running ~ 10000 threads
  - ▷ similar to shared memory model, but different levels of shared memory
  - ▷ nVidia Tesla (CUDA)

Hydrodynamic equations

$$\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{v} = 0$$

$$\frac{D}{Dt} \equiv \frac{\partial}{\partial t} + \mathbf{v}\cdot\nabla$$

$$\rho\frac{D\mathbf{v}}{Dt} = -\nabla p - \rho\nabla\Phi$$

$$p = (\gamma - 1)e = \frac{\rho k T}{\mu m_{\mathrm{H}}}$$

$$\rho\frac{D}{Dt}\left(\frac{e}{\rho}\right) = -p\nabla\cdot\mathbf{v}$$

$$\nabla^2\Phi = 4\pi G\rho$$

- conservation laws
- can be derived as velocity moments of Boltzmann eq.
- equation of state closes the set
- additional physicas may be included by adding more terms and eqs. (in a similar way as gravity)

Advection equation

HD eqs can be writen as:

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

i $=$ 1,2,3 (depends on dimensionality of the problem)

General advection equation has a form:

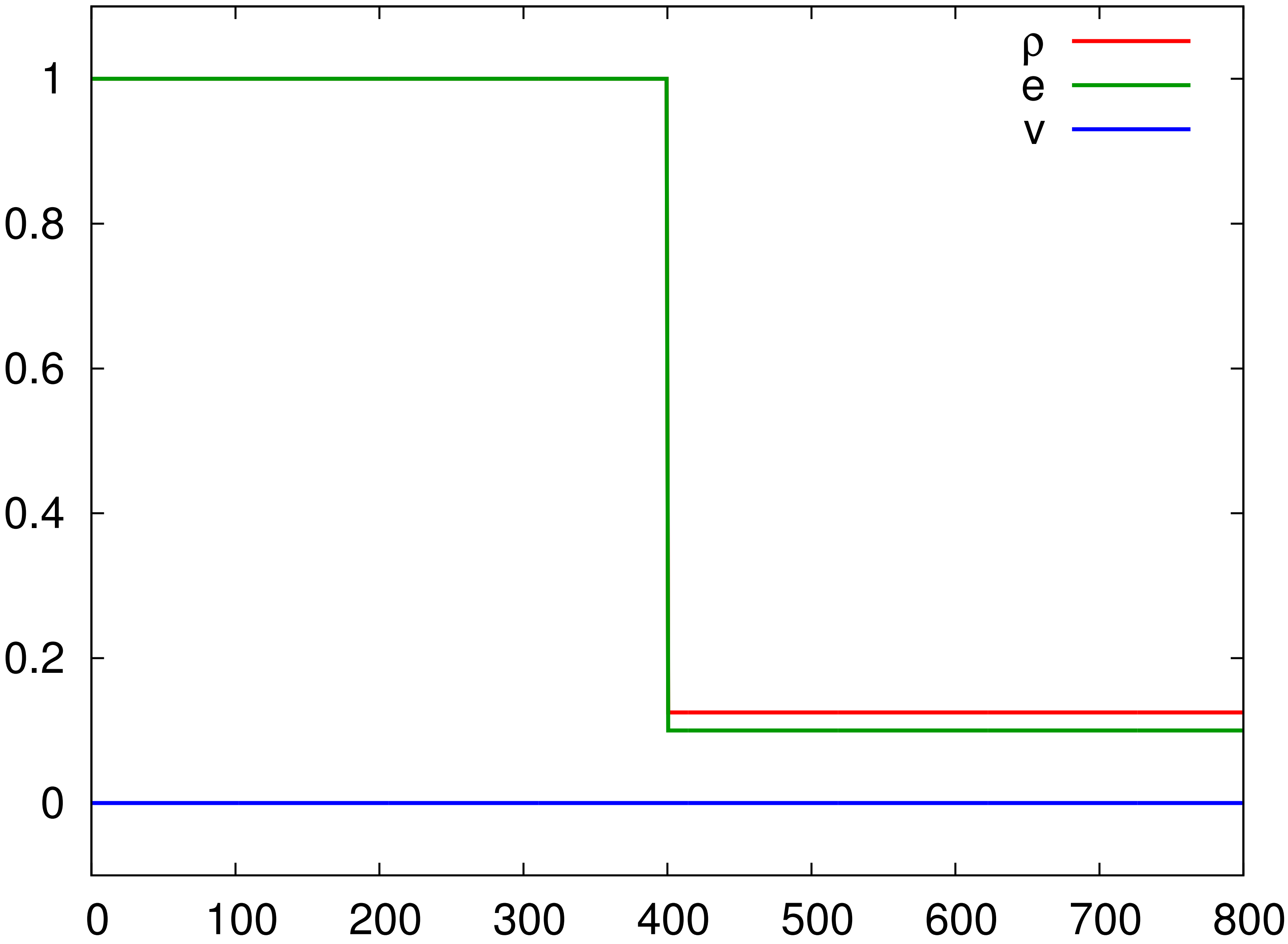
$$\frac{\partial q}{\partial t}+\frac{\partial F}{\partial x}=S$$

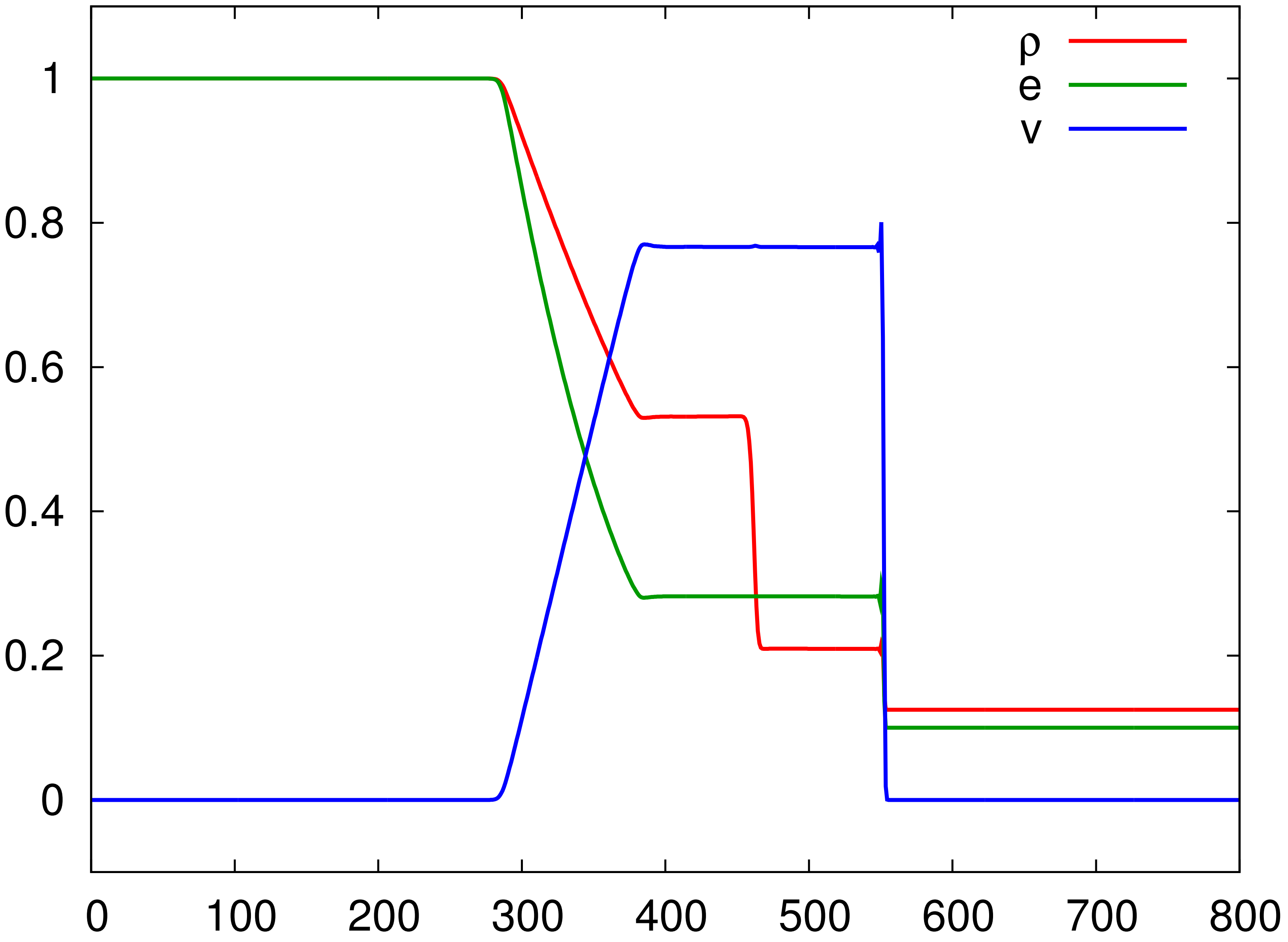
For simplicity consider incompressible fluid and no source terms:

$$\frac{\partial q}{\partial t}+v\frac{\partial q}{\partial x}=0$$

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


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