Luis,
Solve the following problem using les3d-mp:

$$\frac{du}{dt} = -\sin\frac{2\pi t}{T}; \quad u(0) = 0. \tag{1}$$

The analytical solution is

$$u(t) = \frac{T}{2\pi}\left(1 - \cos\frac{2\pi t}{T}\right) \tag{2}$$

Since the solution is not $f(x)$, this remove spatial errors from the analysis. You can do this by running a 4^3 grid, with periodic conditions in x and z and symmetry conditions on top and bottom. The config file is on page 4. I use the following law in `adjust_mean_dpdx.f90`:

```
#ifdef OSC_PRESSURE_GRAD
  mean_dpdx = pg_mean - pg_amp*SIN(2*pi*elapsed_time/pg_period)
#endif
```

Then I run using the `les3d-mp.par` file also attached, with $\Delta t = 0.1$, 0.01, 0.001 and 0.0001, for a total time $3T/2$, at which point

$$u(t) = \frac{T}{2\pi}\left(1 - \cos\frac{2\pi 3T}{2T}\right) = \frac{T}{\pi} \tag{3}$$

For the given data ($T = 10$), this gives $u = 10/\pi$. Check that you get this

```
h5dump -d u Restart/restart.h5
HDF5 "Restart/restart.h5" {
DATASET "u" {
   DATATYPE  H5T_IEEE_F64BE
   DATASPACE  SIMPLE { ( 6, 6, 6 ) / ( 6, 6, 6 ) }
   DATA {
   (0,0,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (0,1,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (0,2,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (0,3,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (0,4,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (0,5,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (1,0,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (1,1,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (1,2,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (1,3,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (1,4,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (1,5,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (2,0,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (2,1,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (2,2,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
```

```
   (2,3,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (2,4,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (2,5,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (3,0,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (3,1,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (3,2,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (3,3,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (3,4,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (3,5,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (4,0,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (4,1,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (4,2,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (4,3,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (4,4,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (4,5,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (5,0,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (5,1,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (5,2,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (5,3,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (5,4,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309,
   (5,5,0): 3.18309, 3.18309, 3.18309, 3.18309, 3.18309, 3.18309
   }
}
}
```

If at this point you do

```
hpc2068@sflogin0$ tail -1 history
  1.500000000E+01  0.000000000E+00 -3.552713679E-15 3.183088390E+00
```

the last number will give you the velocity, and subtracting $10/\pi$ from it you get the error. This was my result

```
0.1 3.182051595
0.01    3.183088390
0.001   3.183098757
0.0001  3.183098861
```

which gives the desired second-order slope (see figure).

I hope that you are convinced of the second-order accuracy of the time advancement, and we can put this issue to rest once and for all, so that you can work on something that will move your research forward.

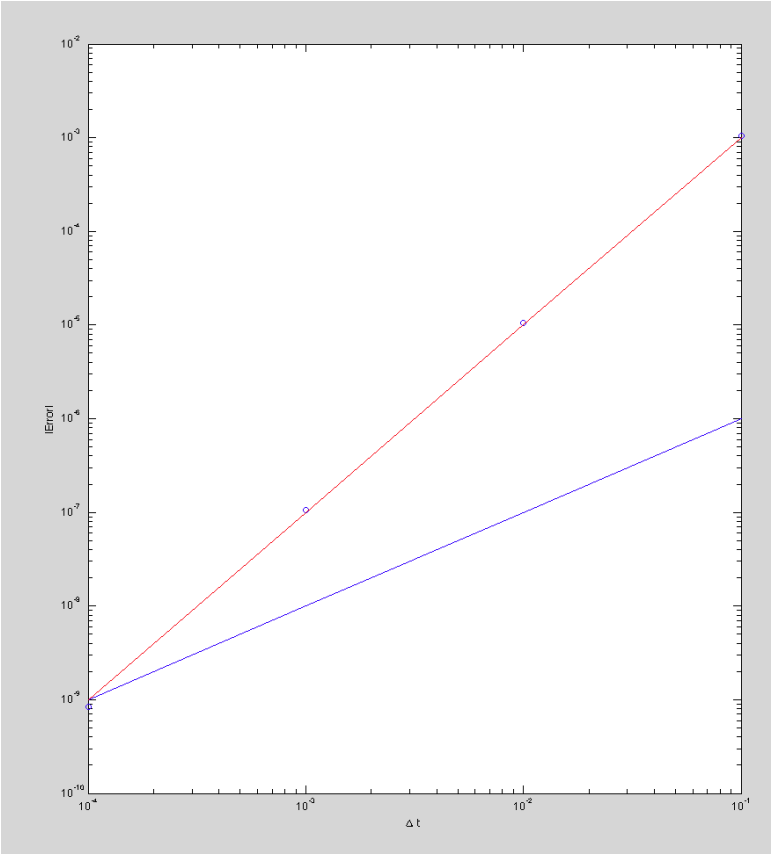

Figure 1: Error

config file

```
# -- Time advancement (USE_RK3 or USE_AB2)
CPPFLAGS USE_RK3

# -- Number of nodes and grid generation equations
OPTION ni 4
OPTION nj 4
OPTION nk 4
OPTION gridx x
OPTION gridy x
OPTION gridz x

# -- Reynolds number
#CPPFLAGS FIXED_PRESSURE_DROP
VARIABLE real Re 1.
#VARIABLE real Re 130000.

# ---- oscillating pressure gradient
CPPFLAGS OSC_PRESSURE_GRAD
VARIABLE real    pg_mean    0.
VARIABLE real    pg_amp     1.
VARIABLE real    pg_period  10.

# -- Initial conditions
INCLUDE initial_conditions/initial_constant.f90
CONSTANT real initial_conditions_mean  0.
CONSTANT real initial_conditions_noise 0.

# -- Wall boundary conditions
INCLUDE boundary_conditions/bottomwall_symmetry.f90
INCLUDE boundary_conditions/topwall_symmetry.f90

# -- Periodic boundary conditions
CPPFLAGS PERIODIC_IN_X
INCLUDE adjust_mean_dpdx.f90
INCLUDE calculate_mass_flux.f90
VARIABLE logical constant_mass_flux true
VARIABLE real    mean_dpdx -1.0
VARIABLE real    mass_flux  1.0

# -- Poisson solver
INCLUDE poisson_solvers/poisson_solver_fft.f90
```

les3d-mp.par file

```
# Time integration parameters
nt = 1500
dt = 0.01
cfl = 10
fixed_cfl = f

# Simulation restart parameters
restart = f
restart_file = Restart/restart.h5

# Checkpoint parameters
checkpoint = false
checkpoint_every = 50
rolling_checkpoints = 2
checkpoint_dir = ./Restart/

# Snapshot parameters
snapshot = false
snapshot_every = 5.
first_snapshot_number = 1
snapshot_dir = ./Snap/

# Estimated finish time printout every
eta_every = 200
reset_elapsed_time = t
```