Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona,Spain, July 9-13, 2018

ICCFD10-2018-373

Developing 2-D Stretching in a High Order DNS Code: Application to Turbulent Flow in a Square Duct

C. Moulinec*, Sylvain Laizet** and David R. Emerson*
Corresponding author: charles.moulinec@stfc.ac.uk

* STFC Daresbury Laboratory, Warrington, UK.
** Imperial College London, London, UK.

Abstract: A Direct Numerical Simulation open-source software supporting high order finite difference schemes, with 1 direction of stretching in space is modified to support stretching in 2 directions in space. The path to change the code is explained for general turbulent flows and some preliminarly results are presented.

Keywords: High Order Schemes, Finite Difference, Stretching, DNS.

1 Introduction

Square and rectangular ducts are very much used in industry, as for heating, ventilation and air conditioning (HVAC) and, nuclear engineering for instance. With the raise in computing power, it is now possible to envisage to run nearly industrial Direct Numerical Simulations (DNSs) using high order finite difference software. The objective of this work is to explain how to retain 6^{th} order accuracy in space for flows better represented by stretching in 2 directions. The original code, namely Incompact3d [1] [2] is first presented, followed by the modifications to allow for stretching in 2 directions. The flow in a square duct (see Fig. 1) is used as a test case to demonstrate the new capability. The performance of the code is presented before drawing some final remarks.

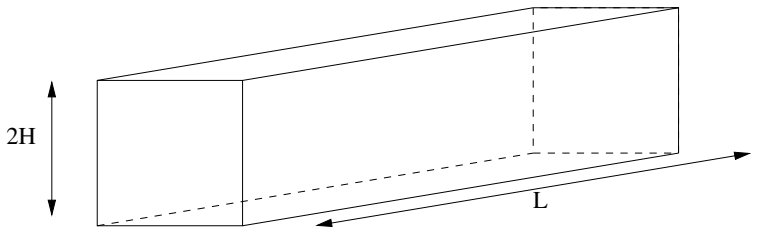

Figure 1: Sketch of the square duct.

2 1-D stretching

2.1 Brief introduction to the code

Incompact3d is an open-source high order flow solver that can undertake turbulence-resolving simulations of fluid flow phenonema on Tier–0 systems [2]. It solves the incompressible Navier–Stokes equations on a Cartesian mesh using up to 6^{th} order schemes for spatial discretisation and a conventional 4^{th} order Adams–Bashforth scheme for time advancement. To ensure incompressibility, a fractional step method is used. It requires to solve a Poisson equation, which is fully performed in the spectral space.

Combined with the concept of the modified wave number [3], a direct (i.e. non-iterative) technique allows the implementation of the divergence-free condition up to machine accuracy. A partially staggered mesh [4] is used where the pressure grid is shifted from the velocity grid in each direction. This type of grid organisation leads to physically realistic pressure fields without unwanted spurious oscillations.
As the pressure is computed in the spectral space, stretching there is built from an approximation of a double convolution product between 2-D homogeneous operators (considering that the 1-D grids are regular) and the operators used to account for stretching. The stretching is defined by an analytical function. To account for refinement at walls in the case of a channel for instance, the derivative h' of this function reads [1] [4]:

$$\frac{1}{h'(s)} = \frac{2}{H}\left((\frac{\alpha}{\pi} + \frac{1}{2\pi\beta}) - \frac{1}{4\pi\beta}(e^{i2\pi(\gamma s+\delta)} + e^{-i2\pi(\gamma s+\delta)})\right)$$

where α, β and γ are coefficients to be tuned to control y^+, also accounting for the number of nodes in a given direction.

2.2 Parallelism - Performance at scale

For the conventional version of the code, a high level of parallelism is achieved up to over 1 million MPI–processes thanks to a highly scalable 2-D domain decomposition library and a distributed Fast Fourier Transform (FFT) interface [2]. The computational domain (X, Y, Z) is divided in pencils in each direction, X being usually the streamwise direction. The pressure Poisson equation is solved in the spectral space, and for a straight channel flow, 1-D FFTs are computed in each homogeneous direction, whereas 5-band matrix systems are solved in the stretched direction, by means of an internal direct solver.
Table 1 shows the performance of the 1-D stretched version for a 17 B node mesh. A very good speedup of 2.91 is observed going from 262,144 to 1,048,576 MPI tasks of an IBM Blue Gene/Q (MIRA, Argonne Laboratory, US).

MPI Tasks	Time per time-step	Speedup	Ideal speedup
262,144	5.39 s	1.00	1.00
524,288	2.88 s	1.87	2.00
1,048,576	1.85 s	2.91	4.00

Table 1: Performance at scale. Simulation for a 17 B node mesh.

3 Modifications to allow for 2-D stretching

Allowing for stretching in 2 directions prevents from using FFTs in these 2 directions, when solving the pressure in the spectral space. The pencil technique (see Fig. 2 (left)) described above implies that extra MPI communications are required to solve the pressure in the 2 stretched directions, whereas a slab-based approach (see Fig. 2 (right)) would not. The current code (supporting 1-D stretching) is designed with the assumption that the homogeneous streamwise direction is X. Unfortunately, the 2-D domain decomposition in pencils (*rows*, *cols*) does not support 'serial' slabs, e.g. slabs with no MPI communication inside them. However, if the homogeneous streamwise direction is changed to Z, and the number of *rows* set to 1, the code does support 'serial' slabs, where stretching in X and Y is possible without communication within the slabs.
The pressure being solved in the spectral space, 1-D FFTs are used in the streamwise direction and a series of 2-D systems built from an approximation of a double convolution product between 2-D homogeneous operators (considering that the 2-D grids are regular) and the operators used to account for stretching in the 2 other directions. This leads to a series of 9-band matrices which are solved using a direct solver.

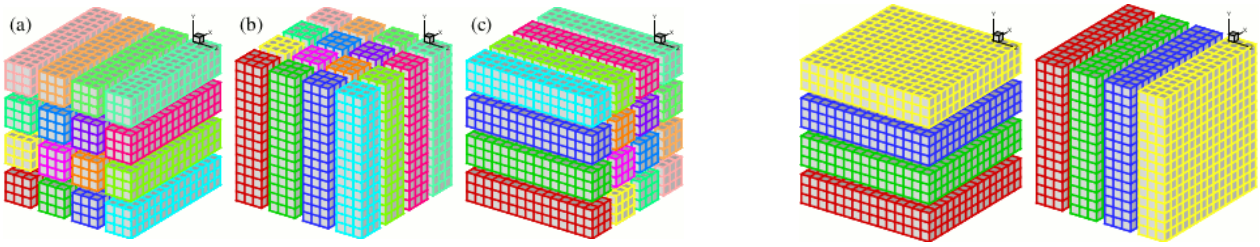

Figure 2: Left: Pencil decomposition. Right: Slab decomposition.

4 Simulations in a square duct

4.1 Laminar FLow

The first test case is run laminar at Re=100, where the Reynolds number is based on the bulk streamwise velocity and the duct height ($2H$). The stretching is applied in both X and Y directions and Fig. 3 shows the stretched grid. The isocontours of the streamwise velocity are also plotted.

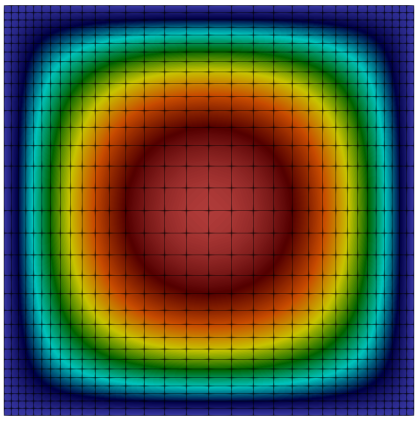

Figure 3: Laminar flow - Cross section.

4.2 Turbulent flow - Comparison for X-stretching and Y-stretching

A DNS is conducted in a square duct at Re=$4,400$ based on bulk streamwise velocity and duct height. Two cases are considered to show that stretching works in X or Y direction, with stretching in X (resp. Y) and an homogeneous distribution of nodes in Y (resp. X). Figures 4 show a snapshot of the instantaneous fully developed flow, with the isocontours of the streamwise velocity. The grids are added to both figures to show the stretching in the X direction (left) and in the Y direction (right).
Figures 5 left and right present the mean streamwise (Z) component of the velocity. On the vertical (resp. horizontal) walls, when the strecthing is applied in X (resp. Y) direction, the boundary layers are better represented than on the walls of the opposite direction. This is also confirmed on Figs. 6 left and right where the secondary vectors are more shifted towards the walls where stretching applies. However, if symmetry along X=Y would be applied to Fig. 5 (resp. 6) left, it would match Fig. 5 (resp. 6) right.

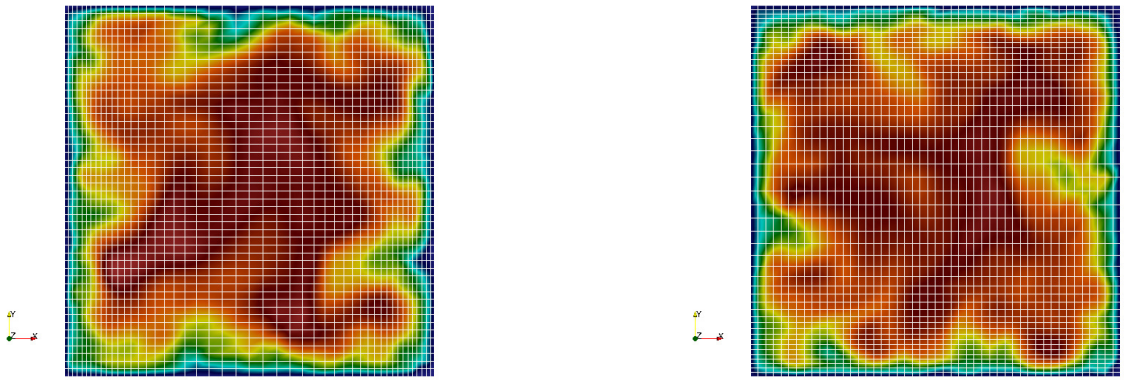

Figure 4: Instantaneous streamwise velocity. Left: stretching is applied in X and not in Y. Right: stretching is applied in Y and not in X.

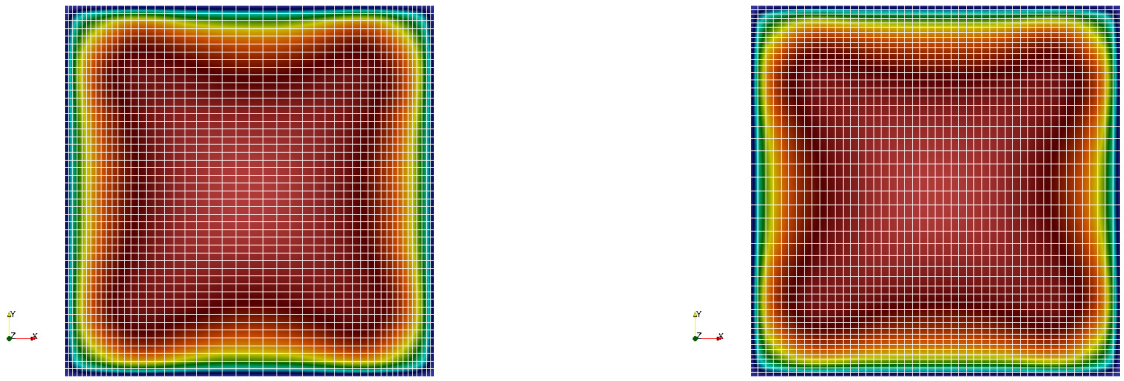

Figure 5: Mean streamwise velocity. Left: stretching is applied in X and not in Y. Right: stretching is applied in Y and not in X.

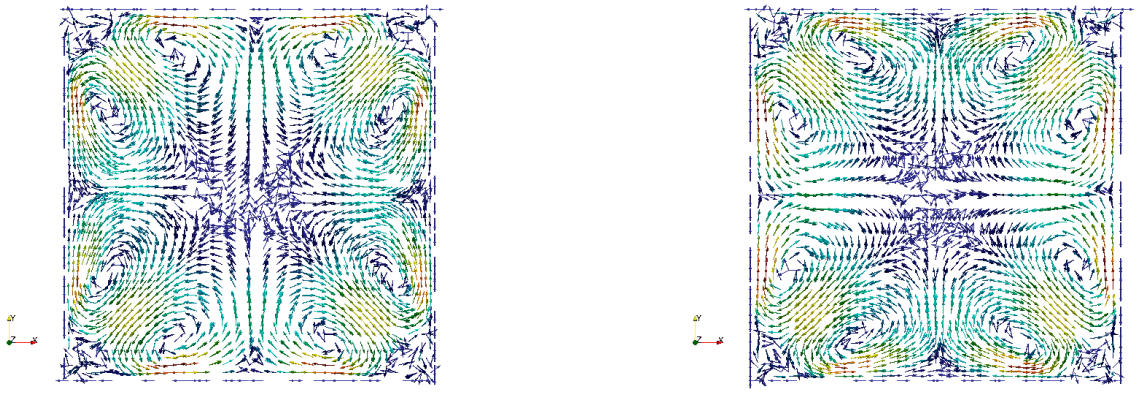

Figure 6: Secondary vortices. Left: stretching is applied in X and not in Y. Right: stretching is applied in Y and not in X.

4.3 Parallelism

When double stretching is used, parallelism with MPI is only carried out in the streamwise (Z) direction. Figure 7 shows the speedup for a $257 \times 257 \times 256$ node test case, where good performance is observed up to 64 MPI tasks, even if some speedup still happens for 128 MPI tasks. The machine is a Bull Sequana X1000 supercomputer.

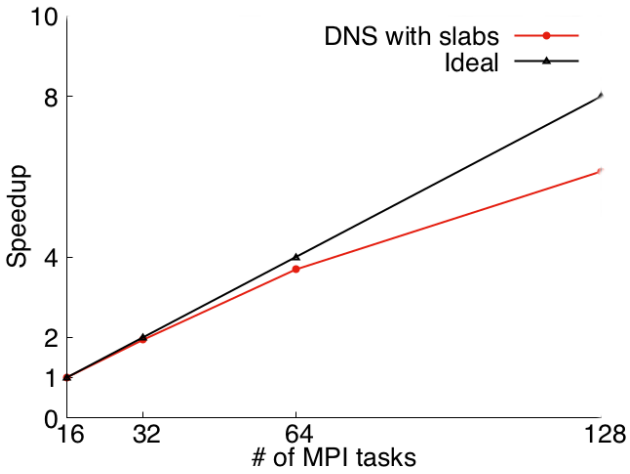

Figure 7: Speedup for a DNS using slabs for a $257 \times 257 \times 256$ node test case.

5 Final remarks

Adding stretching in a second direction of a high order finite difference has been briefly presented. Switching the streamwise (X) direction used with pencils to Z as the new streamwise direction allows to use X-Y slabs. Some first results have been obtained in the case of a laminar flow in a square duct with a grid with double stretching. More comparisons with [5] will be presented at the conference for turbulent flows, focusing on mean and secondary flows.

Acknowledgements

The authors would like to thank the Engineering and Physical Sciences Research Council (EPSRC) for financial support under the UK Turbulence Consortium (EP/L000261/1) grant. An award of computer time was provided by the Innovative and Novel Computational Impact on Theory and Experiment (INCITE) program. This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. The authors also would like to thank the Hartree Centre for granting access to their Bull machine.

References

[1] S. Laizet and E. Lamballais. High-order compact schemes for incompressible flows: A simple and efficient method with quasi-spectral accuracy. *J. Comp. Phys.*, 228:5989–6015, 2009.

[2] S. Laizet and N. Li. Incompact3d: A powerful tool to tackle turbulence problems with up to O(10^5) computational cores. *Int. J. for Num. Meth. in Fluids*, 67:1735–1757, 2001.

[3] S.K. Lele. Compact finite difference schemes with spectral-like resolution. *J. Comp. Phys.*, 103:16–42, 1992.

[4] E. Avital, N. Sandham, and K. Luo. Stretched Cartesian grids for solution of the incompressible Navier-Stokes equations. *Int. J. for Num. Meth. in Fluids*, 33:897–918, 2000.

[5] Y. Joung, S.-U. Choi, and J.-I. Choi. Direct numerical simulation of turbulent flow in a square duct: analysis of secondary flows. *J. Eng. Mech.*, 133:213, 2007.