Introduction to CHAPSim

Wei Wang

SCD, STFC-Daresbury Laboratory, UKRI

wei.wang@stfc.ac.uk

November 4, 2020

Overview

License

The GNU General Public License v3.0 (GPLv3)

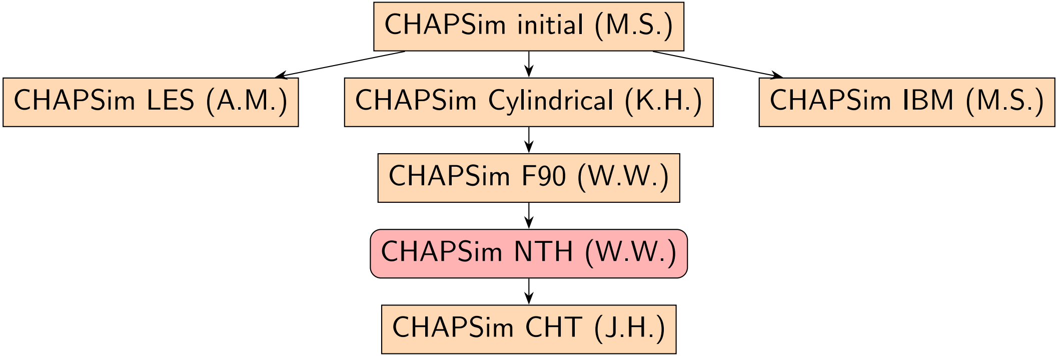
`https://www.gnu.org/licenses/quick-guide-gplv3.en.html`

Main Contacts

- Wei Wang, STFC-DL, UKRI. (wei.wang@stfc.ac.uk)
- Mehdi Seddighi, Liverpool John Moors University. (m.seddighi@ljmu.ac.uk)
- Shuisheng He, The University of Sheffield. (s.he@sheffield.ac.uk)

History of *CHAPSim*

A CHannel And Pipe flow Simulation solver *CHAPSim*) is an incompressible Direct Numerical Simulation (DNS) code for flow and heat transfer with MPI parallelization.



History of *CHAPSim*(cont'd)

Functions existed/under-development but not yet merged to the main branch:

- Large Eddy Simulation (LES) (Smagorinsky Model, Dynamic Geromano-Lily Model,, WALE model)
- Immersed Boundary Method (IBM) (for roughness)
- Conjugate Heat Transfer (CHT)
- Boundary layer developing flow (under development in UoS)
- Unsteady pulsating flow (under development in LJM)

Code Building

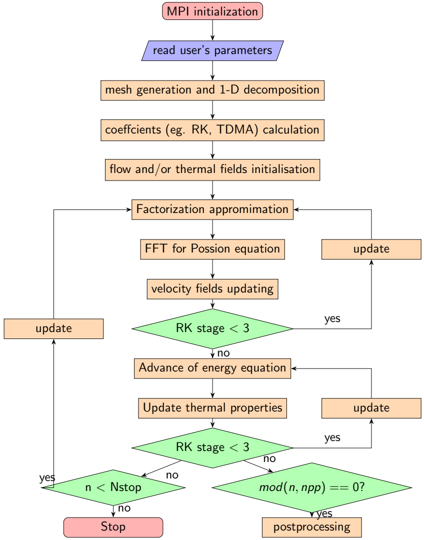
- Acquire the source code by cloning the git repository:
  ```
  $git clone git@github.com:WeiWangSTFC/CHAPSim.git
  ```
- Compile the codes:
  ```
  $mkdir bin obj
  $make all
  ```
- Debugging mode compiling:
  ```
  $make cfg=gnu
  $make cfg=intel
  ```

Numerical Methods

	Methods
Parallel	MPI
Mesh	Structured, generated on the fly
Spacial Discretization	Finite Difference
Nonlinear terms	Divergence form
	2nd order spacial accuracy
	explicit Runge-Kutta & Adams-Bashforth method for temporal discretization
Viscous terms	implicit Crank-Nicolson method for temporal discretization
Pressure	FFT and Fractional step method
Thermodynamics	Quasi-incompressible flow
	Thermal properties updated by table-searching or specified functions of temperature

Figure: Numerical Methods

Code Structure



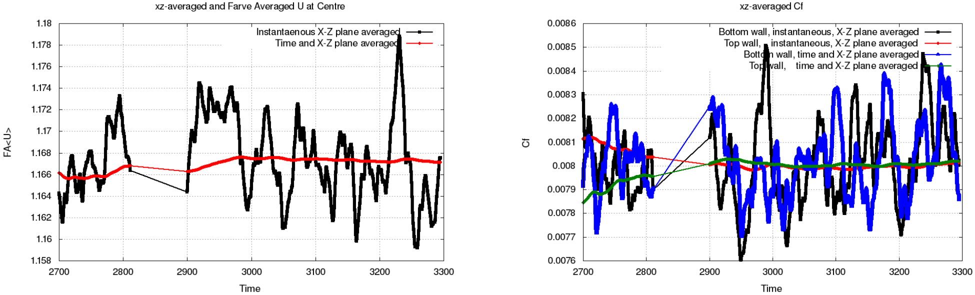
Directory Structure

bin	1 item	Folder	2020-10-27 15:45:41
docs	3 items	Folder	2020-10-21 16:57:44
lib	3 items	Folder	2020-10-15 10:44:27
obj	140 items	Folder	2020-10-27 15:45:41
scripts	13 items	Folder	2020-10-15 10:44:27
src	66 items	Folder	2020-11-04 01:02:10
test_cases	18 items	Folder	2020-10-19 10:46:32
test_cases_additional	10 items	Folder	2020-10-19 10:46:15
test_cases_template	17 items	Folder	2020-10-19 10:59:01
test_loop	2 items	Folder	2020-10-15 10:44:27
.git	12 items	Folder	2020-10-28 08:36:28
.vscode	1 item	Folder	2020-10-20 09:48:04
CHAPSim_workspace.code-workspace	60 bytes	Unknown	2020-10-22 20:54:37
LICENSE	35.1 kB	Text	2020-10-16 09:08:58
Makefile	6.5 kB	Text	2020-10-27 14:39:54
README.md	3.6 kB	Text	2020-10-26 16:46:50
.gitignore	394 bytes	Text	2020-10-23 14:38:02

Read in parameters

https://github.com/WeiWangSTFC/CHAPSim/blob/main/test_cases_template/TC4_Channel_IO_thermal_ScpWater/readdata.ini

Science and Technology Facilities Council



(a) Uc(t) (b) Cf(t)

Figure: Simulation Monitoring

Simulation - data storage

Instantanous Data

- u, v, w, p as function of (x, y, z, t)
- T, D, E, M, K as function of (x, y, z, t)
- Binary format for both restarting and further post processing.

RA/FA Data

- ASCII format. Tecplot format.
- for peridoc x and z, up to third order momentum and heat flux
- for peridoc z only, up to second order momentum and heat flux (* included in Jundi's version?)

Output Data explanation

The file format is '.dat', which can be opened by any text editor, like Notepad, Gedit, etc. Lines starting with '%' or '#' are comment lines.

Example file: **'Result.IO.undim.Profile.Flow.Favre.dat'**
Variables in this file are all dimensionless, scaled by the reference state labeled as 0, which are given in the file 'table.plt'. The variables in this file is dimensionless, scaled by the reference state, and Favre Averaged.

`Y`	$y = y^*/L0$, the scaled distance to the wall, $(-1, 1)$
`Y+`	$y^+ = y \cdot Re_\tau$, the distance scaled by wall parameter
`Utau`	U_τ, skin velocity for each side wall
`Ux`	$\widetilde{U}_x$, the streamwise velocity
`Uy`	$\widetilde{U}_y$, the wall-normal velocity
`Uz`	$\widetilde{U}_z$, the spanwise velocity
`P`	p, pressure

Output Data explanation (cont'd)

`TKE` $\widetilde{k} = \frac{1}{2}\left(\overline{\rho u'' u''} + \overline{\rho v'' v''} + \overline{\rho w'' w''}\right)$, turbulent kinetic energy

`Ruu` $\overline{\rho u'' u''}$, Favre Averaged Reynolds Stress

`Ruv` $\overline{\rho u'' v''}$, Favre Averaged Reynolds Stress

`Ruw` $\overline{\rho u'' w''}$, Favre Averaged Reynolds Stress

`Rvv` $\overline{\rho v'' v''}$, Favre Averaged Reynolds Stress

`Rvw` $\overline{\rho v'' v''}$, Favre Averaged Reynolds Stress

`Rww` $\overline{\rho w'' w''}$, Favre Averaged Reynolds Stress

$\mathtt{Ruv}_{\mathtt{vis}}$ $\mu \frac{\partial \widetilde{U}_x}{\partial y}$, Viscous shear stress

`dUdY` $\frac{\partial \widetilde{U}_x}{\partial y}$, Mean velocity gradient

Output Data explanation (cont'd)

Example file: **'Result.IO.undim.Profile.Heat.Transfer.dat'**

Variables in this file are all dimensionless, scaled by the reference state labeled as 0, which are given in the file 'table.plt'.

`Y`	$y = y^*/L0$, the scaled distance to the wall, $(-1, 1)$
`Y+`	$y^+ = y \cdot Re_\tau$, the distance scaled by wall parameter
`D`	$\overline{\rho}$, Mean Density
`T`	$\overline{T}$, Mean Temperature
$\mathtt{H_{ra}}$	$\overline{h}$, Reynolds Averaged Mean Enthalpy
$\mathtt{H_{fa}}$	$\widetilde{h}$, Favre Averaged Mean Enthalpy
`M`	$\overline{\mu}$, Mean Viscousity
`Drms`	$\sqrt{\overline{\rho'^2}}$, RMS of density
`Trms`	$\sqrt{\overline{T'^2}}$, RMS of Temperature
$\mathtt{Hrms_{ra}}$	$\sqrt{\overline{h'^2}}$, RMS of RA enthalpy
$\mathtt{Hrms_{ra}}$	$\sqrt{\overline{h''^2}}$, RMS of FA enthalpy

Output Data explanation (cont'd)

Science and Technology Facilities Council

`D(T)`	$\rho(\overline{T})$, Density, table-searched based on $\overline{T}$
`M(T)`	$\mu(\overline{T})$, Viscousity, table-searched based on $\overline{T}$
`K(T)`	$\kappa(\overline{T})$, thermal conductivity, table-searched based on $\overline{T}$
`Cp(T)`	$c_p(\overline{T})$, specific heat capacity, table-searched based on $\overline{T}$
`Pr(T)`	$Pr = \mu(\overline{T}) \cdot c_p(\overline{T})/\kappa(\overline{T})$, Prandtl number
$\mathtt{thfx}_{\mathtt{ra}}$	$\overline{\rho}\overline{u'h'}$, RA turbulent heat flux in the streamwise direction
$\mathtt{thfy}_{\mathtt{ra}}$	$\overline{\rho}\overline{v'h'}$, RA turbulent heat flux in the wall-normal direction
$\mathtt{thfz}_{\mathtt{ra}}$	$\overline{\rho}\overline{w'h'}$, RA turbulent heat flux in the spanwise direction
$\mathtt{thfx}_{\mathtt{fa}}$	$\overline{\rho u''h''}$, FA turbulent heat flux in the streamwise direction
$\mathtt{thfy}_{\mathtt{fa}}$	$\overline{\rho v''h''}$, FA turbulent heat flux in the wall-normal direction
$\mathtt{thfz}_{\mathtt{fa}}$	$\overline{\rho w''h''}$, FA turbulent heat flux in the spanwise direction

$\mathtt{qflux_x}$	$\overline{\kappa \frac{\partial T}{\partial x}}$	conductive heat flux in the streamwise direction
$\mathtt{qflux_y}$	$\overline{\kappa \frac{\partial T}{\partial y}}$	conductive heat flux in the wall-normal direction
$\mathtt{qflux_z}$	$\overline{\kappa \frac{\partial T}{\partial z}}$	conductive heat flux in the spanwise direction
$\mathtt{du_{per}}$	$\overline{\rho' u'}$	
$\mathtt{dv_{per}}$	$\overline{\rho' v'}$	
$\mathtt{dw_{per}}$	$\overline{\rho' w'}$	
$\mathtt{dh_{per}}$	$\overline{\rho' h'}$	

Simulation - data visualisationn

Contour check

- subroutine `PP-TEC360-DATA-CHECK.f90`
- stored in folder `5-instant-pltdata`
- format `tecplot 360` (tecplot, paraview)

Averaged contours check

- subroutine `PP-TEC360-DATA-CHECK-xzt.f90`
- stored in folder `4-averaged-pltdata`

Averaged data visualisation

- subroutine `WRT-TEC-AVERAGE-XZperiodic-XX.f90`
- stored in folder `4-averaged-pltdata`

Example - Channel (periodic streamwise)

The benchmark data for turbulence in a fully developed channel flow is the channel flow at $Re_\tau = 180$ (named "KMM180"). The reference data of [?] can be downloaded via `http://turbulence.ices.utexas.edu/MKM_1999.html`

DNS data	Kim et al. (1987)	Current DNS
Re_τ	180	180
Re_b	3300	3300
Method	Spectral	finite difference
Domain size	$4\pi\delta \times 2\delta \times 2\pi\delta$	$12.8\delta \times 2\delta \times 3.5\delta$
mesh size	$192 \times 128 \times 160$	$512 \times 192 \times 200$
Δy^+_{min}	0.05	0.2
Δy^+_{max}	4.4	3.5
Δx^+	12	4.5
Δz^+	7	3.15
Δt^+	not given	0.08
Averaging time	$10\Delta t^+$	$10\Delta t^+$

Example - Channel (cont'd)

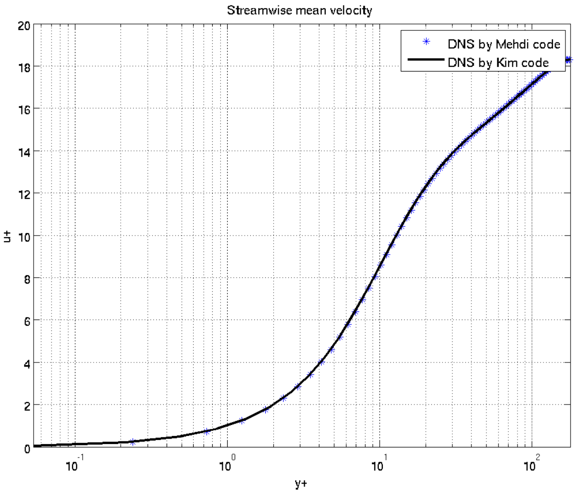


Figure: Streamwise velocity normalized by the friction velocity

Example - Channel (cont'd)

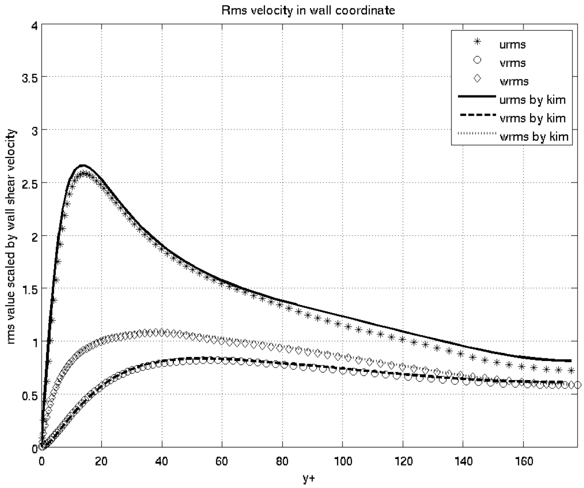


Figure: Root mean square of the streamwise, wall-normal and spanwise velocity fluctuations normalized

Example - Channel (cont'd)

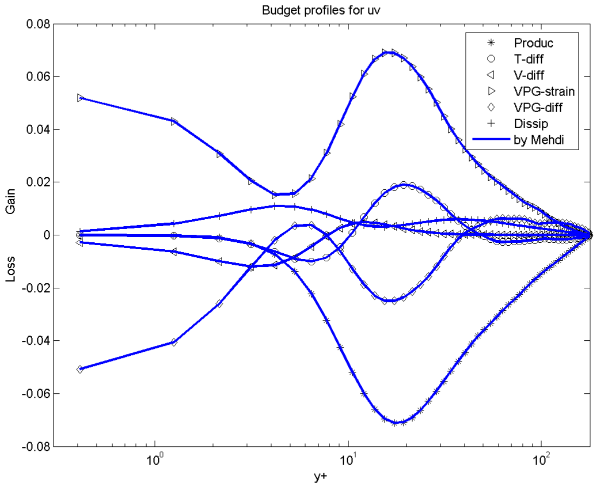


Figure: Transport of $\overline{u'v'}$

How to upload your version to Github?

- Work on your own branch
- Review your code before merging to the main branch

Website and Forum

- Website: `www.ccpnth.ac.uk` under development.
- Forum: `chapsim.slack.com`

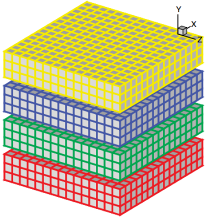
Work Plan Discussion

- Code Modularisation

```
PROCEDURE (DERIVATIVE_X) derx_00,derx_11,derx_12,derx_21,derx_22,&
      derxx_00,derxx_11,derxx_12,derxx_21,derxx_22
PROCEDURE (DERIVATIVE_X), POINTER :: derx,derxx,derxS,derxxS
PROCEDURE (DERIVATIVE_Y) dery_00,dery_11,dery_12,dery_21,dery_22
PROCEDURE (DERIVATIVE_Y), POINTER :: dery,deryS
PROCEDURE (DERIVATIVE_YY) &
      deryy_00,deryy_11,deryy_12,deryy_21,deryy_22
PROCEDURE (DERIVATIVE_YY), POINTER :: deryy,deryyS
PROCEDURE (DERIVATIVE_Z) derz_00,derz_11,derz_12,derz_21,derz_22,&
      derzz_00,derzz_11,derzz_12,derzz_21,derzz_22
PROCEDURE (DERIVATIVE_Z), POINTER :: derz,derzz,derzS,derzzS
```

Figure: Example of modules

- 2-D decompsition and MPI



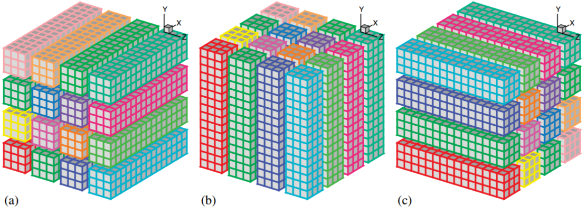


Figure 3. 2D domain decomposition example using a 4×3 MPI processes.

The End