OpenFoam implementation of two-dimensional isotropic homogenous turbulence decay problem

By: Omar Sallam

March 25, 2024
Source : http://ocean-cfd.engr.tamu.edu/doc/turbdecay

Attachment : 2D_Decay.zip

Report No.: **OCEN CFD Technical Report #240001**

1 Introduction and problem description

The 2-D isotropic homogenous turbulence decay problem is a classical and benchmark Direct Numerical Simulation (DNS)problem. The geometry is a simple square domain with a dimension of $[2\pi \times 2\pi]$ and all four walls have periodic boundary conditions. The energy spectrum $E(\kappa)$ for this problem follows the enstrophy cascade theory (Kraichnan and Montgomery, 1980) or the local theory of 2-D turbulence (Saffman, 1971) where the scaling power ranges between $[\kappa^{-3}, \kappa^{-4.2}]$ in the inertial range (Kida, 1985), κ is the wave number.

This article presents the Direct Numerical Simulation (DNS) implementation of the 2-D incompressible isotropic homogenous turbulence decay problem using OpenFoam (Jasak et al., 2007). The article includes some OpenFoam code snippets for demonstration. In addition, all OpenFoam files required to reproduce the results are attached. OpenFoam-10 is used for this simulation and can be downloaded from `https://OpenFoam.org/release/10/`.

2 OpenFoam solver

pimpleFoam (Holzmann, 2016), an OpenFoam incompressible flow solver, is used to numerically solve the continuity Eq. (1) and the momentum Eq. (2).

$$\frac{\partial u_i}{\partial x_i} = 0 \tag{1}$$

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \tag{2}$$

A uniform square grid is generated for the $[2\pi \times 2\pi]$ square geometry with $[1024 \times 1024]$ cells using the blockMesh utility, where all boundaries have cyclic boundary conditions. The blockMesh file is located in the system folder.

The numerical schemes are assigned in the fvSchemes file in the system folder. Backward Euler is used to discretize the temporal derivative term, 3^{rd} order Gauss scheme is used for the gradient, divergence, and Laplacian operators. The faces' boundary center variables are computed using 3^{rd} order interpolation.

The linear algebraic solvers for the pressure/velocity fields and the PIMPLE algorithm controls are defined in the system/fvSolution file. The first pressure field predictor is computed using the Generalized Geometric-Algebraic MultiGrid solver (GAMG) solver with GaussSeidel as a smoother. The final corrected pressure field is computed using the Diagonal Incomplete-Cholesky GaussSeidel (DICGaussSeidel) solver.

The PIMPLE solver's outer and inner corrector number of iterations is 2. The nonorthogonality corrector iteration (*nNonOrthogonalCorrectors*) is set to 0 because the mesh is uniform, square, and orthogonal.

The code snippet below is from the parameters files in the system folder, this file is not an OpenFoam standard format but we created it to define all variables in a single place to reduce the file preparation time and errors.

In this file, the simulation variables are defined such as the box geometry, mesh size, kinematic viscosityν, total kinetic energy of the initial field Ea, and the peak energy wave number of the initial field $\kappa 0$. This file is called by other OpenFoam standard files to pull the variables from.

Listing 1: Simulation parameters and variables (Not an OpenFoam standard file). Location: system/parameters

```

x_min   0;
x_max   6.283 ;
y_min   0 ;
y_max   6.283  ;
z_min    -0.05 ;
z_max 0.05;
Grid_x    1024 ;
Grid_y 1024 ;
nu     0.00001;
Ea    180;
k0    12 ;
```

3 Boundary conditions

The boundary conditions for the velocity and pressure fields are shown in listings 2 and 3. The initial conditions for both velocity and pressure are set to zero for the entire domain (internalField), these initial conditions will be overwritten later using the boxTurb utility. All sides of the square domain have cyclic (periodic) boundary conditions, since it is a 2-D problem, the frontAndBack faces boundary condition are set to empty. The velocity field file is named U.orig, orig means original, after using the boxTurb utility to generate the initial velocity field, a new velocity file (U) will be created.

Listing 2: Boundary conditions for the velocity field. Location: 0/U.orig

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  10
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    format      ascii;
    class       volVectorField;
    location    "0";
    object      U.orig;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 1 -1 0 0 0 0];
internalField   uniform (0 0 0);
boundaryField
{
    top
    {
        type            cyclic;
    }
    bottom
    {
        type            cyclic;
    }
    right
    {
        type            cyclic;
    }
    left
    {
        type            cyclic;
    }
    frontAndBack
    {
        type            empty;
    }

}
```

Listing 3: Boundary conditions for the pressure field. Location: 0/p

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFOAM: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://openfoam.org
    \\  /    A nd           | Version:  10
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    format      ascii;
    class       volScalarField;
    location    "1";
    object      p;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
dimensions      [0 2 -2 0 0 0 0];
internalField   uniform 0;
boundaryField
{
    top
    {
        type            cyclic;
    }
    bottom
    {
        type            cyclic;
    }
    left
    {
        type            cyclic;
    }
    right
    {
        type            cyclic;
    }
 frontAndBack
    {
        type            empty;
    }
}
// ************************************************************************* //
```

4 Initial conditions

In the 2-D turbulence decay problem, the initial random velocity field is generated such that it satisfies the divergence-free constraint, $\nabla \cdot \mathbf{u} = 0$. In OpenFoam, the boxTurb generates this random divergence-free velocity field. The generated velocity field follows the spectrum shown in Eq. (3). In equation Eq. (3), κ is the wave number, κ_p is the peak energy spectrum wave number, and Ea is a constant to scale the total energy contained in the domain.

Ea and κ_p values are assigned in the turbBoxDic file located in the constant directory as shown in listing 4. In the BoxTurbDict file, the system/parameters file is included to pull the Ea and the k0 values.

The turbBox utility originally creates 3-D velocity fields. For 2-D problems, it creates 3-D velocity fields, but after the first time step, the 3^{rd} velocity component vanishes.

$$E(\kappa) = E_a \left(\frac{\kappa}{\kappa_p}\right)^4 e^{-2\left(\frac{\kappa}{\kappa_p}\right)^2} \tag{3}$$

Listing 4: BoxTurbDict for the velocity initial field generation. Location: constant/BoxTurbDict

```
/*--------------------------------*- C++ -*----------------------------------*\
  =========                 |
  \\      /  F ield         | OpenFoam: The Open Source CFD Toolbox
   \\    /   O peration     | Website:  https://OpenFoam.org
    \\  /    A nd           | Version:  10
     \\/     M anipulation  |
\*---------------------------------------------------------------------------*/
FoamFile
{
    format      ascii;
    class       dictionary;
    location    "constant";
    object      boxTurbDict;
}
// * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * //
#include "../system/parameters"
Ea              $Ea;
k0              $k0;
```

5 Simulation control and Allrun file

Other simulation controls are defined in the system/controlDict file, such as the simulation time = 50, initial time step = 0.0001, maximum Courant number =0.7, and the solution writing intervals = 0.2.

The Allrun bash script shown in listing 5 is used to run all the OpenFoam commands automatically. First, it calls the application name from the controlDict file using the $(getApplication) command. then, the grid is generated using blockMesh, then the initial velocity field is generated using boxTurb command. The problem is solved in parallel, hence the decomposePar command is used to distribute the mesh cells to the different processor threads. The number of processors can be controlled from the system/decomposeParDict file. Finally, the run-Parallel $application command is used to start the pimpleFoam solver in parallel. The last two commands are for postprocessing, such as combining the solutions from the processor threads and computing the enstrophy field, respectively.

Listing 5: Allrun file used to run all commands automatically.

```
#!/bin/sh
cd ${0%/*} || exit 1    # Run from this directory

# Source tutorial run functions
. $WM_PROJECT_DIR/bin/tools/RunFunctions

# Get application name
application=$(getApplication)
runApplication blockMesh
runApplication boxTurb
runApplication decomposePar -cellDist
runParallel $application
runApplication reconstructPar
runApplication -s enstrophy  postProcess -func enstrophy
```

6 Results

Fig. 1 shows the evolution of the vorticity contours for the decay turbulence. At the first time step, the vorticity contours are for the initial random divergence-free field with wave number $\kappa_p = 12$. With time evolution, coherent large-scale vortices and small-scale vortices form. The size of the large-scale vortices increases with time, and the small-scale vortices dissipate due to the viscous dissipation. This can also be observed from the energy spectrum shown in Fig. 2, at later time steps the inertial range of the spectrum shift to the left (lower wave number or larger vortex size), and the energy spectrum magnitude decrease in the viscous dissipation range at high wave number values. The energy spectrum slope follows $E(\kappa) \propto \kappa^{-4}$ aligning with the local theory of 2-D turbulence (Saffman, 1971), (Kida, 1985).

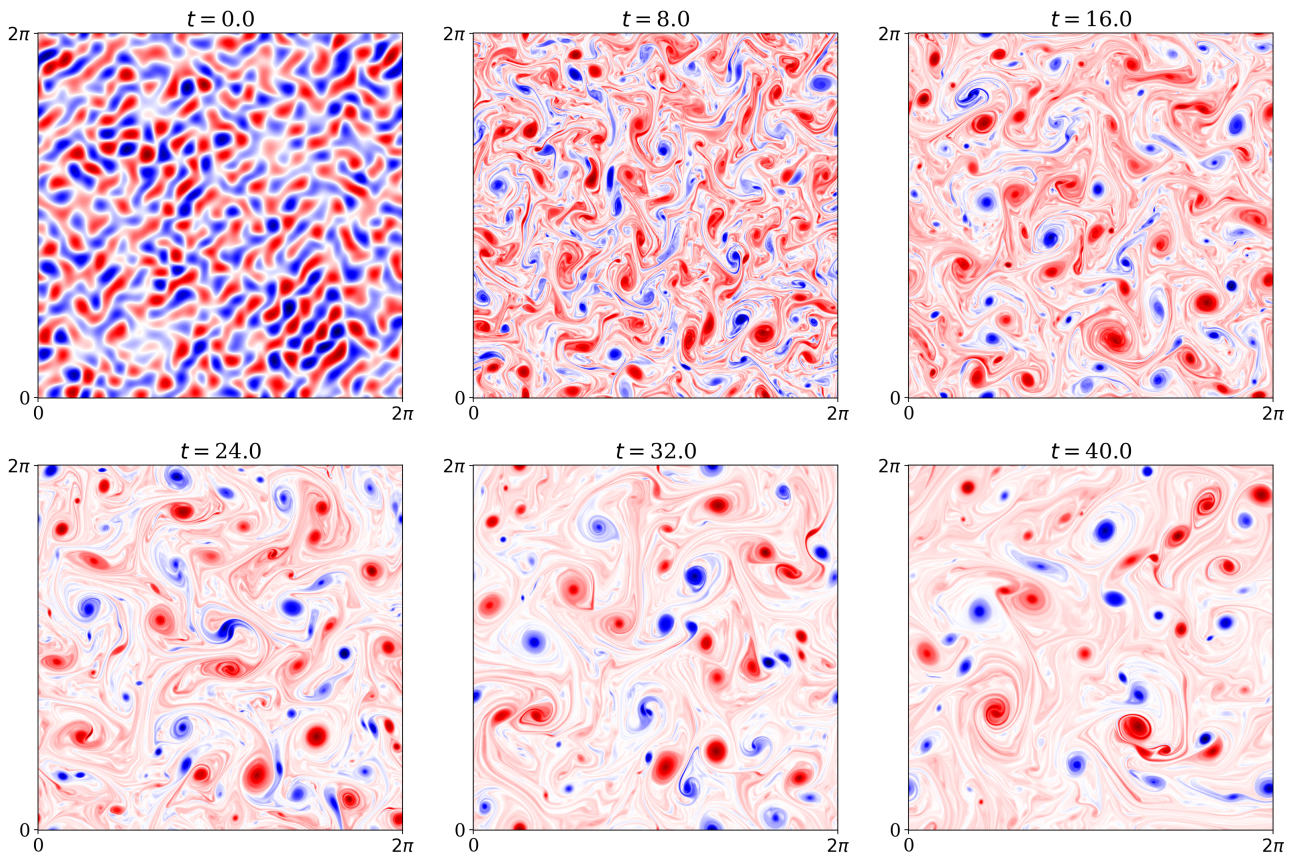

Figure 1: Vorticity contours evolution for the 2-D isotropic homogenous decay turbulence problem with random initial divergence-free field.

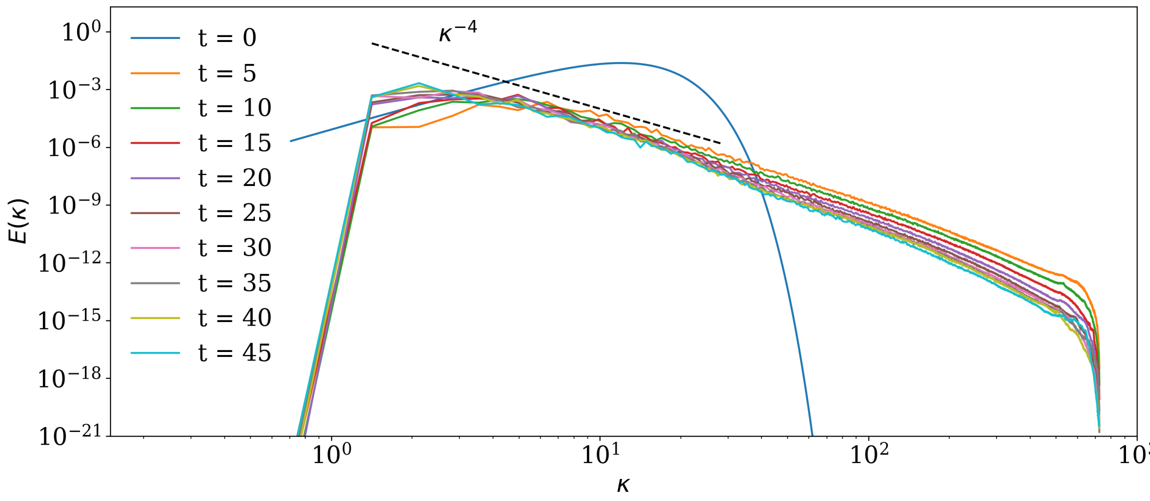

Figure 2: Energy spectrum evolution for the 2-D isotropic homogenous decay turbulence problem with random initial divergence-free field.

References

Holzmann, T. (2016). Mathematics, numerics, derivations and openfoam®. *Loeben, Germany: Holzmann CFD.*

Jasak, H., Jemcov, A., Tukovic, Z., et al. (2007). Openfoam: A c++ library for complex physics simulations. In *International workshop on coupled methods in numerical dynamics*, volume 1000, pages 1–20.

Kida, S. (1985). Numerical simulation of two-dimensional turbulence with high-symmetry. *Journal of the Physical Society of Japan*, 54(8):2840–2854.

Kraichnan, R. H. and Montgomery, D. (1980). Two-dimensional turbulence. *Reports on Progress in Physics*, 43(5):547.

Saffman, P. (1971). On the spectrum and decay of random two-dimensional vorticity distributions at large reynolds number. *Studies in Applied Mathematics*, 50(4):377–383.