Library Updates: So what has been going on in 2016/17?

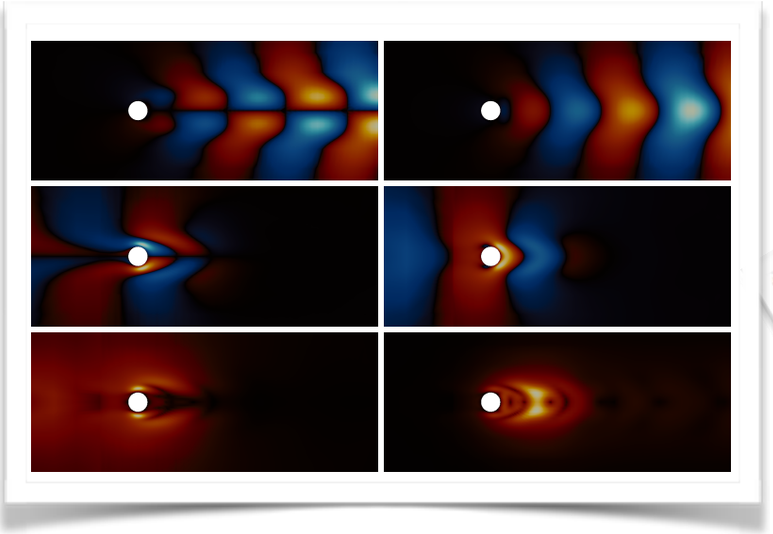
1406 Commits

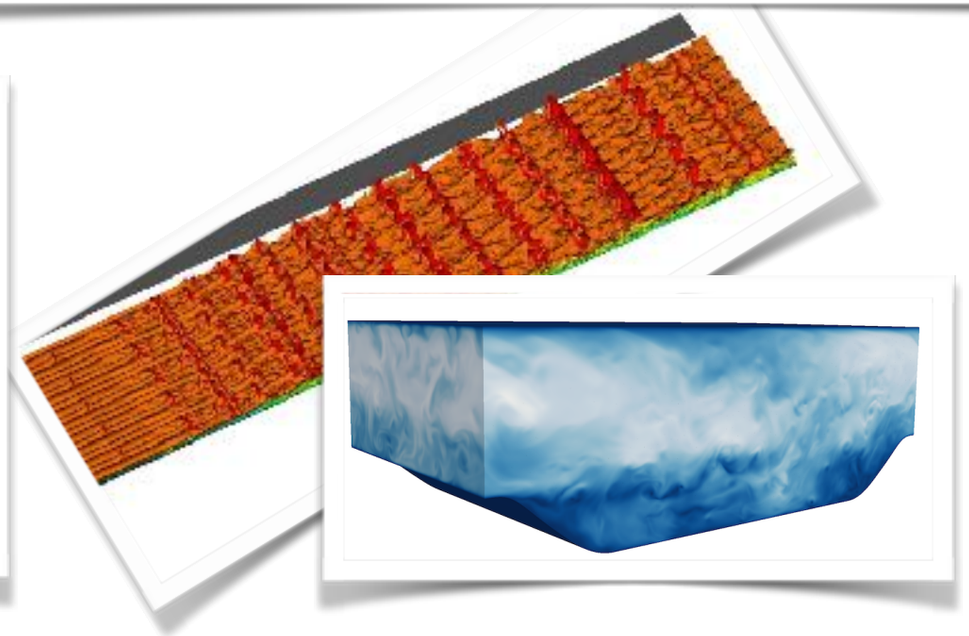
153K Lines of code added

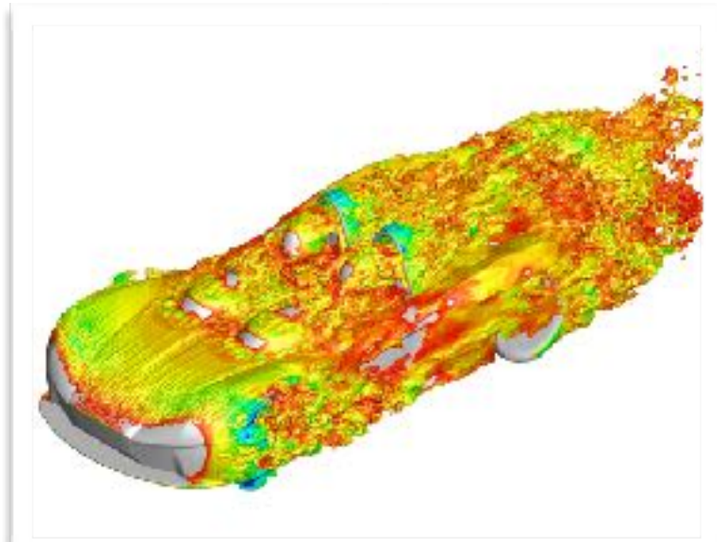
60K Lines of code deleted

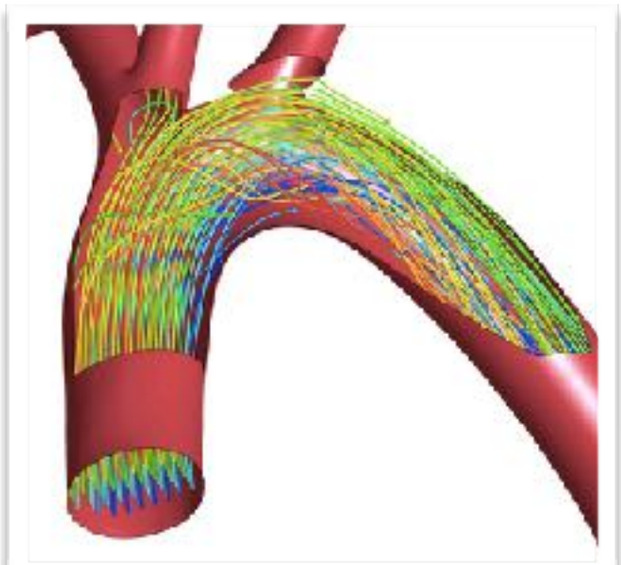
Imperial College London

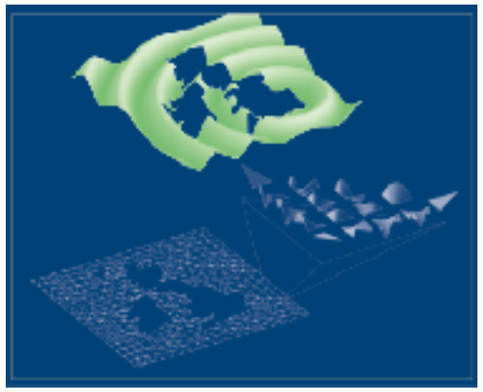
Nektar++ Applications

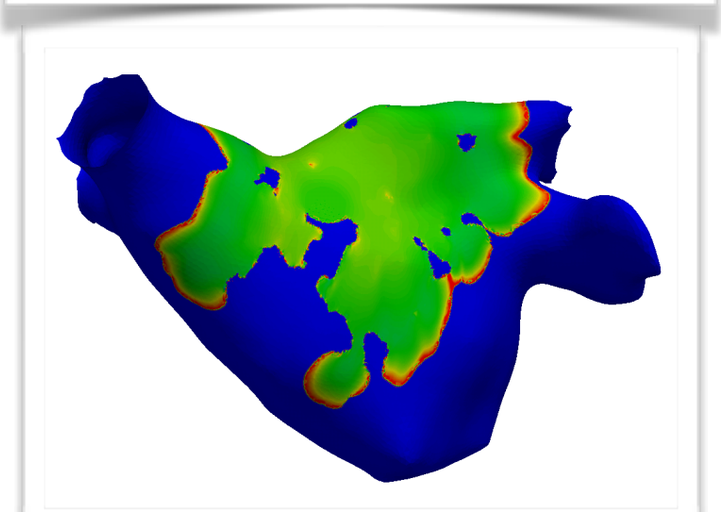


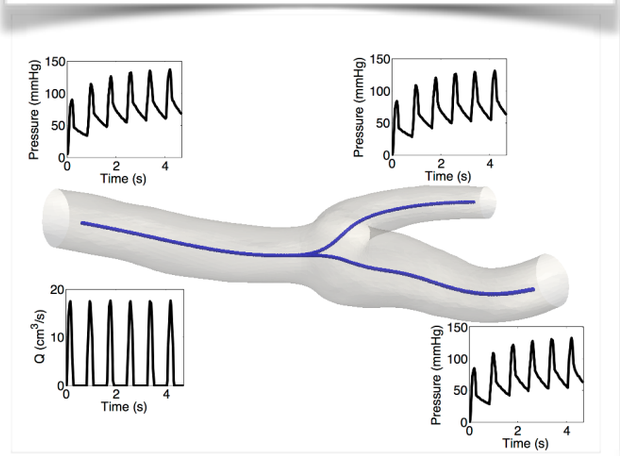


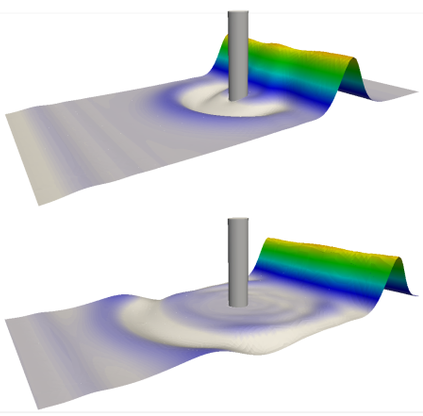


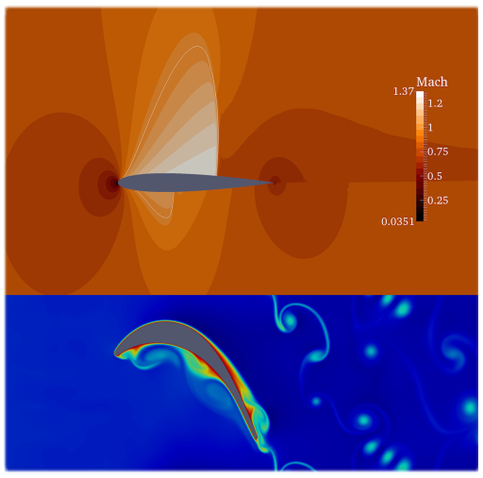


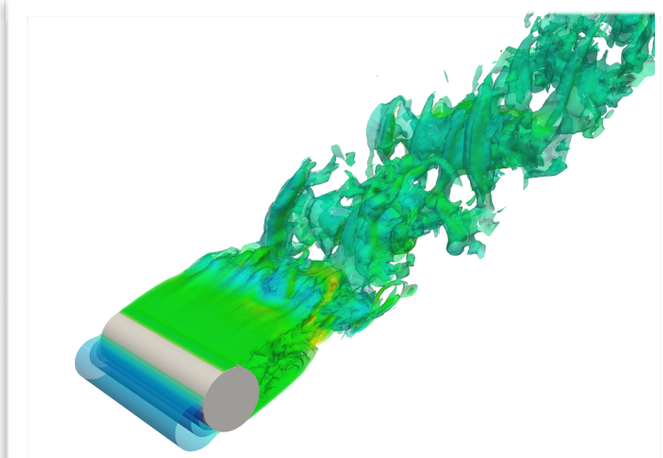






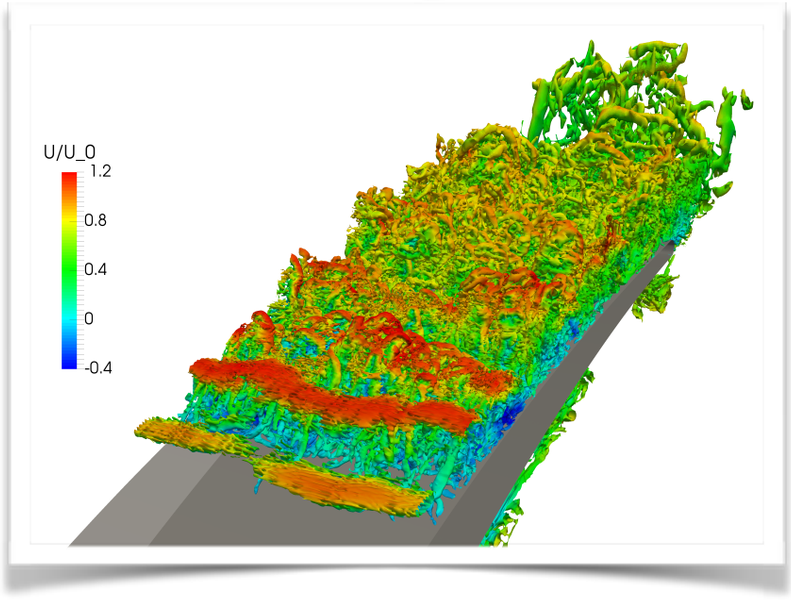


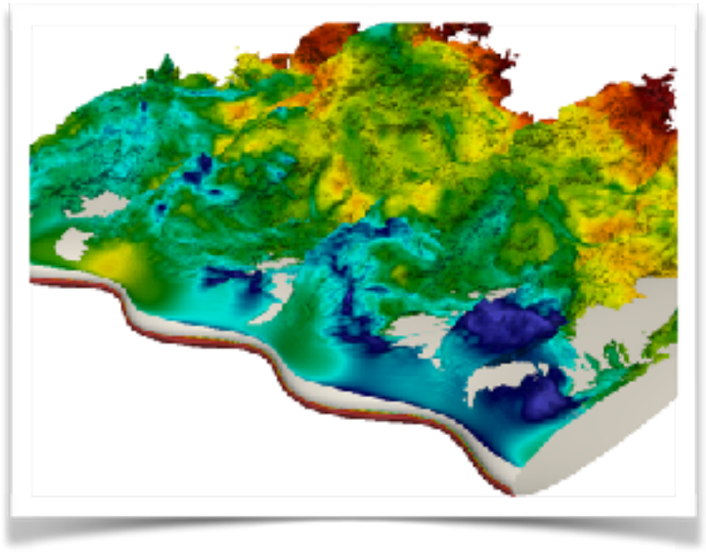


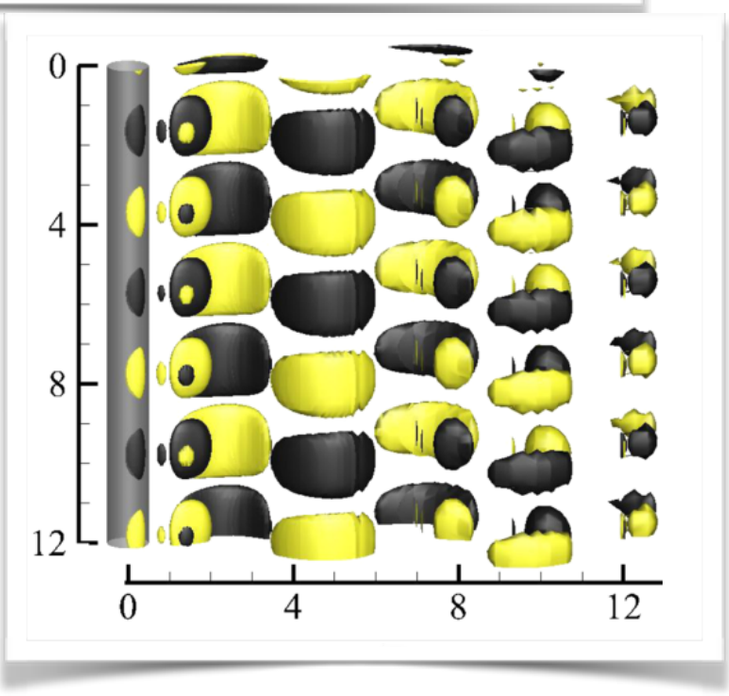


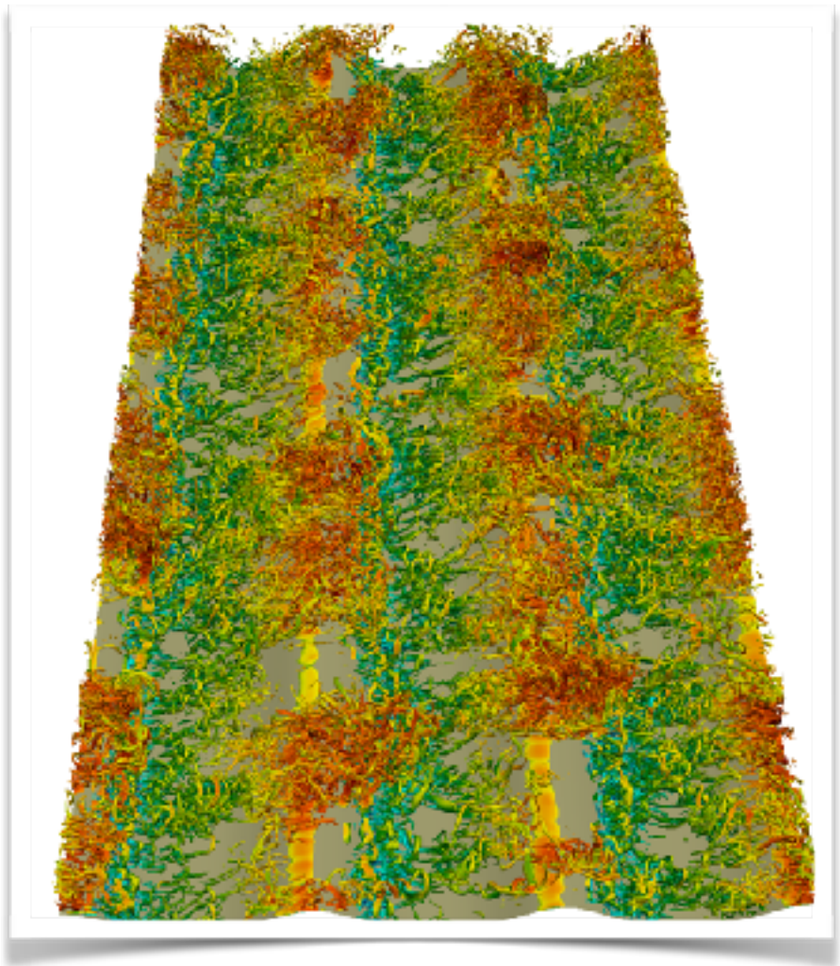
Imperial College London

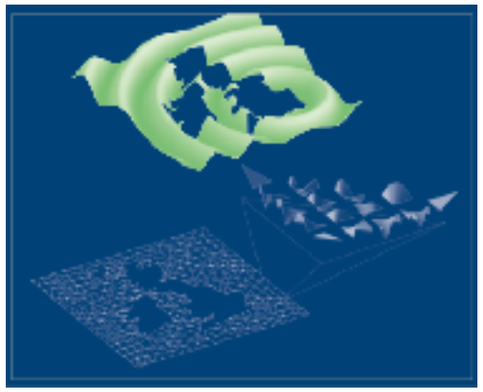
Nektar++ Applications

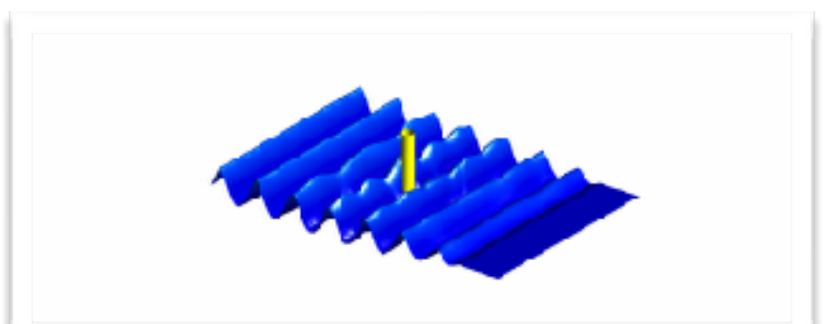


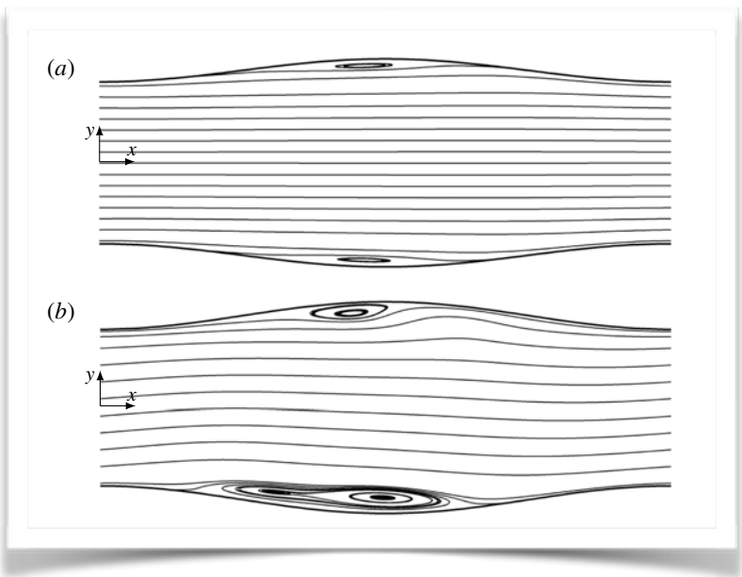


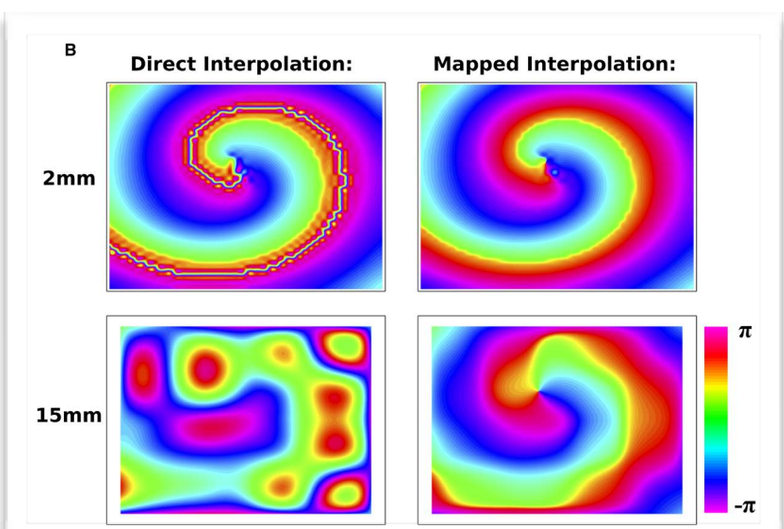












Change Log

https://gitlab.nektar.info/nektar/nektar/blob/master/CHANGELOG.md
$NEKTARHOME/CHANGELOG.md

v4.3.5

Library:

- Fix bug in DG with hybrid meshes (!694)
- Fix issue with parallel output (!699)
- Fix performance issue with iterative full solver (!693)
- Enforced precision on history point output (!706)

Documentation

- Update build instructions in user guide for Windows (!692)

Tester

- Fix bug in tester when no parameters specified for test executable (!701)

v4.3.4

Library:

- Fix performance issue with `v_ExtractDataToCoeffs` for post-processing of large simulations (!672)
- Added additional assertions to ensure homogeneous simulations have an even number of planes per process (!666)
- Fix compilation with NEKTAR_USE_MESHGEN option
- Fix IterativeFull solver in parallel (!685)
- Fix error message for missing fld file (!689)

IncNavierStokesSolver:

- Fix 2nd order time-integration for VCSMapping (!687)

Change Log v4.4

NekMesh:

- Modify curve module to allow
- Add STL surface writer modul
- New module for inserting an a
- Add curve projection routines
- Extensive clean-up of NekMe
- Improvements to mesh lineari
- Add support for Gmsh high-o
- Move CAD classes to factory
- Add module to check topolog
- Add option to `linearise` mo
- Add reader for Nek5000 mes
- Add option to `linearise` to
- Add flag to `insertsurface`
- Bug fix to get two meshgen ro
- Remove libANN in deference
- Refactor library to use NekMe
- Add `varopti` process modul
- Add a mesh extract option to
- 2D to 3D mesh extrusion mod
- Add new two-dimensional me
- Fix inverted boundary layer in 2D (!736)
- More sensible element sizing with boundary layers in 2D (!736)
- Change variable names in mcf file to make more sense (!736)
- Fix issues in varopti module so that in can be compiled without meshgen on (!736)
- Replace LAPACK Eigenvalue calculation with handwritten function in varopti (!738)
- Improved node-colouring algorithm for better load-balancing in varopti (!738)
- Simplified calculation of the energy functional in varopti for improved performance (!738)

ADRSolver:

- Add a projection equation system for C^0 proj

APESolver:

- Use a continuous basefield projection and re
- Added ability to compute CFL number (!664)
- Output Sourceterm (!664)
- Use the Forcing framework to define source

IncNavierStokesSolver:

- Add ability to simulate additional scalar fields
- Improve performance when using homogene
- Fix linearised advection for full 3D cases (!70
- Added a weak pressure formulation following
- Added a convective like outflow boundary co
- Added the ability to specifiy Womersley bour

CardiacEPSolver:

- Added a Python translator utility to generate

FieldConvert:

- Allow equi-spaced output for 1D and 2DH1D
- Update quality metric to include scaled Jaco
- Allow multiple XML files to be specified in Int
- Fix issues with isocontour module (!719)
- Fix issue with interpolator routine (!746)

FieldConvert:

- Move all modules to a new library, FieldUtils, to support post-processing during simulations (!589)
- Add module to stretch homogeneous direction (!609)
- Add module to add composite ID of elements as a field (!674)
- Add reader for Nek5000 field files (!680)

Tester:

- Fix output not displayed on segfault or system error (!745)

Library:

- Add support for variable polynomial order for 3D simulations with continuous Galerkin discretisation (!604)
- Bump version of gsmpi to suppress autotuning output unless `--verbose` is specified (!652)
- Add support for variable polynomial order with periodic boundary conditions (!658)
- Statistics are now printed for lowest level of multi-level static condensation (!656)
- Sped up interpolataion from pts files and fixed parallel pts import (!584)
- Increased required boost version to 1.56.0 (!584)
- New FieldUtils library allows support for most `FieldConvert` post-processing operations during simulation using
- Adjust CMake dependencies to reduce compile time (!671)
- Homogeneous1D dealiasing improvements (!622)
- Add support for HDF5 as an alternative output to XML-based output, including refactoring of FieldIO, improvemen conditions (!615)
- Allow expansions to be loaded directly from field file (!617)
- New options for load balancing (DOF or BOUNDARY) in mesh partitioner (!617)
- Rework nodal utilities to support nodal prismatic elements (!660)
- Update Body/Field forces at each timestep (!665)
- Update nodalutil to include quad and hex elements and introduce SPI nodal points (!696)
- Add ability to restart time-averaging and Reynolds stresses from checkpoint file (!678)
- Extend ExtractDataToCoeffs to support interpolation between basis types for quads and hexahedra (!682)
- Enabled MUMPS support in PETSc if a Fortran compiler was found and added 3D support to the Helmholtz smoot
- Fix bug in `Vmath::FillWhiteNoise` which caused `ForcingNoise` to have a repeated pattern (!718)
- Fix bug in the calculation of the RHS magnitude in CG solver (!721)
- Fix bug in MPI detection for recent CMake on OS X (!725)
- Fix bug in CMake Homebrew and MacPorts detection for OS X (!729)
- Fix bug in FieldUtils when using half mode expansions (!734)
- Do not read the same fld/pts files again for every variable (!670)
- Fix bug in CMake PETSc detection for Ubuntu 16.04/Debian 9 (!735)
- Fix warnings with Intel compiler (!742)

Change Log v4.4.1

v4.4.1

Library

- Remove m_offset_elmt_id and GetOffsetElmtId which fixed problems in 2D when quad elements are listed before tri elements (!758)
- Remove the duplicate output of errorutil (!756)
- Fix BLAS CMake dependencies (!763)
- Fix interpolation issue with Lagrange basis functions (!768)
- Fix issue with average fields not working with different polynomial order fields (!776)
- Fix Hdf5 output in FilterFieldConvert (!781)
- Fixed extreme memory consumption of Interpolator when interpolating from pts to fld or between different meshes (!783)
- Fix deadlock with HDF5 input (!786)

FieldConvert:

- Fix issue with field ordering in the interppointdatatofld module (!754)
- Fix issue with FieldConvert when range flag used (!761)
- Fix issue when using output-points combined with nequispaced (!775)
- Fix equispacedoutput for 3DH1D with triangles (!787)

NekMesh:

- Fix memory consumption issue with Gmsh output (!747, !762)
- Rework meshing control so that if possible viewable meshes will be dumped when some part of the system fails (!758)
- Add manifold meshing option (!756)
- Fix issue with older rea input files (!765)

IncNavierStokesSolver

- Fix an initialisation issue when using an additional advective field (!779)

Change Log v4.4

NekMesh:

- Modify curve module to allow
- Add STL surface writer modul
- New module for inserting an e
- Add curve projection routines
- Extensive clean-up of NekMe
- Improvements to mesh lineari
- Add support for Gmsh high-o
- Move CAD classes to factory
- Add module to check topolog
- Add option to `linearise` mo
- Add reader for Nek5000 mes
- Add option to `linearise` to
- Add flag to `insertsurface`
- Bug fix to get two meshgen ro
- Remove libANN in deference t
- Refactor library to use NekMe
- Add `varopti` process modul
- Add a mesh extract option to
- 2D to 3D mesh extrusion mod
- Add new two-dimensional me
- Fix inverted boundary layer in 2D (!736)
- More sensible element sizing with boundary layers in 2D (!736)
- Change variable names in mcf file to make more sense (!736)
- Fix issues in varopti module so that in can be compiled without meshgen on (!736)
- Replace LAPACK Eigenvalue calculation with handwritten function in varopti (!738)
- Improved node-colouring algorithm for better load-balancing in varopti (!738)
- Simplified calculation of the energy functional in varopti for improved performance (!73

ADRSolver:

- Add a projection equation system for C^0 projectio

APESolver:

- Use a continuous basefield projection and revert t
- Added ability to compute CFL number (!664)
- Output Sourceterm (!664)
- Use the Forcing framework to define source terms

IncNavierStokesSolver:

- Add ability to simulate additional scalar fields (!62
- Improve performance when using homogeneous c
- Fix linearised advection for full 3D cases (!708)
- Added a weak pressure formulation following Gue
- Added a convective like outflow boundary conditi
- Added the ability to specifiy Womersley boundary

CardiacEPSolver:

- Added a Python translator utility to generate cell r

FieldConvert:

- Allow equi-spaced output for 1D and 2DH1D fields
- Update quality metric to include scaled Jacobian
- Allow multiple XML files to be specified in InterpFi
- Fix issues with isocontour module (!719)
- Fix issue with interpolator routine (!746)

FieldConvert:

- Move all modules to a new library, FieldUtils, to support post-processing during simulations (!589)
- Add module to stretch homogeneous direction (!609)
- Add module to add composite ID of elements as a field (!674)
- Add reader for Nek5000 field files (!680)

Tester:

- Fix output not displayed on segfault or system error (!745)

v4.4.0

Library:

- Add support for variable polynomial order for 3D simulations with continuous Galerkin discretisation (!604)
- Bump version of gsmpi to suppress autotuning output unless `--verbose` is specified (!652)
- Add support for variable polynomial order with periodic boundary conditions (!658)
- Statistics are now printed for lowest level of multi-level static condensation (!656)
- Sped up interpolataion from pts files and fixed parallel pts import (!584)
- Increased required boost version to 1.56.0 (!584)
- New FieldUtils library allows support for most `FieldConvert` post-processing operations during simulation u
- Adjust CMake dependencies to reduce compile time (!671)
- Homogeneous1D dealiasing improvements (!622)
- Add support for HDF5 as an alternative output to XML-based output, including refactoring of FieldIO, improv conditions (!615)
- Allow expansions to be loaded directly from field file (!617)
- New options for load balancing (DOF or BOUNDARY) in mesh partitioner (!617)
- Rework nodal utilities to support nodal prismatic elements (!660)
- Update Body/Field forces at each timestep (!665)
- Update nodalutil to include quad and hex elements and introduce SPI nodal points (!696)
- Add ability to restart time-averaging and Reynolds stresses from checkpoint file (!678)
- Extend ExtractDataToCoeffs to support interpolation between basis types for quads and hexahedra (!682)
- Enabled MUMPS support in PETSc if a Fortran compiler was found and added 3D support to the Helmholtz s
- Fix bug in `Vmath::FillWhiteNoise` which caused `ForcingNoise` to have a repeated pattern (!718)
- Fix bug in the calculation of the RHS magnitude in CG solver (!721)
- Fix bug in MPI detection for recent CMake on OS X (!725)
- Fix bug in CMake Homebrew and MacPorts detection for OS X (!729)
- Fix bug in FieldUtils when using half mode expansions (!734)
- Do not read the same fld/pts files again for every variable (!670)
- Fix bug in CMake PETSc detection for Ubuntu 16.04/Debian 9 (!735)
- Fix warnings with Intel compiler (!742)

Framework Layout

IncNavierStokes **CompressibleFlow** **ADR** **CardiacEP** **...**

SolverUtils

Core Nektar++ libraries

StdRegions **SpatialDomains** **LocalRegions**

MultiRegions **Collections** **Global Mappings**

LibUtilities

Quadrature, bases, partitioning, input/output, linear algebra, interpreter, FFT, ...

Boost **Metis** **TinyXML** **Gslib** VTK PETSc ARPACK

FFTW Scotch Zlib QT

Framework Layout

IncNavierStokes | **CompressibleFlow** | **ADR** | **ImageWarping** | **...**

SolverUtils

Core Nektar++ libraries

NekMesh

MultiRegions | **Collections** | **Global Mappings**

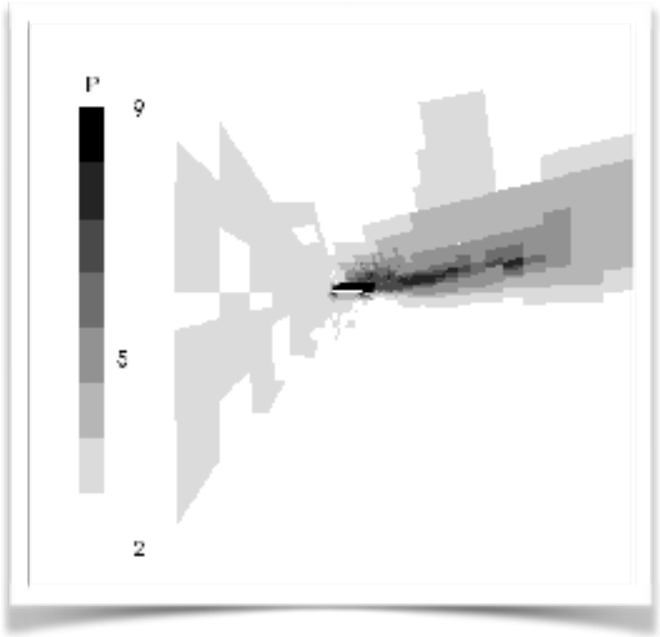
NekMeshUtils | **FieldUtils**

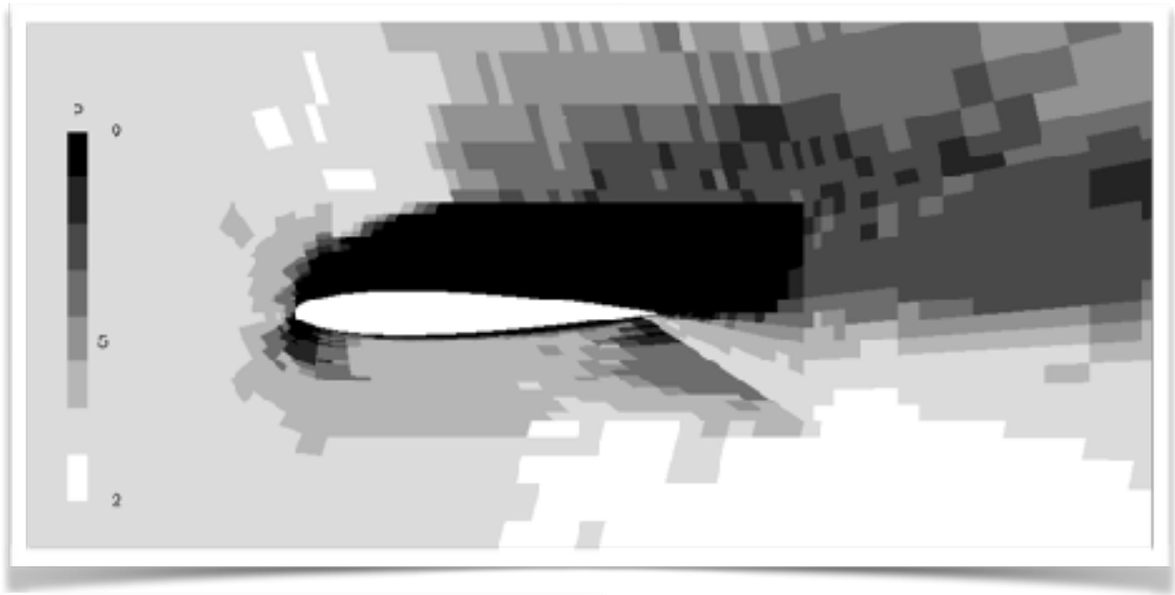
Library Developments:

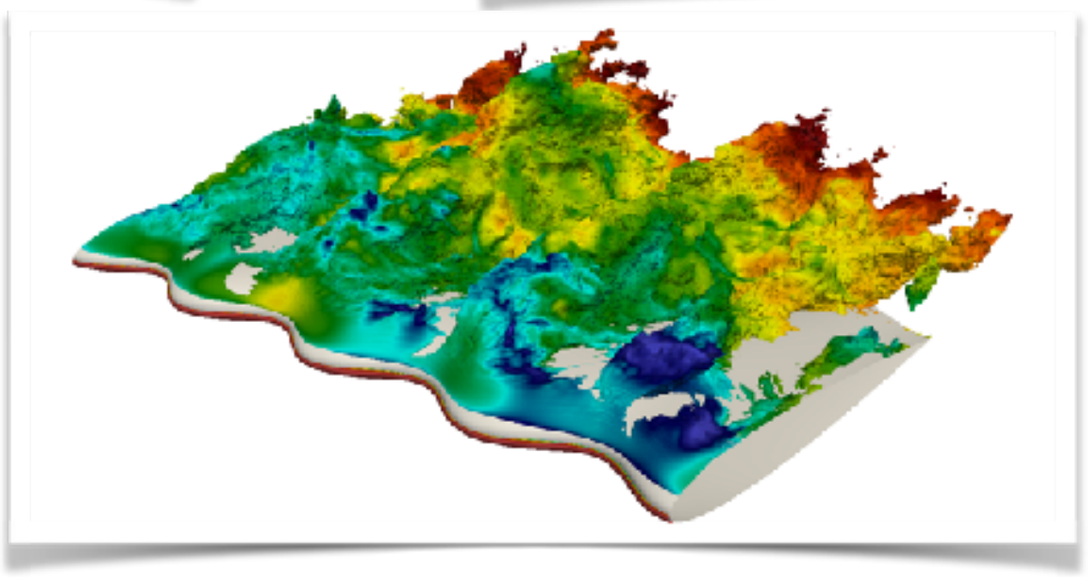
- Variable p (primarily 2, 2.5D)
  - 3D variable polynomial order (not preconditioner)
- HDF5 output
- Incompressible Navier Stokes Solvers
- Tutorials

Library Developments: Adaptive Polynomial Order

- Example: Naca0012 with Re=50,000 and alpha=15 (P_{min} = 2, P_{max} = 9)







- Variable P available in 3D but pyramids and preconditioners only available in branch

Library Developments: Hdf5 Field files

Default parallel output is a directory: myjob_fld /or myjob_chk/

Info.xml P0000000.fld P0000001.fld P0000002.fld P0000003.fld

Have a challenge of generating too many files:
8000 cores for 100 dumps leads to 800K files!

../IncNavierStokesSolver -h

-i [--io-format] arg Default input/output format (e.g. Xml, Hdf5)

../IncNavierStokesSolver -i hdf5 myrun.xml

Incompressible Navier Stokes developments

- Fixed linearised stability
- Weak pressure Poisson system
- Dong Outflow / mixed pressure
- Womersley BCs

Pressure Poisson System

Guermond & Shen 2003 JCP

$$\frac{\gamma_0 \bar{\mathbf{u}}^{n+1} - \mathbf{u}^+}{\Delta t} + \nabla p^{n+1} + \nu(\nabla \times \nabla \times \mathbf{u})^* - \mathbf{N}^* = 0$$

$$\nabla \cdot \bar{\mathbf{u}}^{n+1} = 0$$

*where * represents extrapolation and*
+ represents backward differencing

Doting with $\nabla\phi$ and using some identities:

$$\int_\Omega \nabla p^{n+1} \cdot \nabla\phi d\Omega = \int_\Omega \frac{\hat{\mathbf{u}}}{\Delta t} \cdot \nabla\phi d\Omega$$

$$- \int_\Gamma \phi \left[\nu(\nabla \times \nabla \times u)^*\right] \cdot \mathbf{n} dS$$

$$\hat{\mathbf{u}} = \mathbf{u}^+ + \Delta t \mathbf{N}^*.$$

Imperial College London

This option is implemented as VCSWeakPessure

Pressure Poisson System

Guermond & Shen 2003 JCP

$$\frac{\gamma_0 \bar{\mathbf{u}}^{n+1} - \mathbf{u}^+}{\Delta t} + \nabla p^{n+1} + \nu(\nabla \times \nabla \times \mathbf{u})^* - \mathbf{N}^* = 0$$
$$\nabla \cdot \bar{\mathbf{u}}^{n+1} = 0$$

*where * represents extrapolation and*
+ represents backward differencing

Dotting with $\nabla\phi$, applying the divergence theorem and using some identities:

$$\int_\Omega \nabla p^{n+1} \cdot \nabla\phi \, d\Omega = \int_\Omega \phi \nabla \cdot \left(-\frac{\hat{\mathbf{u}}}{\Delta t} \right) d\Omega + \int_\Gamma \phi \left[\frac{\hat{\mathbf{u}} - \gamma_0 \bar{\mathbf{u}}^{n+1}}{\Delta t} - \nu(\nabla \times \nabla \times \mathbf{u})^* \right] \cdot \mathbf{n} \, dS,$$

$$\hat{\mathbf{u}} = \mathbf{u}^+ + \Delta t \mathbf{N}^*.$$

Imperial College London

This option is implemented as VelocityCorrectionScheme

Incompressible Navier Stokes developments

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

A convective-like energy-stable open boundary condition for simulations of incompressible flows

S. Dong

Center for Computational & Applied Mathematics, Department of Mathematics, Purdue University, USA

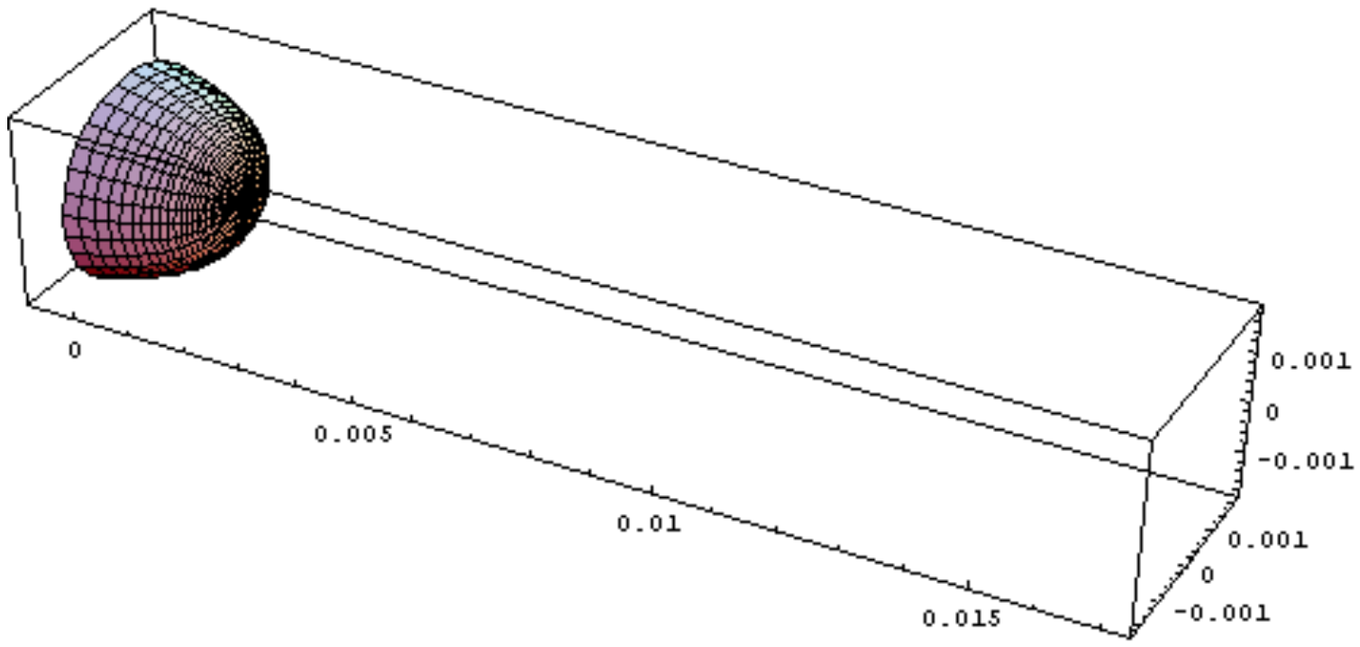
$$\frac{\partial p^{n+1}}{\partial n} + \frac{1}{\nu D_0} p^{n+1} = \mathbf{n} \cdot \left(\mathbf{f}^{n+1} - \mathbf{u}^{*,n+1} \cdot \nabla \mathbf{u}^{*,n+1} \right) - \nu \mathbf{n} \times \boldsymbol{\omega}^{*,n+1} - \frac{1}{\nu D_0} \left[\mathbf{f}_b^{n+1} \cdot \mathbf{n} + \mathbf{n} \cdot \mathbf{E}(\mathbf{n}, \mathbf{u}^{*,n+1}) - \nu \mathbf{n} \cdot \nabla \mathbf{u}^{*,n+1} \cdot \mathbf{n} \right], \quad \text{on } \partial\Omega_o.$$

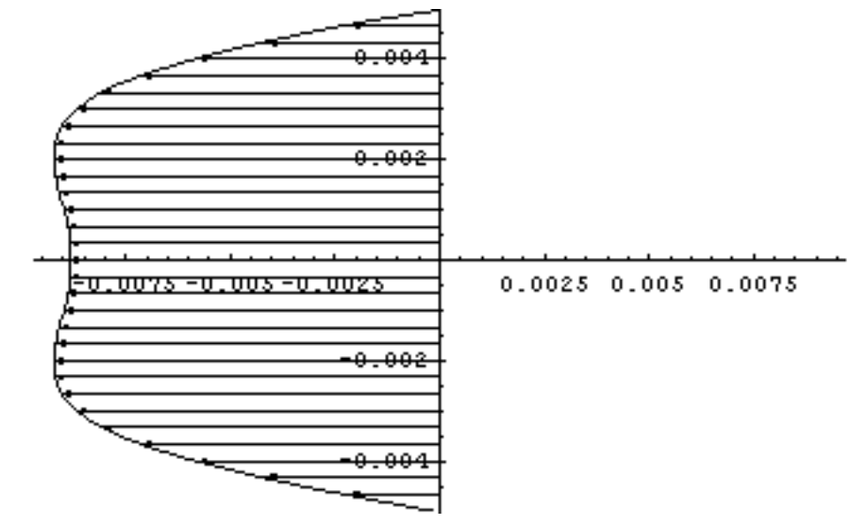
$$\frac{\partial \mathbf{u}^{n+1}}{\partial n} + \frac{\gamma_0 D_0}{\Delta t} \mathbf{u}^{n+1} = \frac{1}{\nu} \left[\mathbf{f}_b^{n+1} + \mathbf{E}(\mathbf{n}, \mathbf{u}^{*,n+1}) + p^{n+1} \mathbf{n} - \nu (\nabla \cdot \mathbf{u}^{*,n+1}) \mathbf{n} \right] + \frac{D_0}{\Delta t} \hat{\mathbf{u}}, \quad \text{on } \partial\Omega_o,$$

- *Robin (weak) boundary conditions on pressure and velocity*
- *Details in user guide*

Imperial College London

Incompressible Navier Stokes developments





http://lions.math.hr/tok-kroz-cijev/tekstovi/womersley/womersley_en.html

$$w(r,t) = A_0(1-(r/R)^2) + \sum_{n=1}^{N} \tilde{A}_n[1 - \frac{J_0(i^{3/2}\alpha_n r/R)}{J_0(i^{3/2}\alpha)}]e^{i\omega_n t}$$

- *Womersley Boundary Conditions for pulsatile flow*
- *Details in user guide*

Imperial College London

Tutorials

Numerical Differentiation

Tutorials

January 11, 2017

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, US

Global Stability Analysis: Flow over a Backward-Facing Step

Tutorials

January 16, 2017

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, USA

Two-dimensional high-order mesh generation

Tutorials

January 30, 2017

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, USA

Advection Solver

Tutorials

January 16, 2017

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, USA

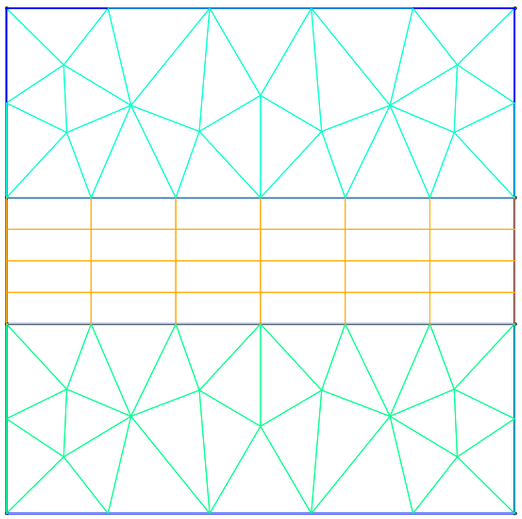
Quasi-3D Computation of the Taylor-Green Vortex Flow

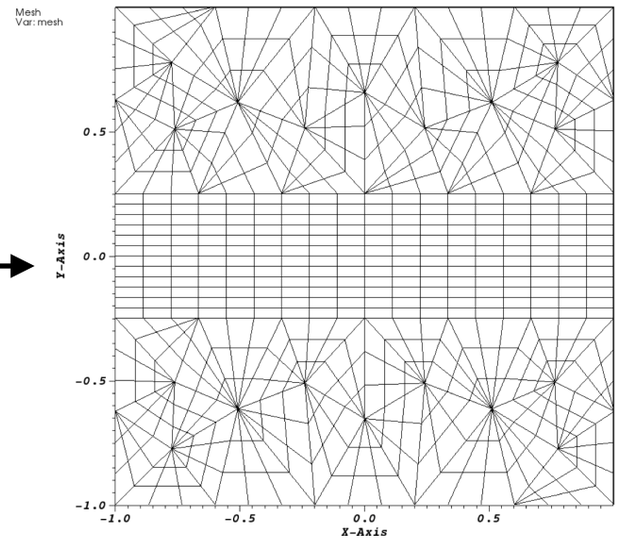
Tutorials

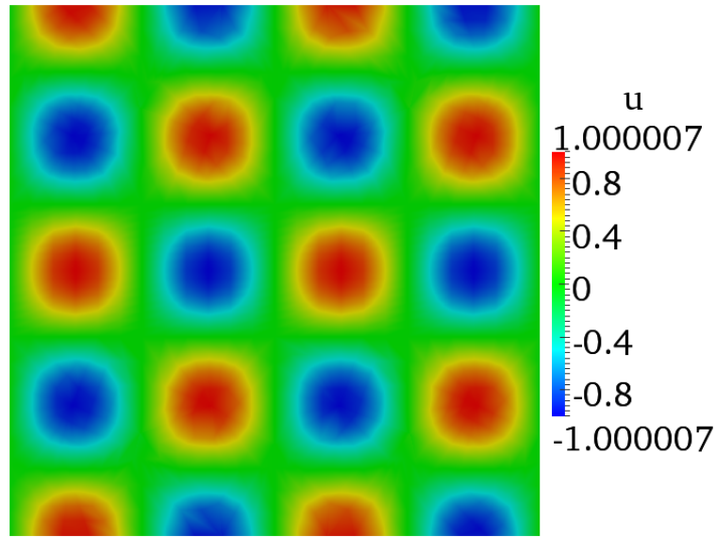
January 16, 2017

Department of Aeronautics, Imperial College London, UK
Scientific Computing and Imaging Institute, University of Utah, USA

Advection problems





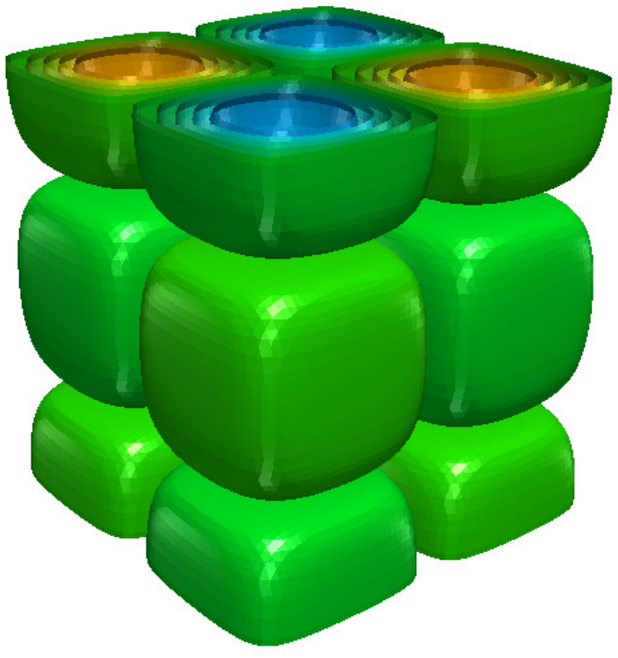


Generate mesh

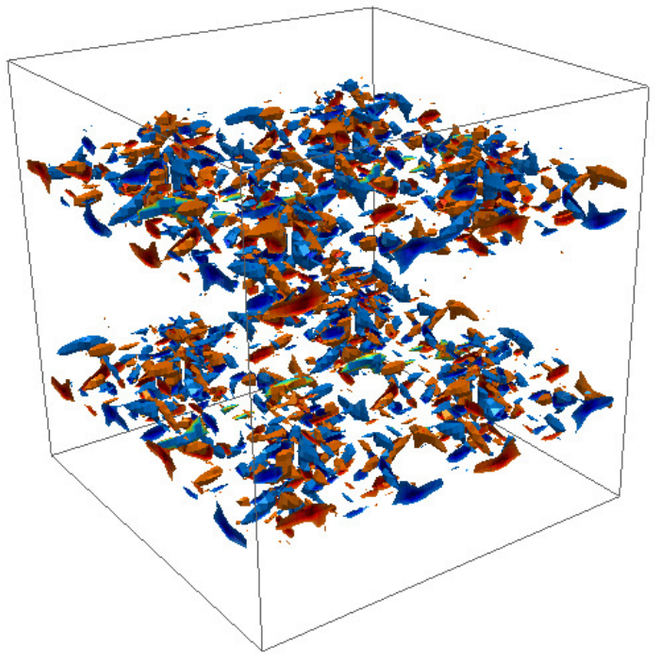
Convert to Nektar++, visualise in Paraview/Visit and configure solver

Run solver (advection) and postprocess

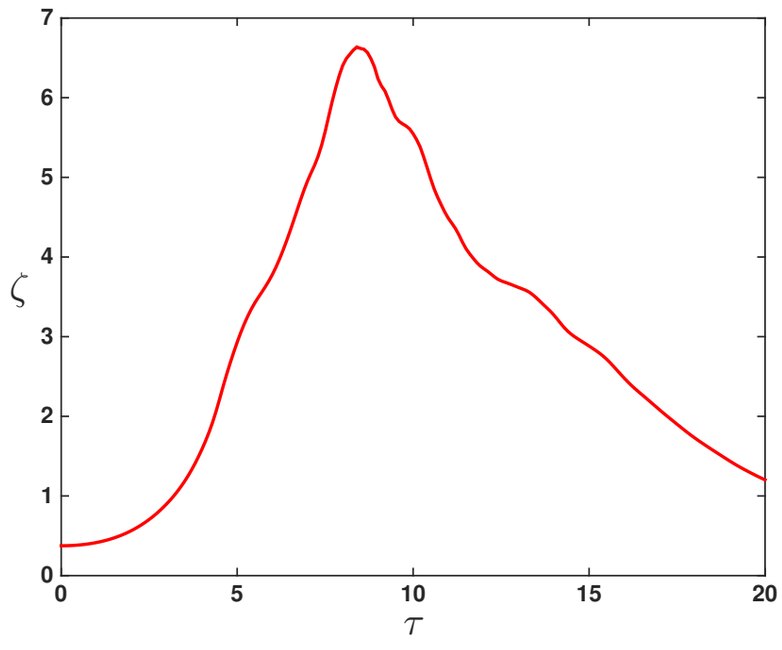
Taylor-Green vortex breakdown



Starting
vortex configuration



Breakdown to
turbulence



Measure enstrophy
and KE production

Summary

- Great activity over the past year
- Focusing on functionality and continue to address fixes
- Focus for coming year will likely be on performance