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

Korchagova V.N.\textsuperscript{1,2}, Sautkina S.M.\textsuperscript{2}, Fufaev I.N.\textsuperscript{3}

\textsuperscript{1}Ivannikov Institute for System Programming of the RAS
\textsuperscripts{2,3}Bauman Moscow State Technical University
\textsuperscript{3}Keldysh Institute of Applied Mathematics of the RAS

Moscow
06.12.2019
Introduction

Gas dynamics specifics

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

\[ \text{FEM} + \text{FVM} = \text{DG} \]

**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\textsuperscript{1}

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

\textit{Part of mesh, 40 cells per diameter (built with SALOME)}

\textsuperscript{1}https://journals.aps.org/pr/abstract/10.1103/PhysRev.76.662
Ladenburg underexpanded jet, 2D formulation

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

**RKDG (in-house code), 40 cpd**

**rhoPimpleCentralFoam, 80 cpd**
Ladenburg underexpanded jet, 2D formulation

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

\( \text{RKDG (in-house code)} \quad \text{OpenFOAM (rhoPimpleCentralFoam)} \)

\( \text{Density field, 40 cells per diameter} \)
### GitHub statistics

**144 repository results**

<table>
<thead>
<tr>
<th>Repository</th>
<th>Description</th>
<th>Language(s)</th>
<th>Stars</th>
</tr>
</thead>
<tbody>
<tr>
<td>flexi-framework/flexi</td>
<td>Open Source High-Order Unstructured Discontinuous Galerkin Fluid Dynamics Solver</td>
<td>Fortran</td>
<td>55</td>
</tr>
<tr>
<td>ABAtanasev/GalerkinSparseGrids.jl</td>
<td>Sparse Grid Discretization with the Discontinuous Galerkin Method for solving PDEs</td>
<td>Julia</td>
<td>25</td>
</tr>
<tr>
<td>inducer/hedge</td>
<td>Hybrid-and-Easy Discontinuous Galerkin Environment</td>
<td>Python</td>
<td>25</td>
</tr>
<tr>
<td>cpraveen/dflo</td>
<td>Discontinuous Galerkin solver for compressible flows</td>
<td>C++</td>
<td>24</td>
</tr>
</tbody>
</table>
What about big codes?

"Is there any software or source code of Discontinuous Galerkin method?"*

*ResearchGate, 2014

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

---

diagram with software logos
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++**

- 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...)

---

Additional note: This slide covers the capabilities and features of NEKTAR++ and Flexi, two software tools designed for computational fluid dynamics. NEKTAR++ is a C++ library developed by Imperial College London and the University of Utah, offering a spectral/hp element framework for compressible flow simulations. Flexi, on the other hand, is a Fortran library designed by Universitat Stuttgart, focused on pure Discontinuous Galerkin (DG) methods with specific numerical fluxes and subcell Flux Reconstruction techniques. Both tools offer high compatibility with various mesh and data formats, making them versatile for use in different computational environments.
Test cases

Sod problem (quasi-1D)

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

Forward step

Double Mach reflection

\[u_x = 8.25 \sin(\pi/3) u_0, \quad u_y = 8.25 \cos(\pi/3) u_0, \quad \rho = 8 \rho_0, \quad \rho = 1.4 \rho_0, \quad \rho = 116.5 \rho_0.\]
Sod problem (quasi-1D)
Forward Step problem

Nektar++: unstable computation

Flexi
Double Mach Reflection problem

Flexi

![Double Mach Reflection problem](image-url)
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!