QGDsolver: open-source framework for development of gas and liquid models based on regularized equations

Matvey Kraposhin

Ivannikov ISP RAS OPEN Conference — 2020

11 December, 2020

Overview

1. Motivation
2. QGD/QHD approach
3. Current state of development
4. OpenFOAM framework for QGD equations
5. Verification and Validation
6. Closure

Modern demands to simulation software

Interdisciplinary

Involves resolution of different phenomena

Multiscale

Allows to resolve spatial and temporal scales with magnitude different in orders of magnitued

Scalable

Allows to use various computational systems (MPI, GPU, OpenMP, etc)

Multidimensional

Account for temporal, spatial changes and variation in physical parameters

Modern computational general purpose libraries contain solution to at least 3 of 4 demands: they are Interdisciplinary, Multiscale and Scalable

Interdisciplinary

1 process or phenomenon per model

✓ Multiscale

× High implementation costs

- methods and libraries coupling costs
- different parallel execution algorithms

× Probably high computational costs

× Problems with scalability

✓ High accuracy due to tailored methods

⇒ good for refinement

1 numerical approach to all phenomena

✓ fast implementation

✓ single parallel execution algorithm

✓ scalability

× lack of accuracy

× problems with multiscale simulations

× sometime lack of convergence

⇒ good for first estimate

Examples of universal programs

- Elmer (FEM)
- OpenFOAM (FVM)
- Phoenics (FEM)
- SU2 (FVM)
- Basilisk (FVM)
- MOOSE (FEM/DG)

Pro's of OpenFOAM

- Quasi-mathematical language for models definition
- Simplified abstraction over MPI standard
- Successfull example of interdisciplinary libraries:
    - Fluid-structure interaction
    - Conjugate heat transfer
    - Adjoint optimization

Where default OpenFOAM apps fail

- Attractor
- Toroid
- Single droplet
- Compressible subsonic viscous flows

Needed improvements to solve listed cases

- Tailored (sophisticated) numerical schemes for terms approximation
- Or: the new algorithms - QGD/QHD approach

ISP RAS

Why QGD/QHD

- applicable to compact FVM computational stencil;
- absence of iterations during time-step;
- simple numerical schemes for convection/diffusion;
- straightforward derivation of numerical algorithms;
- integration with OpenFOAM language for model definition;
- monotonic convergence.

Put equation here

1982 – QGD system derived from Boltzmann equation

1997 – QGD system formulated as conservation laws

Prof. Boris N. Chetverushkin

Prof. Tatiana G. Elizarova

Prof. Yu. V. Sheretov

From then to now

regularized or sometime Quasi Gas Dynamic (QGD) and Quasi Hydro Dynamic (QHD) equations are extensively used for various flows simulations – incompressible, compressible, multicomponent, magnetohydrodynamic, porous flows, two-phase flows – in Russia, Europe and in Keldysh Institute of Applied Mathematics of the RAS `https://keldysh.ru/`

Regularization procedure

If we regularize the transport equation for α by following steps:

1. integration and averaging of equation over small time interval τ

$$\langle\alpha\rangle = \frac{1}{\tau}\int_0^\tau \alpha dt$$

2. substitution of avarage values with instant values

$$\langle\alpha\rangle = \alpha + \tau\frac{\partial\alpha}{\partial t}$$

3. subsitution of first time derivative terms using **original equation**

$$\frac{\partial\alpha}{\partial t} = -\nabla\cdot\left(\vec{U}\alpha\right)$$

4. neglection of second time derivatives

$$\tau\frac{\partial^2\alpha}{\partial t^2} = 0, \tau^2 = 0$$

we can get new smoothed equation with controllable introduced diffusion.

ISPRAS

Pro's and Con's of QGD

Advantages of QGD algorithms

- they can work without flux limiters
- they converge monotonically to real solution
- they do not involve Rieman-solvers
- the procedure of approximation is universal for all types of flows
- they can be integrated with other OpenFOAM models
- by contrast to PISO/SIMPLE they don't involve non-orthogonal or pressure-velocity correctors
- all abovementined features make QGD algorithms a useful tool for studying transient flows phenomena

Drawbacks of QGD algorithms

- they are usually slower (3-4 times) than conventional PISO or Godunov-type methods
- additional conditions are imposed for stability criteria
- they require finer grids and smaller time steps in comparison with PISO algorithm for advection-dominated flows

QGD/QHD vs. PISO

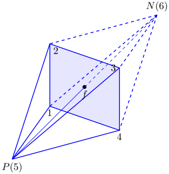


Figure 1: Geometrical scheme of stencil for numerical computation of partial derivatives on finite volume face f: P denotes center of cell to which normal of f points outward, N denotes center of cell to which normal of f points inward

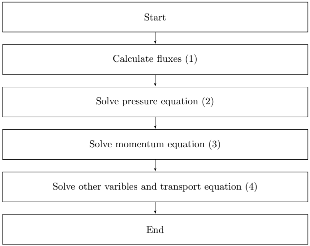


Figure 2: QHD algorithm flowchart

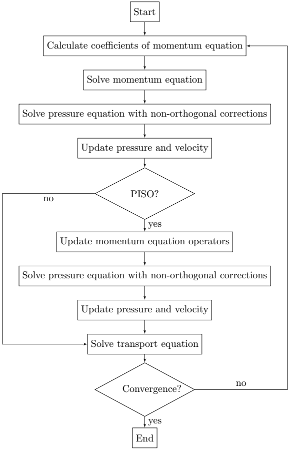


Figure 3: PIMPLE algorithm flowchart

QGD Target audience

According to stated advantages and drawbacks of QGD algorithms, they could be useful to:

- scientists, who want to solve complex set of equations, but still haven't elaborated PISO/SIMPLE or Godunov-type procedure
- researches or engineers who want to validate other methods and programs and numerical models, but they don't have analytic solution
- engineers, who want to simulate complex transient flows which could not be reproduced by PISO/SIMPLE algorithms

QGDsolver framework

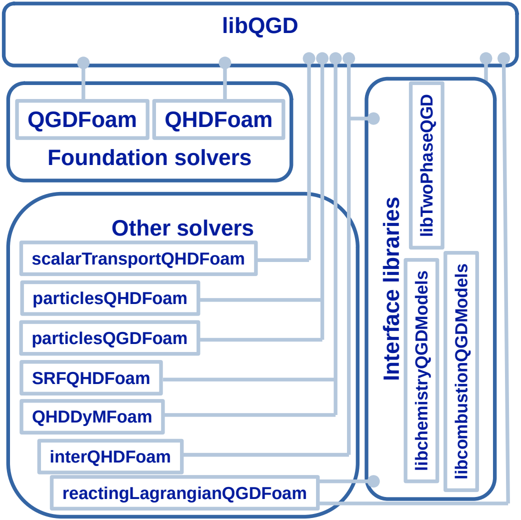
The QGDsolver framework is based on OpenFOAM+ technology and includes:

- Numerical simulation solvers for various types of flows based on the regularized equations
- τ-terms approximation library for calculation of partial derivatives on mesh faces
- Boundary conditions, QGD coefficients library
- Interfaces to some standard OpenFOAM libraries
- Tutorials

Source code stores at GitHub:
`https://github.com/unicfdlab/QGDsolver`

- Branches "digitef-dev-ABCD" are for OpenFOAM+ ver. ABCD
- Latest OpenFOAM version is OpenFOAM+ ver. 1912
- Releases are also available

QGDsolver framework structure



- Each solver implementing QGD algorithm must use libQGD library
- Two foundation solvers *QGDFoam* and *QHDFoam* show essential principles of QGD-algorithms
- Other solver could be regared as combination of foundation algorithms and OpenFOAM models
- Interface libraries are used to connect QGD solver to OpenFOAM models when interfaces have changed

OpenFOAM QGDsolver collaborators

People

- Prof. Tatiana G. Elizarova
- Maria A. Istomina
- Tatiana V. Stenina
- Daniil A. Ryazanov
- Andrey S. Epikhin
- Alexander V. Ivanov
- Kirill A. Vatutin
- Matvey V. Kraposhin

Institutions

- Keldysh institute for Applied Mathematics of the RAS
- Ivannikov Institute for System Programming of the RAS

Solvers groups

Foundation solvers

- QGDFoam – all Mach numbers compressible viscous flow of perfect gas
- QHDFoam – incompressible viscous flow with buoyancy

Miscellaneous solvers

- Mesh motion and reference frame rotation
  - QHDDyMFoam
  - SRFQHDFoam
- Transport phenomena
  - scalarTransportQHDFOam
  - mulesQHDFoam
- Flow with particles
  - particlesQGDFoam
  - particlesQHDFoam
- Multicomponent and multiphase flows
  - reactingLagrangianQGDFoam
  - interQHDFoam

General notes on case setup

0/ folder

- Since all solvers use thermophysical libraries, at least 3 fields are needed to be specified in "0" folder: "T", "U", "p"

constant/ folder

- Thermophysical libraries are used through interfaces: hePsiQGDThermo instead of hePsiThermo, heRhoQGDThermo instead of heRhoThermo
- QGD regularization parameters are set in subdictionary "QGD" of thermophysicalProperties dictionary

system/ folder

- Approximation scheme for $\tau-$terms is set in the "fvsc" subdictionary of fvSchemes dictionary
- Proportionality between maximum time step and τ is controlled via $Ctau$ keyword of controlDict dictionary

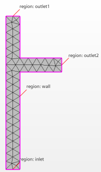
Verification and validation

- Various Riemann problems,
- backward facing step, forward facing step
- cavity flows: square cavity, skewed cavity
- nozzle flows
- tee-junction flow
- free-surface flows (steady droplet, filament collapse, etc)
- detonation combustion (with MELGUIZO GAVILANES Josue)
- flows over airfoils, blunt bodies

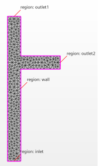
T-junction meshes

2D unstructured mesh

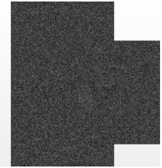
P1 – 106 cells



P4 – 590 cells



P8 – 369,318 cells

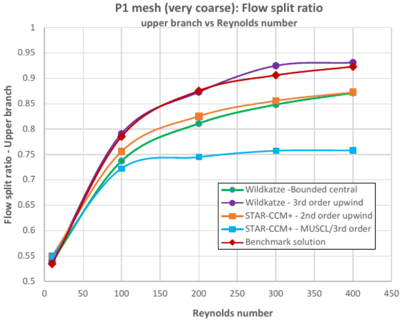


Mesh files were received along with the problem definition.

24-09-2020 Digital Solutions Inc. Japan

- Reynolds numbers for t-junction: 10, 100, 200, 300, 400
- Compared meshes: P1, P4

T-junction results

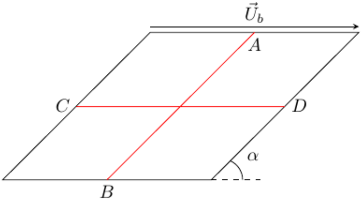


QHD behaves somewhat similar to MUSCL of STAR-CCM or van Albada of OpenFOAM

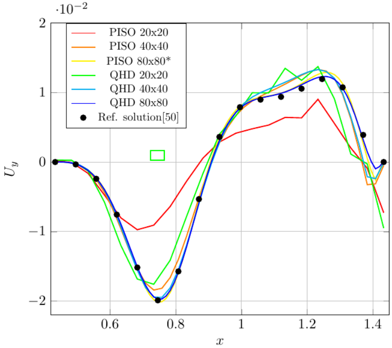
QHD results

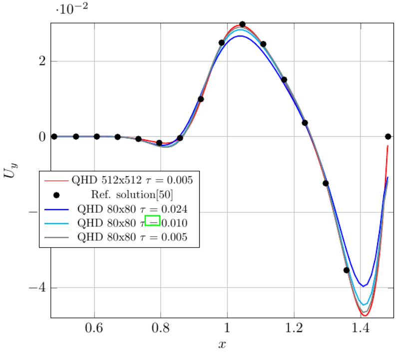
Re	Benchmark	P1, Tau=0.01	P1, Tau=0.02	**P1, Tau=0.1**	P1, Tau=0.5	P4, Tau=0.01	P4, Tau=0.02	**P4, Tau=0.1**	P4, Tau=0.5*
10	0.5344	1.85%	1.88%	**2.07%**	2.70%	1.00%	0.95%	**0.58%**	0.50%
100	0.7853	7.01%	6.97%	**6.68%**	4.89%	2.59%	2.47%	**1.32%**	3.94%
200	0.8755	11.41%	11.20%	**10.02%**	3.88%	3.77%	3.53%	**1.09%**	12.16%
300	0.9062	11.56%	11.13%	**9.11%**	2.06%	3.44%	3.07%	**0.50%**	41.60%
400	0.9232	11.77%	11.11%	**7.76%**	12.31%	3.00%	2.54%	**2.07%**	16.28%

Skewed cavity



QHD algorithm doesn't employ any loops, even for non-orthogonality correction

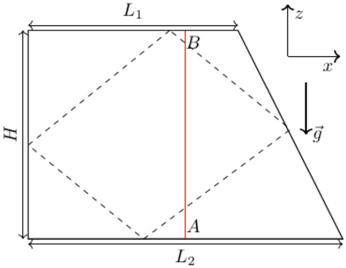




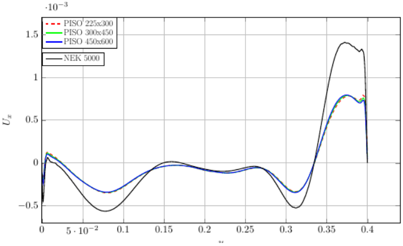
$\alpha = 30^0, Re = 1000$

$\alpha = 15^0, Re = 1000$

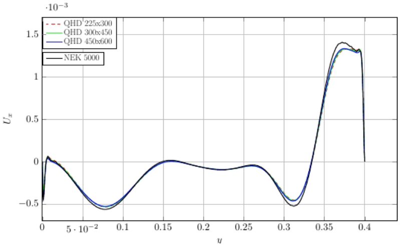
Internal wave attractor



- The default implementation of PISO algorithm doesn't reproduce the attractor
- The modified PISO converges to different solution

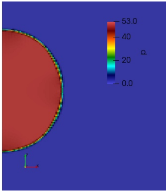


PISO algorithm

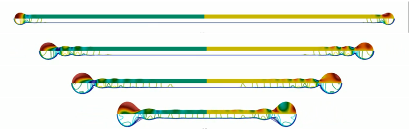


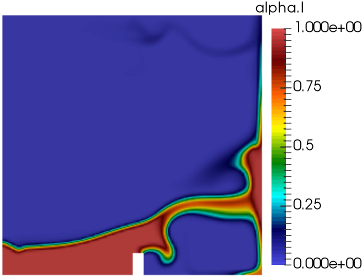
QHD algorithm

Interfacial flows



3D simulation can be viewed on YouTube
https://youtu.be/fqqSXh5t_38





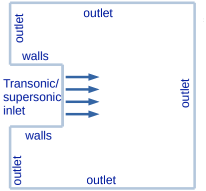
Transient high speed flows

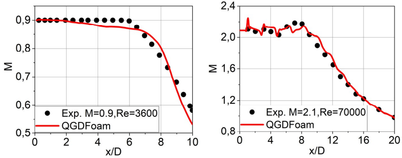
Two free jets of perfect gas with $\gamma = 1.4$ were considered:

1. $Re = 3600$
2. $Re = 70000$

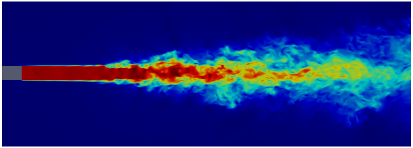
QGD settings was:

- $\alpha^{QGD} = 0.15$, $Co^{max} = 0.15$
- Sc^{QGD} was gradually reduced from 1 at the start to 0 at the end of calculation
- Ostrogradsky-Gauss appoximation of $\tau-$terms
- laminar model for $Re = 3600$ was used
- Smagorinsky model for $Re = 70000$ was used
- Computational mesh with 33 mln cells was used in both cases





(a) (b)



ISPRAS

QGDsolver github page

https://github.com/unicfdlab/QGDsolver

QGDsolver telegram group

https://t.me/qgd_qhd

QGDsolver Training Tracks

- https://github.com/unicfdlab/TrainingTracks/tree/master/OpenFOAM/QHDFoam-OFv1912
- **Q**GDFoam track is under development

The study was supported by

the Russian Science Foundation, Project No. 19-11-00169