3D Taylor-Green Vortex Comparison

Paul Bartholomew

March 1, 2019

Abstract

As part of the verification of `Xcompact3D` we simulate the Taylor-Green vortex and compare with results from a reference 6^{th} order compact finite difference code provided by Eric Lamballais.

Contents

1 Introduction

As a canoncical test case, the Taylor-Green vortex provides a check that the time integration of the Navier-Stokes is working correctly. The Taylor-Green vortex is initialised as follows

$$\boldsymbol{u} = \begin{cases} U \sin\left({}^{x}/_{\pi}\right) \cos\left({}^{y}/_{\pi}\right) \cos\left({}^{z}/_{\pi}\right) \\ -U \cos\left({}^{x}/_{\pi}\right) \sin\left({}^{y}/_{\pi}\right) \cos\left({}^{z}/_{\pi}\right) \\ 0 \end{cases} \tag{1}$$

in the $-\pi \leq \boldsymbol{x} \leq \pi$ periodic box. To save computational effort, the symmetries inherent in the flow field are exploited to simulate only the impermeable sub-domain $0 \leq \boldsymbol{x} \leq \pi$.

2 Computational setup

2.1 Discretisation

To ensure comparability of the results we must first ensure the same schemes are being used - the pertinent variables are `fpi2` and `ailcaix6` (and `y` and `z`) which should be set to $48/7/\pi^2$ and 0.461658 in both codes.

2.2 Runtime parameters

The runtime parameters pertinent to the simulation are given in table 1.

Table 1: Runtime parameters for Taylor-Green Vortex simulations.

Parameter	**Value**	**Notes**
`xlx`	3.14159265358979	`yly` and `zlz` the same
`nx`	65	`ny` and `nz` the same equivalent to a 129^3 domain
`nclx`	1	`ncly` and `nclz` the same corresponds to free-slip
`dt`	0.001	
Time scheme	`RK3`	
`ilast`	20,000	
Output frequency	1,000	Stores snapshots every 1,000 steps

3 Comparison of results

The main statistics of interest for comparison purposes are the kinetic energy and enstrophy, defined as

$$k = \frac{1}{2}\int_{\Omega} \boldsymbol{u}^2 dV \ , \tag{2}$$

and

$$\varepsilon = \int_{\Omega} |\boldsymbol{\omega}|^2 dV \ , \tag{3}$$

where

$$\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u} \ , \tag{4}$$

is the vorticity.

The codes compute these statistics online, here a `python` script has been developed to plot them for comparison. It expects that the data are located in `./x3d/time_evol.dat` and `./e3d/time_evol.dat` with the following format (comment lines beginning with a `#` and and additional columns are ignored)

#	**TIME**	**KE**	**DISS**	**ENST**	**DISS2**
	t1	k1	d1	e1	D1
	t2	k2	d2	e2	D2
	...				
	tn	kn	dn	en	Dn

which can be read by following `python` snippet given in listing 1.

```python
def read_stats(filename):
  t = []
  enst = []
  ke = []

  with open(filename, "r") as data:
    for row in data:
      if not (row[0]=="#"):
        words = row.split()
        t.append(float(words[0]))
        enst.append(float(words[3]))
        ke.append(float(words[1]))

  return t, enst, ke
```

Listing 1: Python code to read statistics for TGV case.

The data are plotted using `matplotlib` in listing 2.

```python
def plot_stats(x3d_t, x3d_dat, x3d_lab, e3d_t, e3d_dat, e3d_lab,
               xlab, ylab, outfile, figsize=(5.0, 3.5)):

  plt.figure(figsize=figsize)

  plt.plot(x3d_t, x3d_dat, label=x3d_lab)
  plt.plot(e3d_t, e3d_dat, label=e3d_lab)

  plt.xlabel(xlab)
  plt.ylabel(ylab)
  plt.legend(prop={"family":"serif",
                   "size":11})

  plt.savefig(outfile, bbox_inches="tight")
  plt.close()
```

Listing 2: Python code to plot comparison of `Xcompact3D` and Eric's reference code.

And finally, the following script (`plot_tgv.py`) plots the data in *fig.* 1 and *fig.* 2.

```python
import matplotlib.pyplot as plt
plt.rc("text", usetex=True)
plt.rc("font", family="serif")
plt.rc("font", size=11)

<<src:read-stats.py>>
<<src:plot-stats.py>>

x3d_t, x3d_enst, x3d_ke = read_stats("./x3d/time_evol.dat")
x3dgpu_t, x3dgpu_enst, x3dgpu_ke = read_stats("./x3d-gpu/time_evol.dat")
e3d_t, e3d_enst, e3d_ke = read_stats("./e3d/time_evol.dat")

plt.figure(figsize=(5.0, 3.5))
plt.plot(x3d_t, x3d_enst, label="X3D")
plt.plot(x3dgpu_t, x3dgpu_enst, label="X3D-GPU")
plt.plot(e3d_t, e3d_enst, label="Eric")
plt.xlabel(r"$t$")
plt.ylabel(r"$\varepsilon$")
```

```
plt.legend(prop={"family":"serif",
                 "size":11})
plt.savefig("tgv_enstrophy.eps", bbox_inches="tight")
plt.close()

plt.figure(figsize=(5.0, 3.5))
plt.plot(x3d_t, x3d_ke, label="X3D")
plt.plot(x3dgpu_t, x3dgpu_ke, label="X3D-GPU")
plt.plot(e3d_t, e3d_ke, label="Eric")
plt.xlabel(r"$t$")
plt.ylabel(r"$k$")
plt.legend(prop={"family":"serif",
                 "size":11})
plt.savefig("tgv_ke.eps", bbox_inches="tight")
plt.close()
```

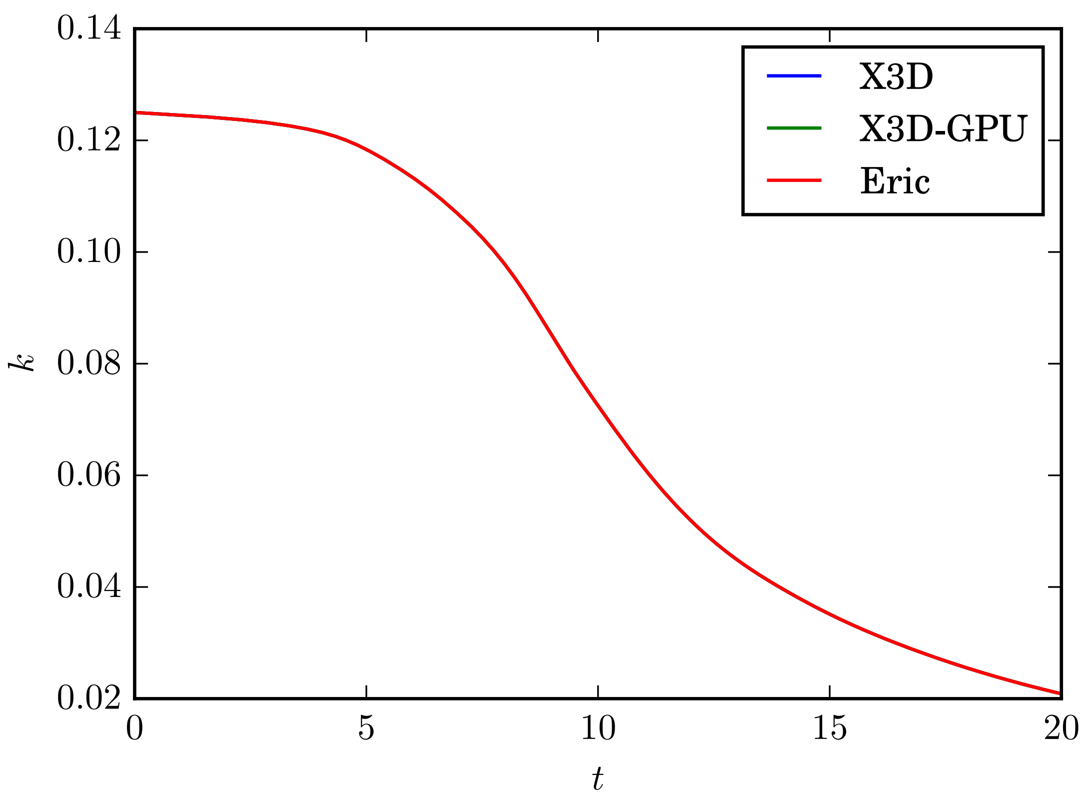

Figure 1: Comparison of kinetic energy

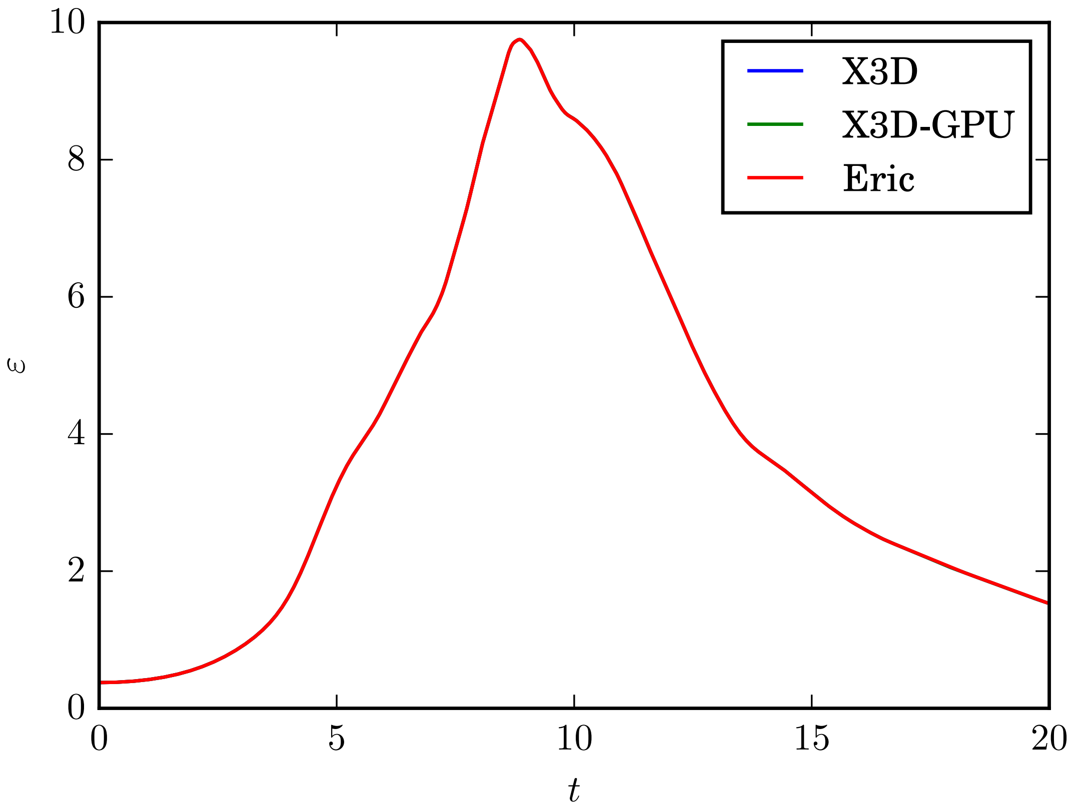

Figure 2: Comparison of enstrophy