Review of Open-Source Software Based on DG Method for Simulation of Ideal Gas Flows on Unstructured Meshes

Korchagova V.N.[1,2], Sautkina S.M.[2], Fufaev I.N.[3]

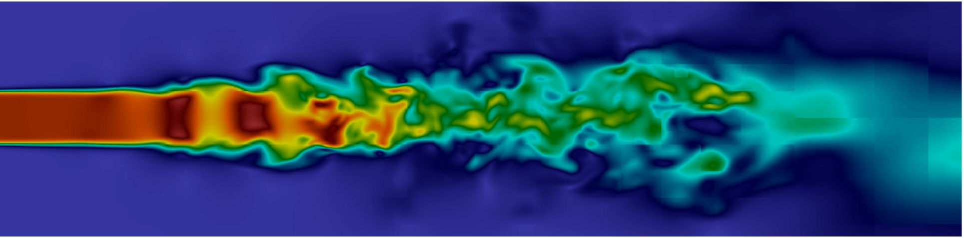
[1]*Ivannikov Institute for System Programming of the RAS*
[2]*Bauman Moscow State Technical University*
[3]*Keldysh Institute of Applied Mathematics of the RAS*

Moscow
06.12.2019

Introduction

Gas dynamics specifics

- Discontinuity of solution
- Hydro- and gas dymanic instabilities (Rayleigh–Taylor, Kelvin–Helmholtz, *etc.*)
- Different directions of disturbances propagation in subsonic and supersonic flows



Discontinuous Galerkin method

FEM + FVM = DG

Advantages

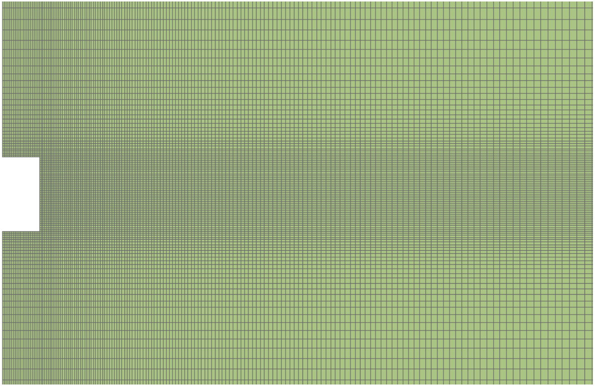
- Compact stencil
- Easy to increase the order of accuracy
- Strong theory of numerical fluxes (Lax–Friedrichs, HLL, HLLC, *etc.*)

Main difficulties

- Monotonization of solution is required nearby strong discontinuities
- Complexity of implementation

Ladenburg underexpanded jet, 2D formulation[1]

$t^* = 7.6 \cdot 10^{-4}$ s; Co = 0.5; HLL flux

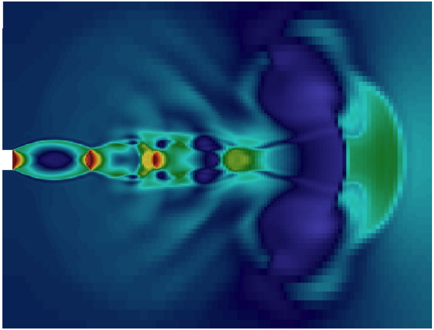


Part of mesh, 40 cells per diameter (built with SALOME)

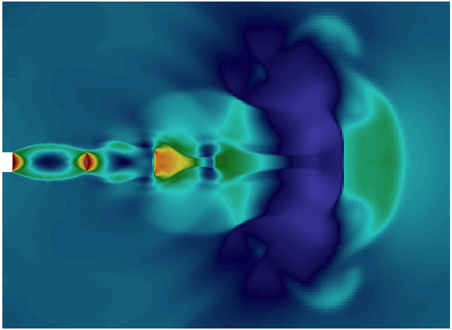
[1] https://journals.aps.org/pr/abstract/10.1103/PhysRev.76.662

Ladenburg underexpanded jet, 2D formulation

$t^* = 7.6 \cdot 10^{-4}$ s; Co $= 0.5$; HLL flux



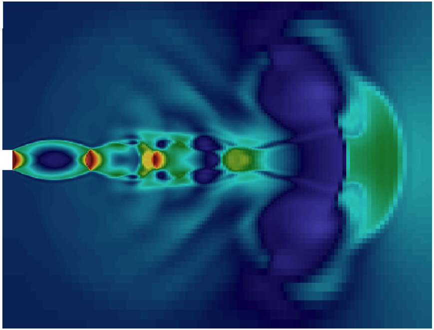
RKDG (in-house code), 40 cpd



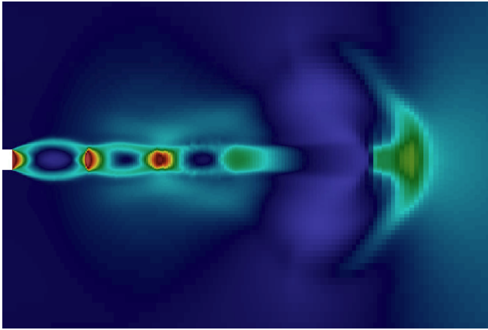
rhoPimpleCentralFoam, 80 cpd

Ladenburg underexpanded jet, 2D formulation

$t^* = 7.6 \cdot 10^{-4}$ s; Co = 0.5; HLL flux



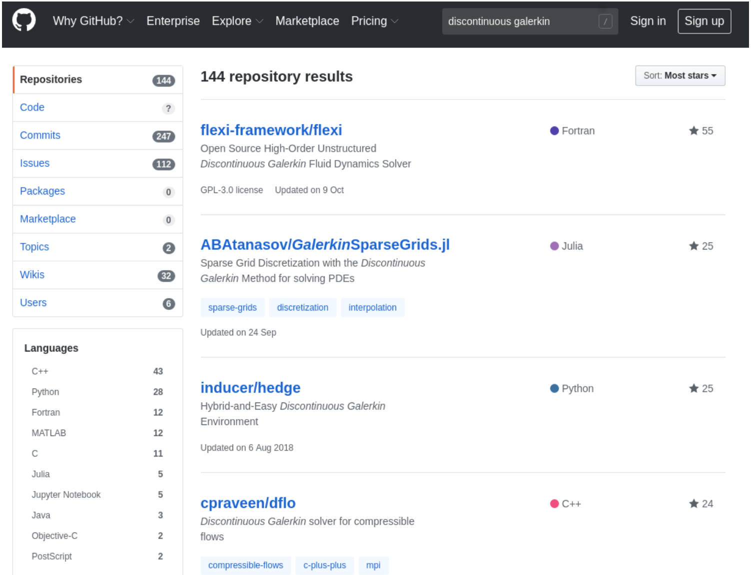
RKDG (in-house code)



OpenFOAM (rhoPimpleCentralFoam)

Density field, 40 cells per diameter

GitHub statistics

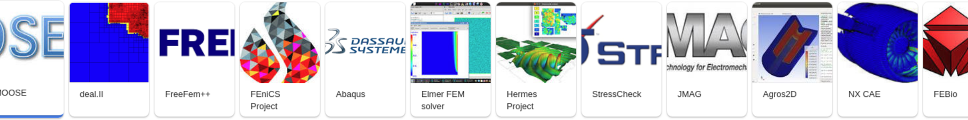


What about big codes?

"Is there any software or source code of Discontinuous Galerkin method?"[a]

[a] ResearchGate, 2014

- Last answer: 03.11.2019
- **11** codes are proposed
- **8** codes are alive now



How to choose the code?

View of a scientific user

Code "at a glance":

- contains needed features;
- documentation;
- set of tutorials;
- community (workshops, feedbacks, seminars...);
- compatibility with other formats.

First experience:

- fast and comfortable installation;
- running of tutorials;
- verification with own tests;
- readability and flexibility of code.

Large set of features

Diversity of features

- DG as one of the FEM-based approaches
- different solvers for various problems
- unstructured meshes and adaptive mesh refinement
- massive parallelism

- Common structure of codes: FEM libraries + DG support + some addons
- First steps in DG: solvers for advection equation
- Most of packages: **DG only for problems with continuous solutions**
- Compressible flow solvers are rare

Codes with compressible flow solver

NEKTAR++
SPECTRAL/HP ELEMENT FRAMEWORK

- C++
- Imperial College London and University of Utah
- 2015
- DG, Spectral Element Galerkin, Flux Reconstruction
- Numerical fluxes: Lax–Friedrichs, HLL, HLLC, AUSM, Roe, Toro
- Artificial viscosity in troubled cells; author's indicator
- High compatibility (gmsh, Star-CCM, VTK...)

Flexi

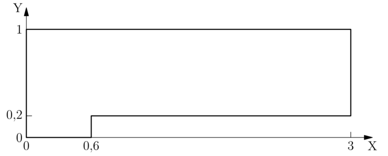
- Fortran
- Universitat Stuttgart
- 2019
- Pure DG
- Numerical fluxes: Lax–Friedrichs, HLL, HLLC, Roe
- Subcell FVM in troubled cells; indicators – Piersson, Jameson, Ducros
- Own pre- and postprocessor HOPR + conversion to different formats (gmsh, VTK...)

Test cases

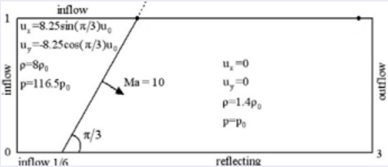
Sod problem (quasi-1D)

$$(\rho,\, u,\, v,\, w,\, p) = \begin{cases} (1,\, 0,\, 0,\, 0,\, 1), & x \le 0.5, \\ (0.125,\, 0,\, 0,\, 0,\, 0.1), & x > 0.5. \end{cases}$$

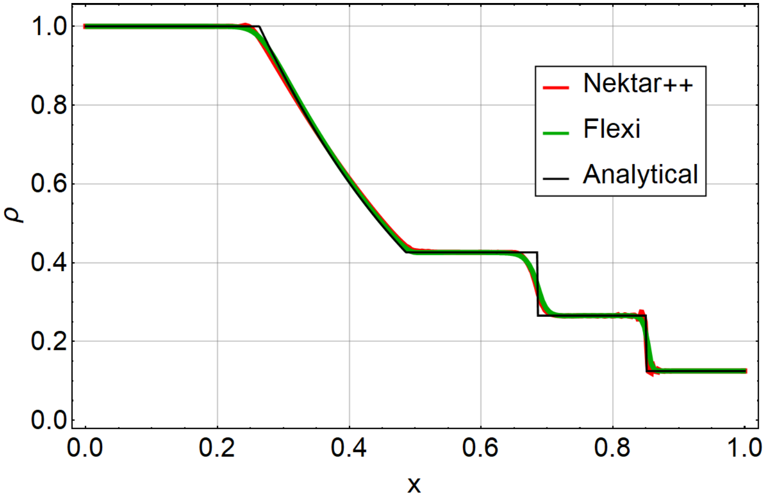
Forward step



Double Mach reflection



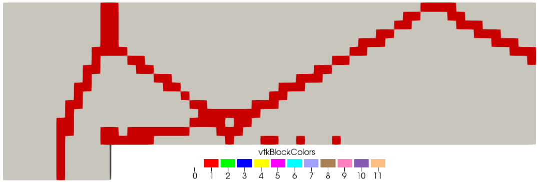
Sod problem (quasi-1D)

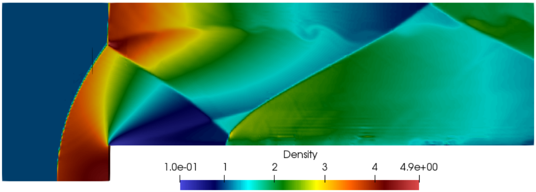


Forward Step problem

Nektar++: unstable computation

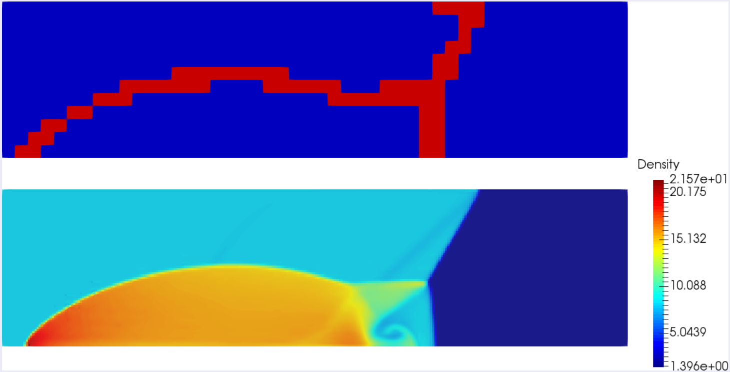
Flexi





Double Mach Reflection problem

Flexi



Summary

- Large set of codes implies DG approach
- Too much difficulties in using apart from developers:
  - installation problems: compatibility of versions of shared libraries;
  - difficulties of setup;
  - holes in documentation
- Seems promisingly for modification and improvement

Thank you for your attention!