Pseudo-spectral DNS - parallel code user manual

Parallel code

It is possible to use this parallel code on every computer or cluster (i.e. HPC or multicore processor pc) on which are installed MPI parallel libraries. The code outputs every saved instant, three files u1.xy.bin, u2.xy.bin, u3.xy.bin in single precision format. Moreover it is not necessary compile again codes when resolution is changed: number of points is read when codes are running and variables are dinamically allocated depending on resolution (FORTRAN 90).

1 Code compiling

Compiler of the serial code is gfortran, while compiler of the parallel code is mpi90 (fortran+librerie mpi).
All the instructions are written in the Makefile. The source code are:

Running of the simulations
- dns14_mpi.x → Navier-Stokes equations solution;
- dns14_scal.x → Navier-Stokes equations and one (or more) passive scalars solution.

Pre-processing e e post-processing
- dns14_mpi.x → computes 3D spectrum, integral scale, Taylor scale, energy and dissipation for isotropic turbulence;
- dns14_ske.x → compute second moment, skewness and kurtosis of the three components of velocity (average on planes with \( x_3 = \text{cost} \));
- dns14_ske_scal.x → Computes second moment, skewness and kurtosis of the passive scalar (average on planes with \( x_3 = \text{cost} \));
- dns14_genera.x → outputs initial data of isotropic turbulence (using a generator of random number);
- dns_genera.x→ similar to dns14_genera.x, ma non è un codice parallelo.
2 Codes running

In the same directory:

1. Main code (file .x);
2. files has parameters and/or initials data needed by main code);
3. file with “script” to run the main parallel code.

\textbf{dns14\_mpi.x}

It solves Navier-Stokes equations (pseudospectral+Runge-Kutta 4 method) and main source is the file \texttt{dns14\_conf.f90}.

Input file:

- \texttt{param\_ns.txt} → contains following parameters:
  
  \begin{itemize}
  \item \( N \) points number in directions 1, 2;
  \item \( N3 \) points number in directions 3;
  \item \( \Delta t \) temporal step;
  \item \( Ntot \) total steps number;
  \item \( N\text{salva} \) steps number saving;
  \item \( Re \) Reynolds number;
  \item \texttt{nomefile} file name with three velocity components (each for every component) to use as initial conditions.
  \end{itemize}

- file containing initial consitions.

Outputs file:

- \texttt{avvio.txt} → parameters file;
- the three components of velocity for each saved instant \texttt{u1.xy.bin, u2.xy.bin, u3.xy.bin} →.

\textbf{Notes}

1. The domain is a parallelepiped with dimensions \( 2\pi \times 2\pi \times 2\pi \frac{N3}{N} \) (fig.1).

2. It is not necessary recompile code if data dimensions are changed. It is only necessary update file \texttt{param\_ns.txt}.

3. It is not limited to a cube.
Figura 1: Computational domain.

**dns14_scal.x**

It solves Navier-Stokes equations as *dns14_mpi.x*; Moreover it solves passive scalar transport equations:

\[ \partial_t \vartheta + \nabla \cdot (u \vartheta) = \frac{1}{Sc \cdot Re} \nabla^2 \vartheta \]

in adimensional form, with \( \vartheta \) is the passively transported quantity by the fluid, with Re Reynolds number, Sc Schmidt number, as:

\[ Re = \frac{UL}{\nu} \]

\[ Sc = \frac{\nu}{D} \]

with \( U \) fluid velocity in one direction, \( L \) characteristic dimension, \( \nu \) cinematic viscosity and \( D \) transported scalar diffusivity.

In adimensional form:

\[ \partial_t \vartheta + \nabla \cdot (u \vartheta) = D \nabla^2 \vartheta. \]

Input files:

- `param_ns_scal.txt` contains the following parameters:
  - \( N \) points number in directions 1, 2;
  - \( N3 \) points number in directions 3;
  - \( \Delta t \) temporal step;

---

1 Vectors are in bold.
\( N_{tot} \) total steps number;
\( N_{salva} \) steps number saving;
\( Re \) Reynolds number;
\( nomefile \) file name with three velocity components (each for every component) to use as initial conditions.

\( N \) points number in directions 1, 2;
\( N3 \) points number in directions 3;
\( \Delta t \) temporal step;
\( N_{tot} \) total steps number;
\( N_{salva} \) steps number saving;
\( Re \) Reynolds number;
\( nomefile \) file name with three velocity components (each for every component) to use as initial conditions.

\( N_{scal} \) number of scalars;
\( SC1 \) Schmidt number of the first scalar;
\( SCn \) Schmidt number of the \( n \)-esimo scalar

- \( nomefile(1), nomefile(2), nomefile(3) \rightarrow \) components velocity file.

If \( Izero = 0 \) initial conditions for the passive scalar are whole generated\(^2\):

\[
\begin{align*}
\vartheta_1 (x_1, x_2, x_3) &= \begin{cases} 
1 & \text{se } x_3 < L/2 \\
0 & \text{se } x_3 > L/2 
\end{cases} \\
\vartheta_2 (x_1, x_2, x_3) &= \begin{cases} 
0 & \text{se } x_3 < L/2 \\
1 & \text{se } x_3 > L/2 
\end{cases} \\
\vartheta_3 (x_1, x_2, x_3) &= 1 \\
\forall (x_1, x_2, x_3) \\
\vartheta_k (x_1, x_2, x_3) &= \vartheta_1 (x_1, x_2, x_3) \quad \forall k \geq 4
\end{align*}
\]

If \( Izero > 0 \) initial conditions are read by files \( scalk-xy.old \), where \( k \) is the reference number of the scalar and \( xy \) value \( Izero \) with two digit. Outputs files:

- \( u1.xy.bin, u2.xy.bin, u3.xy.bin; \)
- \( scalk-xy.bin \rightarrow \) scalar saved for each instant \( xy \).

\(^2\)Con \( L \) it is defined the dimension of the domain in the direction 3.
dns_genera.x

Outputs data in pseudo-random way in a cube with \( N^3 \) points. The velocity field is:

\[ u = \nabla \wedge A \]

with \( A \) calculated by means generation of the random number.

So, \( \nabla \cdot u = 0. \)

We can write:

- \( \text{Re} \left[ \hat{A}_i(k) \right] = \alpha(k) \cos(2\pi\xi) \)
- \( \text{Im} \left[ \hat{A}_i(k) \right] = \alpha(k) \sin(2\pi\xi) \)

where:
- \( \hat{A}_i \) is the Fourier transform of \( A_i \);
- \( \alpha(k) \) is the magnitude calculated by means the energy spectrum as:

\[ \alpha(k) = \sqrt{\frac{E(k)}{k}} \]

\( \xi \) is the random number computed by an uniform distribution with an uniform probability density densità in \([0, 1]\) range.

Spectrum is chosen as fig.2.

\[
\begin{align*}
E(k) & \sim k^S & \text{per} & k < k_0 \\
E(k) & \sim k^{-5/3} & \text{per} & k < k_{max}
\end{align*}
\]

\[ k_0 \] value is chosen by user, while \( k_{max} \) can be chosen as \( N/2 \) or \( N/3 \).

This scalar code can be executed with ./dns_genera.x.

The parameters must be input by clipboard. Outputs files are: u1.99.bin, u2.99.bin, u3.99.bin.
**dns14_genera.x**

Similar to dns_genera.x but is a parallel code. Parameters are written in file param_genera.txt, that contains:

- $N$ number of points;
- $k_0$ energy spectrum maximum;
- $S$ low wavenumber spectrum exponent;
- $j_{tutti}$ $1 \rightarrow k_{max} = N/2$ $0 \rightarrow k_{max} = N/3$;
- $u'_{rms}$ root mean square of velocity fluctuations ($E = \frac{3}{2} u'^2$);
- $nomefile$ files name with three velocity components of the generate field;
- $1$ not used parameter.

**dns14_spettro.x**

Executable file for computing of spectrum and scales.

Input files:

- param_ns.txt;
- uxy.bin → simulation data, con xy temporal instant;
- in.skder→ parameters:
  - $I_{max}$ index of the first computing file;
  - $I_{max}$ index of the last computing file;
  - $I_{p}$ step between files (1→ all the files $n$→each file every $n$).

Output files:

- spettro_xy.txt → spectrum;
- decadimento.txt → contains energy, dissipation and scales in function of the time.
**dns_ske_u.x**

It computes energy, skewness, kurtosis for three velocity components. It executes average in planes \((x_1, x_2)\), as showed in fig.3.

![Diagram](image)

Figura 3: Calculation of the average property on a plane in \(x_3 = \text{const.}\)

\[
\bar{f}(x_3) = \frac{1}{L^2} \int_0^L \int_0^L f(x_1, x_2, x_3) \, dx_1 dx_2
\]

Inputs file
- `param_ns.txt`;
- `uxy.bin` → simulation data, with \(xy\) temporal instant;
- `in.skder` → parameters:
  - \(I_{\text{min}}\) index of the first file to compute;
  - \(I_{\text{max}}\) index of the last file to compute;
  - \(I_{p}\) step between of files (1→all the files \(n\)→each file every \(n\)).

Outputs files:
- `ske_1_xy.txt`, `ske_2_xy.txt`, `ske_3_xy.txt` → files with four columns (in order \(x_3\), second moment \(u'^2\), skewness \(S\) and kurtosis \(K\)); It is written the scalar component of \(u\) \((1, 2 \text{ o } 3)\).

**Nota** It can be \(N \neq N_3\).
They compute passive scalar statistics. Output files give the following values:

- \( x_3, \bar{\theta}(x_3), \overline{\theta^2}(x_3), S_{\theta'}, K_{\theta'} \);
- \( x_3, \bar{u\theta}(x_3), \overline{u^2\theta}(x_3) \).

The average is calculated on planes at \( x_3 = \text{cost} \) as \text{dns14_ske.x}.

### 3 Visualizations

To visualize data we use the software VisIt. It needs convert files into the format .vtk (visual toolkit):

1. From initial file .bin to file .bin with each antitransformed variable in space for every file.
   grid.out is generated with coordinates.

2. Passaggio dal file .bin al file .vtk tramite the source code bin2vtk. Convert files .bin into .vtk

To execute the code it is used the following syntax:

\[
\text{bin2vtk nome file -float -vname NOME}
\]