DNS and LES Simulations for Particle-Laden Turbulent Flows with Two-Way Momentum Coupling

Maciej Manna

January 27, 2021

Acknowledgements

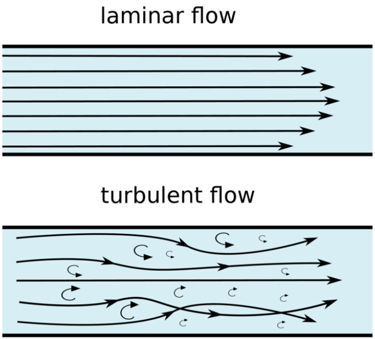
First half of a project made as part of:

- a project of the National Science Center (NCN) OPUS14/2017: *Turbulent flow analysis with the dispersion phase - the impact of two-sided coupling of momentum and gravity on particle motion statistics* supervised by Prof. Bogdan Rosa (IMGW-PIB);
- a MSc in Computer Science on Jagellonian University supervised by Prof. Bogdan Rosa (IMGW-PIB) and Dr Sylwester Arabas (UJ);
- also, computational resources and access to supercomputer *Okeanos* thanks to Interdisciplinary Centre for Mathematical and Computational Modeling (ICM) at Warsaw University (grant GA73-14).

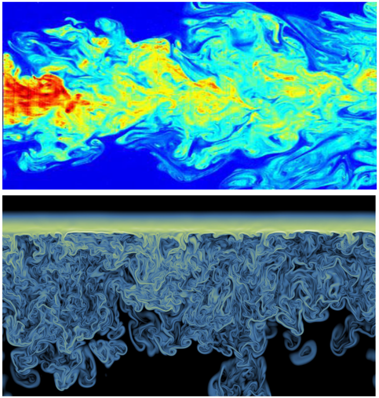
Talk Outline

Introduction

Turbulent Flow



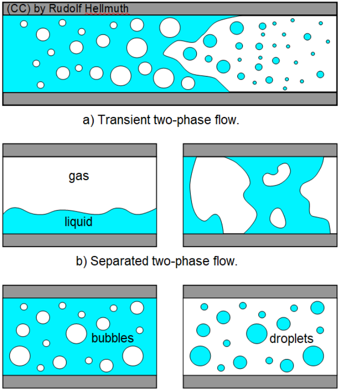
Source: www.cfdsupport.com/OpenFOAM-Training-by-CFD-Support/node334.html



Source: https://www.nortekgroup.com/knowledge-center/wiki/new-to-turbulent-flow-1

Turbulent flow, "not steady", involves fluctuations and vortices at different scales, that get smaller when energy is dissipated due to viscosity (*energy cascade*).

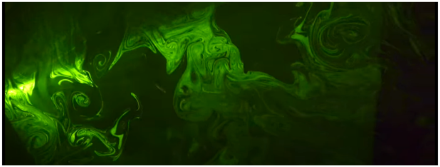
Particle-Laden Flow



a) Transient two-phase flow.

b) Separated two-phase flow.

c) Dispersed two-phase flow.



Source:https://www.youtube.com/watch?v=_UoTTq651dE&ab_channel=3Blue1Brown

Source: https://www.infectioncontroltoday.com/view/sneezing-produces-complex-fluid-cascade-not-simple-spray

Particle-laden flow, with two phases interacting with each other: *continuous* (fluid) and *disperse* (particles, that are insoluble and immiscible).

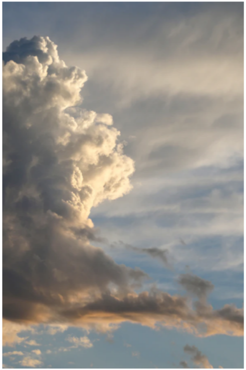
Real-Life Applications

INDUSTRY TECHNOLOGY

– combustion of pulverized coal
– transport in pipelines (bubbles)
– combustion of liquid fuel
– furnace dust handling

ECOLOGY AGRICULTURE

– spraying of fertilizers, etc.
– transport of pollutants
– transport of sediments in water



METEOROLOGY CLIMATE SCIENCE

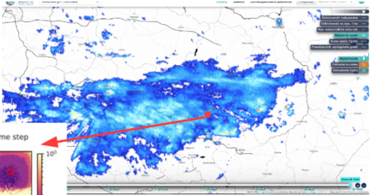
– modelling of atmospheric clouds (i.e. small aerosol particles dispersed in the air)

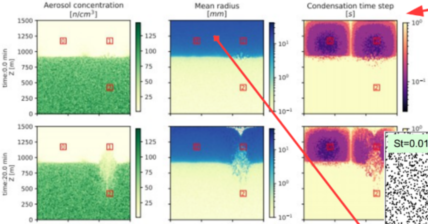
Sources: http://www.chinaeaf.com/products/Dust-collection-system.html; https://www.agairupdate.com/birtle-crop-dusters-add-firefighting-to-their-profile/; Photo by David Ballew (see: https://unsplash.com/photos/pH6-eomaijQ)

Cloud Modelling

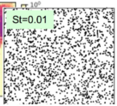
Atmospheric Cloud Modelling

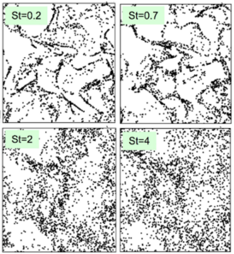
IMGW-PIB METEO Forecasting Model
Domain size: hundreds/thousands of kms
Grid cell size (horizontal): ~2km
(B. Rosa et al.)

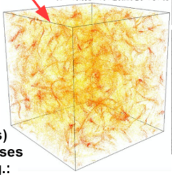




PySDM Cloud Modelling Package (P. Bartman)
Domain size: few kms
Grid cell size: less then m to few m



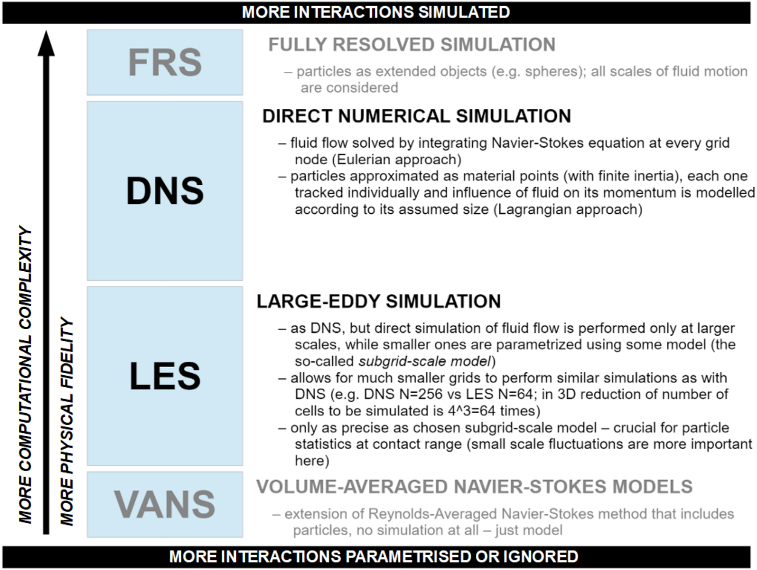




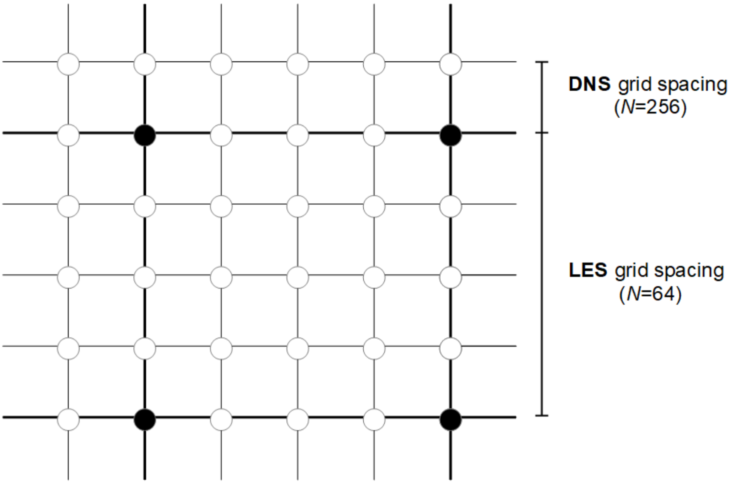
Our interest:
Domain size: less then m to few m
Grid cell size: order of mm and less

- **homogeneous and isotropic turbulence (HIT)**
- **observation of phenomena specific to this system (e.g. droplet/particle clustering)**
- **calculation of statistics (e.g. collision kernels) that are needed to parametrize cloud processes In Numerical Weather Prediction models (e.g.: COSMO, WRF, Aladin)**

Simulation Methods



DNS vs LES – Grid Density



Modelling Fluid-Particle Interactions

FLOW-ONLY SIMULATION

– uses 3D Navier-Stokes equation (momentum equation for fluids):

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{U}\times\boldsymbol{\omega} - \nabla\left(\frac{P}{\rho} + \frac{1}{2}\mathbf{U}^2\right) + \nu\nabla^2\mathbf{U} + \mathbf{f}(\mathbf{x},t) + \mathbf{f}^{(p)}$$

– due to simple domain shape (3D box) and boundary conditions (periodic), very convenient *spectral method* may be used to solve N-S equation

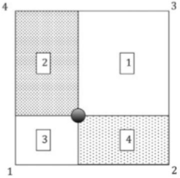
ONE-WAY MOMENTUM COUPLING

– momentum is transferred from fluid to particles

$$\frac{d\mathbf{V}^i(t)}{dt} = -f(Re_p)\frac{\mathbf{V}^i(t) - \mathbf{U}(\mathbf{Y}^i(t),t)}{\tau_p} + \mathbf{g} \qquad \frac{d\mathbf{Y}^i(t)}{dt} = \mathbf{V}^i(t)$$

TWO-WAY MOMENTUM COUPLING

– momentum is also transferred from particles to fluid in neighbouring grid nodes (split according to proximity of tracked particle to surrounding 8 grid nodes (projection onto neighbouring nodes (PNN) method)



FOUR-WAY MOMENTUM COUPLING

State of Current Research

Study		Flow	Grid size (N)	DNS	LES	Two-way coupling	Gravity	Collisional statistics
Year	Author(s)							
1990	Squires & Eaton	HIT	64	✓	✗	✓	✗	✗
1993	Elghobashi & Truesdell	decaying turbulence	96	✓	✗	✓	✓	✗
1993	Wang & Maxey	HIT	96	✓	✓	✗	✓	✗
1998	Boivin *et al*	HIT	96	✓	✗	✓	✗	✗
1998	Yang & Lei	HIT	DNS: 96 LES:128	✓	✓	✗	✓	✗
1999	Sundaram & Collins	decaying turbulence	128	✓	✗	✓	✗	✗
2000	Boivin *et al.*	HIT	LES: 96	✓	✓	✓	✗	✗
2003	Vermorel *et al.*	particle-laden slab	128	✓	✗	✓	✗	✗
2006	Bosse *et al.*	HIT	64	✓	✗	✓	✓	✗
2008	Ayala *et al.*	HIT	128	✓	✗	✗	✓	✓
2010	Jin *et al.*	HIT	DNS: 256 LES: 64	✓	✓	✗	✓	✓
2011	Dejoan	HIT	512	✓	✗	✓	✓	✗
2012	Abouelmagd & Lee	HIT & decaying turb.	128	✓	✗	✓	✗	✗
2013	Rosa *et al.*	HIT	1024	✓	✗	✗	✓	✓
2013	Onishi *et al.* (a)	HIT	2000	✓	✗	✗	✓	✓
2014	Good *et al.*	HIT	512	✓	✗	✗	✓	✗
2016	Ireland *et al.*	HIT	2048	✓	✗	✗	✗	✓
2016	Rosa *et al.*	HIT	1024	✓	✗	✗	✓	✗
2017	Breuer & Hoppe	turbulent channel	128	✗	✓	✓	✓	✗
2017	Rosa *et al.*	HIT	256	✓	✓	✗	✓	✓
2017	Monchaux & Dejoan	HIT	64	✓	✗	✓	✓	✗
2018-	**proposed study**	**HIT**	**up to 1024**	**✓**	**✓**	**✓**	**✓**	**✓**

Goals of This Study

- DNS simulations require a lot of computational resources, and still, even on such small scales, are limited to grid sizes that are not always sufficient;
- LES simulations might be an alternative that significanltly reduces required grid size to obtain similar resolution and amount of turbulence (Reynolds number);
- we need to rigorously study:
  1. whether LES simulations provide accurate enough results to serve as a valid replacement; and
  2. what exact gains in performance LES simulations provide.

Numerical Method

Fluid Flow Simulation

- fluid is simulated on $N \times N \times N$ **3D cube** (for convenience we assume its edge length: $L = L_x = L_y = L_z = 2\pi$);
- we use **periodic boundary conditions**;
- **Eulerian approach** is used to model fluid properties, i.e. its velocity field (calculated for N^3 grid points);
- flow is considered to be **incompressible** (incompressible fluid vs flow);
- fluid velocity is simulated by integrating **3D Navier–Stokes equation** in its rotational form.

Rotational N–S Equation in 3D

It consists of *continuity* and *momentum* equation, i.e.:

$$\begin{cases} \nabla \cdot \mathbf{u} = \mathbf{0} \\ \frac{\partial}{\partial t}\mathbf{u} = \mathbf{u} \times \omega - \nabla\left(\frac{p}{\rho} + \frac{u^2}{2}\right) + \nu \nabla^2 \mathbf{u} + \mathbf{f} \end{cases}$$

where:

- $\mathbf{u} := \mathbf{u}(\mathbf{x}, t)$ is a **fluid velocity field** ($u^2 = \mathbf{u} \cdot \mathbf{u}$),
- $\omega := \omega(\mathbf{x}, t) = \nabla \times \mathbf{u}(\mathbf{x}, t)$ is its **vorticity**,
- $\mathbf{f} := \mathbf{f}(\mathbf{x}, t)$ is an **external, large scale force** (acting on large scales, i.e. affecting low wavenumbers),
- p is **pressure**,
- ρ is **density of a fluid** (constant due to incompressibility);
- ν is **dynamic viscosity** of a fluid.

N–S Equation in Spectral Space

For convenience we define:

- $\mathbf{N_1} := \mathbf{u} \times \omega$
- $N_2 := \frac{p}{\rho} + \frac{u^2}{2}$

Then, our *momentum equation* in **spectral space** (i.e. after Fourier transfor) is:

$$\frac{\partial}{\partial t}\hat{\mathbf{u}} = \hat{\mathbf{N}}_\mathbf{1} - i\mathbf{k}\hat{N}_2 - \nu k^2\hat{\mathbf{u}} + \hat{\mathbf{f}}\,.$$

Pseudo-Spectral Method

We use **pseudo–spectral method**, i.e. most of calculations are performed in spectral space, but some are done in "physical" space. To describe one entire step of evolution in time of our model, we assume thet we solved our problem for time steps up to n, i.e. we have calculated:

$$\hat{\mathbf{u}}(\mathbf{k}, \Delta t),\ \hat{\mathbf{u}}(\mathbf{k}, 2\Delta t),\ \ldots,\ \hat{\mathbf{u}}(\mathbf{k}, n\Delta t)\,.$$

Below, we present procedure to solve for $\hat{\mathbf{u}}(\mathbf{k}, (n+1)\Delta t)$ that consists of four steps...

Pseudo-Spectral Method (Step 1)

We obtain physical velocity field $\mathbf{u}$ from the one we have in spectral space by applying inverse fast Fourier transform, i.e.:

$$\hat{\mathbf{u}}(\mathbf{k}, n\Delta t) \xrightarrow{\mathsf{FFT}^{-1}} \mathbf{u}(\mathbf{x}, n\Delta t)\,.$$

Pseudo-Spectral Method (Step 2)

We calculate vorticity $\hat{\omega}$ in spectral space and then use inverse fast Fourier transform to convert it back to physical space, that is:

$$\hat{\omega}(\mathbf{k}, n\Delta t) = i\mathbf{k} \times \hat{\mathbf{u}}(\mathbf{k}, n\Delta t) \xrightarrow{\mathsf{FFT}^{-1}} \omega(\mathbf{x}, n\Delta t)\,.$$

Pseudo-Spectral Method (Step 3)

We compute nonlinear vector term $\mathbf{N}_1$ in physical space for n-th time step using vorticity obtained in step 2, and, consequently, use fast Fourier transform to get that term in spectral space, i.e.:

$$\mathbf{N}_1(\mathbf{x}, n\Delta t) = \mathbf{u}(\mathbf{x}, n\Delta t) \times \omega(\mathbf{k}, n\Delta t) \xrightarrow{\text{FFT}} \hat{\mathbf{N}}_\mathbf{1}(\mathbf{k}, n\Delta t)\,.$$

Pseudo-Spectral Method (Step 4)

Finally, having calculated all important terms, we evolve our velocity field in spectral space in time to obtain $\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t)$ using properly discretized version of Navier-Stokes equation in that space, which takes the following form:

$$\begin{aligned}\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t)-\hat{\mathbf{u}}(\mathbf{k},n\Delta t) &= \frac{3}{2}\hat{\mathbf{N}}_{\mathbf{1}}(\mathbf{k},n\Delta t)\Delta t-\frac{1}{2}\hat{\mathbf{N}}_{\mathbf{1}}(\mathbf{k},(n-1)\Delta t)\Delta t\\ &- i\mathbf{k}\hat{N}_2(\mathbf{k},(n+1)\Delta t)\Delta t\\ &- \frac{\nu k^2}{2}\left[\hat{\mathbf{u}}(\mathbf{k},n\Delta t)+\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t)\right]\Delta t\\ &+ \hat{\mathbf{f}}(\mathbf{k},n\Delta t)\Delta t\,,\end{aligned}$$

where we apply: (1) *Adams-Bashforth scheme* for terms involving $\hat{\mathbf{N}}_{\mathbf{1}}$, (2) *Crank-Nicholson scheme* for diffusive term (third line on the right hand side of above equation), and (3) simple *Euler scheme* for forcing term including $\hat{\mathbf{f}}$.

Pseudo-Spectral Method (Step 4) – Details (1)

In practice, we first apply explicit terms (not referring to $(n+1)$-th time step), i.e.:

$$\hat{\mathbf{l}}(\mathbf{k}) = \frac{3}{2}\hat{\mathbf{N}}_\mathbf{1}(\mathbf{k}, n\Delta t)\Delta t - \frac{1}{2}\hat{\mathbf{N}}_\mathbf{1}(\mathbf{k}, (n-1)\Delta t)\Delta t - \frac{\nu k^2}{2}\hat{\mathbf{u}}(\mathbf{k}, n\Delta t)\Delta t + \hat{\mathbf{u}}(\mathbf{k}, n\Delta t)\,,$$

and than, also forcing term:

$$\hat{\mathbf{l}}_\mathbf{f}(\mathbf{k}) := \hat{\mathbf{l}}(\mathbf{k}) + \hat{\mathbf{f}}(\mathbf{k}, n\Delta t)\Delta t\,;$$

Pseudo-Spectral Method (Step 4) – Details (2)

This leaves us with remaining terms that come from implicit schemes:

$$\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t) = \hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k}) - i\mathbf{k}\hat{N}_2(\mathbf{k},(n+1)\Delta t)\Delta t - \frac{\nu k^2}{2}\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t)\Delta t\,.$$

Using discretised *continuity equation* in spectral form ($i\mathbf{k}\cdot\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t) = \mathbf{0}$), we obtain:

$$i\mathbf{k}\cdot\hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k}) + k^2\hat{N}_2(\mathbf{k},(n+1)\Delta t)\Delta t = \mathbf{0}\,,$$

or, equivalently:

$$\hat{N}_2(\mathbf{k},(n+1)\Delta t)\Delta t = \frac{i\mathbf{k}\cdot\hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k})}{k^2}\,.$$

Pseudo-Spectral Method (Step 4) – Details (3)

We can substitute above results to obtain:

$$\left(1+\frac{\nu k^2 \Delta t}{2}\right)\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t)=\hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k})-\mathbf{k}\frac{\mathbf{k}\cdot\hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k})}{k^2}$$

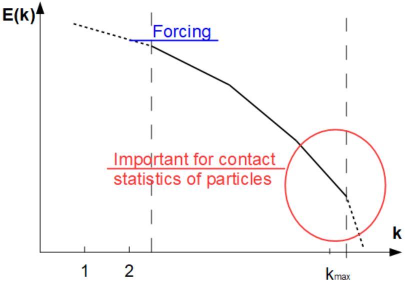
which can be solved to get final expression for $\hat{\mathbf{u}}$ in $(n+1)$-th time step, namely:

$$\hat{\mathbf{u}}(\mathbf{k},(n+1)\Delta t)=\frac{\hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k})-\mathbf{k}\frac{\mathbf{k}\cdot\hat{\mathbf{I}}_{\mathbf{f}}(\mathbf{k})}{k^2}}{1+\frac{\nu k^2\Delta t}{2}}.$$

Pseudo-Spectral Method – Pros and Cons

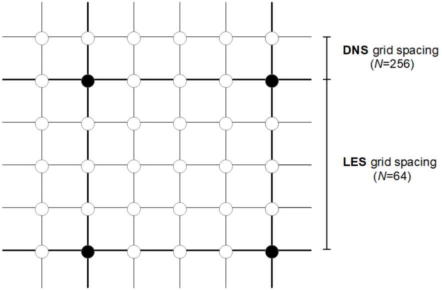
+ easy application of external, large–scale forcing;
+ works well with periodic boundary conditions;
+ relatively precise and stable method;
- requires a lot of Fourier transform (computationally expensive and non–local).

Forcing Term



- necessary to preserve homogenous and isotropic turbulence,
- **deterministic scheme** – at each time step set energies for first two wavenumbers to preset values ($E(1) = 0.55544$ and $E(2) = 0.159843$, respectively), that will cascade to smaller scales (higher wavenumbers).

Large Eddy Simulations (LES)



- small–scale structures are not directly resolved in simulations (N is smaller, but k_{max} is smaller as well), but they are parametrised using **subgrid–scale model**,
- quality of that model is important – small–scale interactions are most important for statistics we are calculating.

Large Eddy Simulations (LES) – Subgrid–Scale Model (1)

- we use common subgrid–scale model, based on the concept of spectral eddy viscosity (ν_e) introduced in S. B. Pope, *Turbulent Flows* (CUP, 2000);
- our *momentum equation* is:

$$\left[\frac{\partial}{\partial t} + (\nu + \nu_e(k|k_c))k^2\right] \hat{\mathbf{u}}(\mathbf{k}, t) = \mathbf{P}(\mathbf{k}) \cdot \mathrm{FFT}(\mathbf{u} \times \omega) + \hat{\mathbf{f}}(\mathbf{k}, t)\,,$$

  where: $\mathbf{P}$ – projection tensor, and k_c – maximum wavenumber resolved in simulation (here $k_c = \lfloor \frac{N-3}{2} \rfloor$).
- actual *subgrid–scale model* is represented here by the term:

$$\nu_e(k|k_c)k^2\hat{\mathbf{u}}(\mathbf{k}, t)$$

Large Eddy Simulations (LES) – Subgrid–Scale Model (2)

Here we use the following, standard subgrid–scale model, where spectral eddy viscosity is defined by:

$$\nu_e(k|k_c) = \nu_e^+(k|k_c)\sqrt{\frac{E(k_c)}{k_c}}\,,$$

where $E(k_c)$ is energy at cutoff wavenumber (calculated from simulation at each time step), and $\nu_e^+(k|k_c)$ is given by:

$$\nu_e^+(k|k_c) = C_K^{-3/2}(0.441 + 15.2\, e^{-3.03\frac{k_c}{k}})\,.$$

Other subgrid–scale models exist and are still developed.

A priori LES (filtered DNS)

Peculiar method between DNS and LES:

- we perform all simulations as in standard DNS (larger grid),
- then we choose some cutoff wavenumber k_{cutoff}, and apply (in spectral space) simple filter that zeroes all smaller wavenumbers (so that $k = 0$ for $k_{\text{cutoff}} \leq k \leq k_{\text{max}}$) before calculating Stokes drag (momentum transfer from fluid to particles),
- counterproductive for calculation of statistics we are interested in.

Simulating Particles

- particles are resolved using **Lagrangian approach** (every particle is labeled and its velocity and position is tracked in each time step);
- particles are approximated as material points;
- no interparticle forces are modelled in this approach;
- we usually measure amount of particles in the system usin **mass loading** Φ_m, definded as a ratio of masses of all particles and total mass of fluid;
- depending on how dilute our system is, we take into accout more interactions resulting in transfer of momentum.

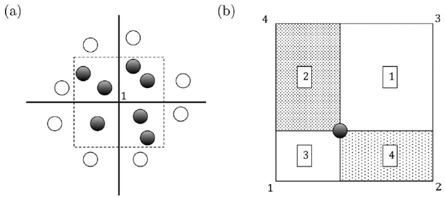
Momentum Transfer from Fluid to Particles (One-Way Coupling)

$$\frac{d\mathbf{V}^i(t)}{dt} = -f(Re_p)\frac{\mathbf{V}^i(t) - \mathbf{U}\big(\mathbf{Y}^i(t), t\big)}{\tau_p} + \mathbf{g}$$

$$\frac{d\mathbf{Y}^i(t)}{dt} = \mathbf{V}^i(t)$$

- i – particle number (label),
- τ_p – Stokes inertial response time,
- $\mathbf{V}^i(t) = \dot{\mathbf{Y}}^i(t)$ – particle velocity,
- $\mathbf{V}^i(\mathbf{Y(t)}, t)$ – fluid velocity at particle position (6 point Lagrangian interpolation),
- $\mathbf{g}$ – gravitational acceleration,
- $Re_p = 2aV_{rel}/\nu$ – particle Reynolds number (V_{rel} – particle-fluid relative velocity);
- f – drag correction factor (here: $f = 1$).

Momentum Transfer from Particles to Fluid (Two-Way Coupling)(1)



- especially when gravity is concerned, it is often assumed that its influence is stronger than Stokes drag, hence inclusion of momentum transfer is relevant in modelling clouds;
- influence of particles on grid points may be determined by one of two methods: *particle–in–cell* (PIC) or *projection onto nearest neighbours* (PNN) – study by Rosa et al. (2020) shows that particle statistics do not depend much on that choice (we use PNN exclusively);
- new term – $\mathbf{f}^{(p)}$, i.e. total force from particles – added to momentum equation for fluid (N–S).

Momentum Transfer from Particles to Fluid (Two-Way Coupling)(2)

$$\mathbf{f}^{(p)}(\mathbf{x},\mathrm{t}) = -\frac{M}{\rho}\sum_{\mathrm{i}=1}^{N_c} m_p^i \left(\frac{\mathbf{U}\left(\mathbf{Y}^i, t\right) - \mathbf{V}^i(t)}{\tau_p} + \mathbf{g} \right) \delta\left(\mathbf{x} - \mathbf{Y}^i\right)$$

$$= -\phi_m(\mathbf{x}, t) \left(\frac{\mathbf{U}(\mathbf{x}, t) - \mathbf{V}_E(\mathbf{x}, t) + \mathbf{V}_{ET}(\mathbf{x})}{\tau_p} \right)$$

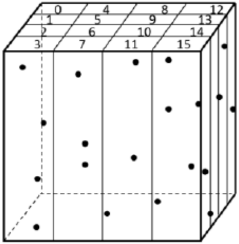
- δ – Dirac delta,
- ϕ_m – local mass loading of particles,
- $\mathbf{V}_E(\mathbf{x}, t)$ – Eulerian particle velocity (calculated from Lagrangian velocities using chosen method/kernel – PIC or PNN),
- $\mathbf{V}_{ET}(\mathbf{x}, t)$ – as above, but terminal velocity.

Superparticle Parametrization

- often, the computational cost of running simulations with higher mass loading is prohibitive (e.g. in this study maximal number of particles simulated was 20M);
- to achieve higher mass loadings, we introduce so–called **superparticles** (*computational particles*, *parcels*) – i.e. single simulated particle represents "parcel" containing multiple particles, that is tracked as one;
- in practice, we have superparameter $M \leq 1$ that is a mass multiplier used for particles that participate in momentum transfer (i.e. one particle is treated as the cluster of M particles);
- if $M = 1$, no superparticle parametrization is used;
- other studies (e.g. Rosa et al. 2020) show that such parametrization (especially for larger values of M) affects precision of simulation and obtained statistics (exact influence has not yet beed rigorously studied).

Notes on Implementation

- simulation code implemented using Fortran language for parallel processing using `mpich`;
- code utilizes two–dimensional division into computational subdomains handled by separate processes;
- parallel FFT implementation provided by `fftw` library.



Notes on Hardware

- simulations run on supercomputer *Okeanos* as part of a grant at ICM UW;
- *Okeanos* – Cray XC40; nodes: 1024; cores/node: 24 (total cores: 26016, each core: 2.6 GHz); memory/node 128 GB;
- every simulation run with number of nodes (DNS – 11, LES – 3) to guarantee at least 1 core per process;
- here every computational subdomain has dimensions: $16 \times 16 \times 256$ for DNS and $16 \times 16 \times 64$ for LES.

Accuracy of DNS and LES (Results)

Flow-only Simulations

- common initial step is to perform simulations without any particles, only integrating fluid flow so it reaches statistically stationary state;
- fluid–only smulations are dimendsionless;
- it has two main purposes:
  - acquire statistically stationary velocity field to serve as input for further particle-laden simulations,
  - calculate scale statistics that are used as parameters in further simulations;
- key flow parameters:
  - N – *grid nodes per dimension*, determine size of the grid, here: 256 for DNS, and 64 for LES;
  - ν – *numerical viscosity*, dimensionless quantity that simulates viscosity (internal friction) of a fluid which influences creation and dissipation of vortices in the flow (turbulence);
  - Δt – *size of time step in seconds*, needs to be tuned to achieve numerical stability of a system (small enough CFL, Courant number).

Flow Statistics (1)

	NJP'13-det-flow	dns-det-flow	les-det-flow
Type	DNS-D	DNS-D	LES-D
N	256	256	64
ν	1.1×10^{-3}	1.5×10^{-3}	1.5×10^{-3}
δt	9.0×10^{-4}	9.0×10^{-4}	9.0×10^{-4}
u'	0.868 ± 0.002	0.873 ± 0.001	0.840 ± 0.002
ϵ	0.200 ± 0.003	0.212 ± 0.002	0.167 ± 0.003
R_λ	196.87 ± 0.78	165.91 ± 0.51	173.01 ± 0.73
$k_{\max}\eta_k$	1.143 ± 0.005	1.423 ± 0.004	—
η_k	$9.033 \pm 0.370 \times 10^{-3}$	$1.125 \pm 0.003 \times 10^{-2}$	$1.194 \pm 0.005 \times 10^{-2}$
τ_k	$7.420 \pm 0.620 \times 10^{-2}$	$8.436 \pm 0.043 \times 10^{-2}$	$9.509 \pm 0.073 \times 10^{-2}$
L_s	1.496 ± 0.005	1.462 ± 0.003	1.550 ± 0.004
λ	$2.494 \pm 0.015 \times 10^{-1}$	$2.851 \pm 0.011 \times 10^{-1}$	$3.091 \pm 0.018 \times 10^{-1}$
T_e	3.774 ± 0.045	3.616 ± 0.028	4.252 ± 0.050
$\mathcal{S}$	$-5.090 \pm 0.010 \times 10^{-1}$	$-5.143 \pm 0.008 \times 10^{-1}$	—
$\mathcal{F}$	6.046 ± 0.046	5.916 ± 0.029	—
CFL	$0.26 \pm (< 0.01)$	0.3 ± 0.0	0.1 ± 0.0

Table 1: Fluid flow statistics for simulations with deterministic forcing scheme. Second and third column show results for DNS and LES simulations, respectively. First column contains statistics for DNS with similar parmeters from Rosa et al. 2013 (NJP).

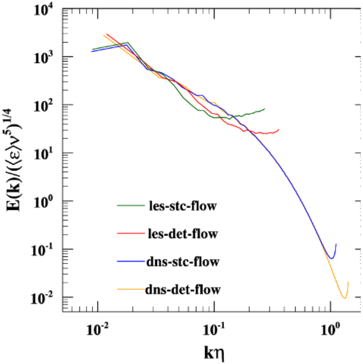
Flow Statistics (2)

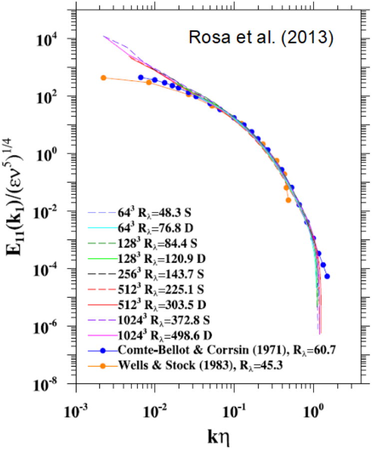
- u' – Reynolds microscale fluctuating velocity (standard deviation of fluid velocity (U), related to the intensity of turbulence);
- ϵ – energy dissipation rate (rate at which turbulence kinetic energy is converted into thermal internal energy, describes "energy cascade");
- η, τ_k – Kolmogorov length and time (smallest scales in turbulent flow, $\eta = (\nu^3 \epsilon^{-1})^{1/4}$, $\tau_k = (\nu \epsilon^{-1})^{1/2}$);
- λ – transverse Taylor microscale (describes intermediate length scales, $\lambda = \sqrt{15 \nu \epsilon^{-1}} u'$);
- L_S – integral length scale (large length scale, related to autocorrelation);
- T_e – eddy turnover time (rotation time of largest vortices; time scale of energy transfer between different scales).

Flow Statistics (2)

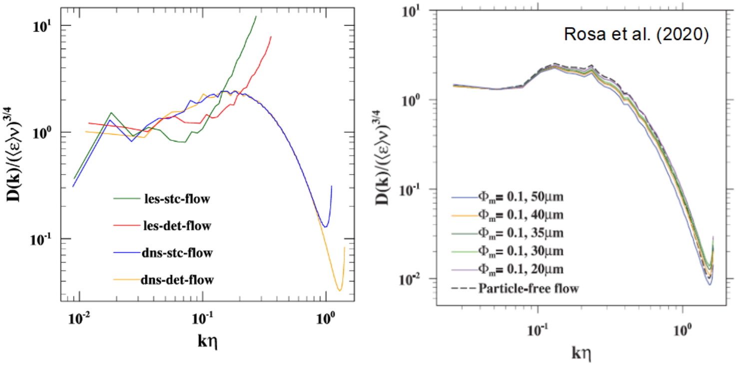
- target and control statistics:
  - R_λ – Taylor microscale Reynolds number (at intermediate length scales, at which viscosity significantly affects dynamics of the flow, $R_\lambda = u'\lambda/\nu$) – the bigger, the better, as it signifies better resolution,
  - $k_{\text{max}}\eta$ – spatial resolution parameter – should be greater than 1, but the closer it is to 1, the better,
  - CFL – Courant number – values not greater than 0.3 are required for stability of pseudo–spectral method (may be controlled by increasing time step, δt);
- other statistics – $\mathcal{F}$, $\int$ – flatness and skewness of the velocity gradient, respecively.

Energy Spectra





Energy Dissipation Rates



Simulations with Particles

- introduction of particles with given radii (a), makes given system dependent on actual physical dimensions;
- we introduce known from experiments parameters of studied environments (in this case, *warm advective clouds*), that can be used to bridge between dimensionless turbulent flow quantities and actual physical magnitudes;
- here we use, e.g. $\epsilon_{CLOUD} = 400\,\mathrm{cm}^2\mathrm{s}^{-3}$ – energy dissipation rates in clouds;
- here we use, e.g. $\nu_{AIR} = 0.17\,\mathrm{cm}^2\mathrm{s}$ – energy dissipation rates in clouds;
- using proportions we may calculate flow scale parameters (e.g. η, τ_k), and other statistics in physical space with actual physical units;
- e.g. for simulations presented below, size of a box (entire domain) may be calculated to be around 30–35cm.

Radial Distribution Function (1)

- measure of the effect of preferential concentration of droplets on the collision rate;
- studies (theoretical, numerical, experimental) show that for small separation $r < \eta$ the following power–law dependence holds:

$$g(r) = c_0(\eta/r)^{c_g(St,S_V)} ,$$

where exponent depends on Stokes' number $(St = \tau_p/\tau_k)$ and settling velocity $(S_V = v_p/v_k)$; c_0 – power law pre–factor.

Radial Distribution Function (2)

Basic computation method at contact distances, i.e. for $r = R := a_1 + a_2$. In such case, RDF may be directly computed from definition as:

$$g_{ii}(r;t) = \frac{2N_{pairs}}{N_i(N_i - 1)} \frac{V_s}{V_B}$$

where:

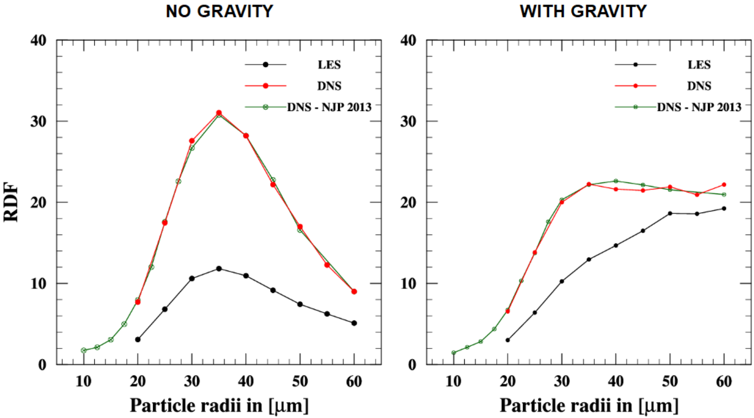
- i – particle radius category;
- N_{pairs} – total number of pairs detected with separation distance r falling into spherical shell of radii $R - \delta$, $R + \delta$ (with small δ, e.g. $\delta = 0.01r$);
- N_i – total number of particles with radius i;
- V_s – volume of probed spherical shell;
- V_B – volume of entire domain (here: $8\pi^3$).

Finally, $g_{ii}(r;t)$ is averaged over time to get $g_{ii}(r = R)$.

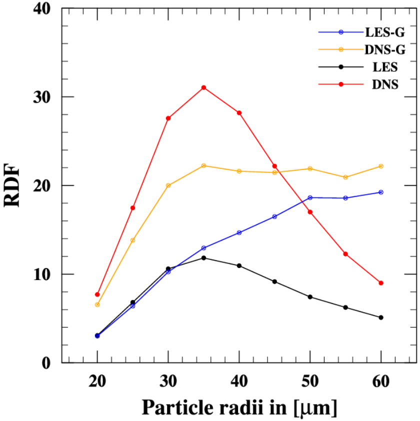
Radial Distribution Function (3)

Here, to achieve better results, we calculate above quantity, but for more separation distances than only R (180 shells/bins in range of $[R, 10R]$), then we fit obtained data into power law. This allows for better estimation of g_{11} at contact distance (R).

[OWC] RDF Results (1)



[OWC] RDF Results (2)

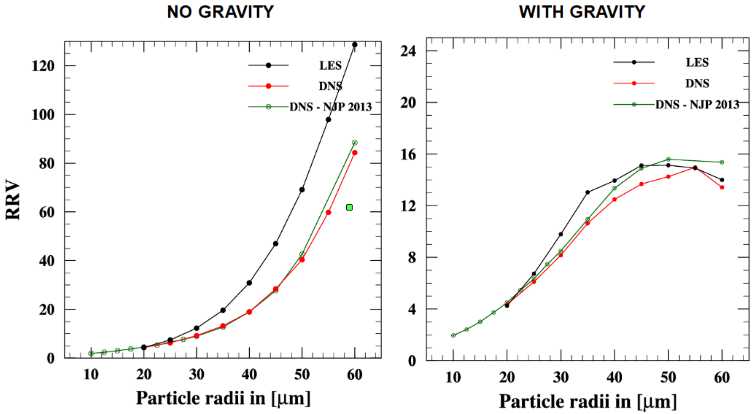


Relative Radial Velocity

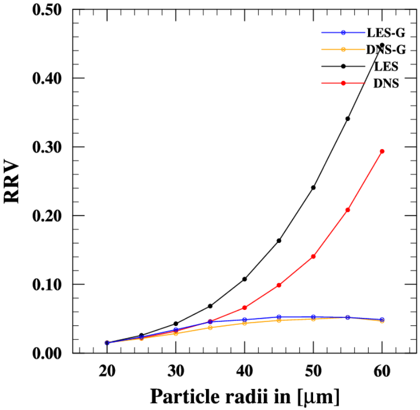
- statistic measuring relative velocities of particles, more precisely, for any particles: $w_r = \frac{\mathbf{w} \cdot \mathbf{r}}{|\mathbf{r}|}$, where: $\mathbf{w}$ – their relative velocity vector, and $\mathbf{r}$ – their separation;
- computation is similar to that for RDF – velocity bins are fitted to similar power law;
- power law for RRV (normalized by Kolmogorov velocity) is given by:

$$\frac{\langle |w_r(r)| \rangle}{v_k} = c_0(\eta/r)^{c_w(St, S_V)} ,$$

[OWC] RRV Results (1)



[OWC] RRV Results (2)



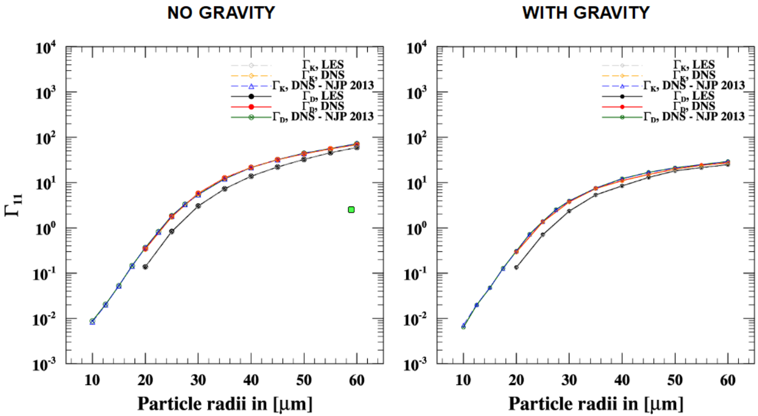
Collision Kernels

- dynamic collision kernel (Γ_{11}^{D}) is defined as the ratio of collision rate to particle pair concentration;
- it can be calculated during simulation by detecting all collision events at each time stem and then averaging it over time;
- alternatively, it was shown that collision kernel, this time referred to as kinematic (Γ_{11}^{K}), may be computed using previously obtained quantities of RDF and RRV at contact distance using following simple formula:

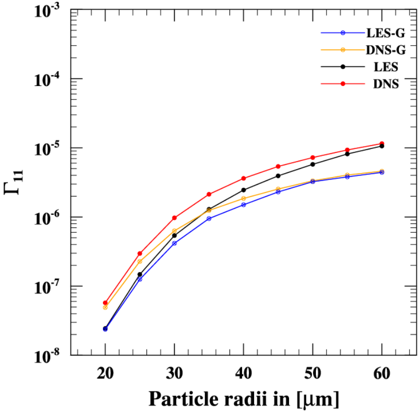
$$\Gamma_{11}^{K} = 2\pi R^2 \langle |w_r|(r = R)\rangle g_{11}(r = R)$$

- values of collision kernels computed using both methods should give same results, which is confirmed by this and previous results.

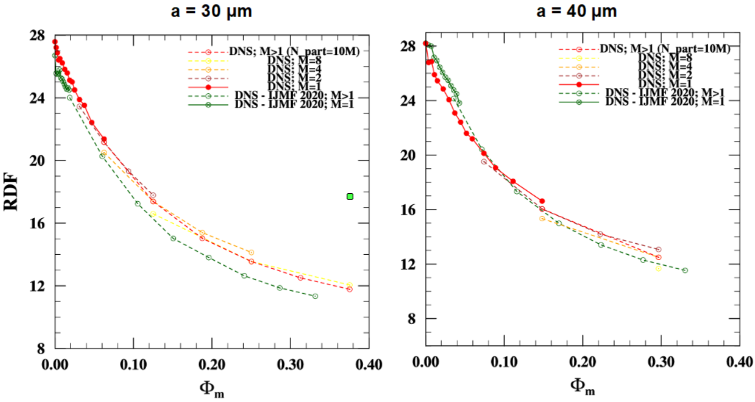
[OWC] Collision Kernel Results (1)



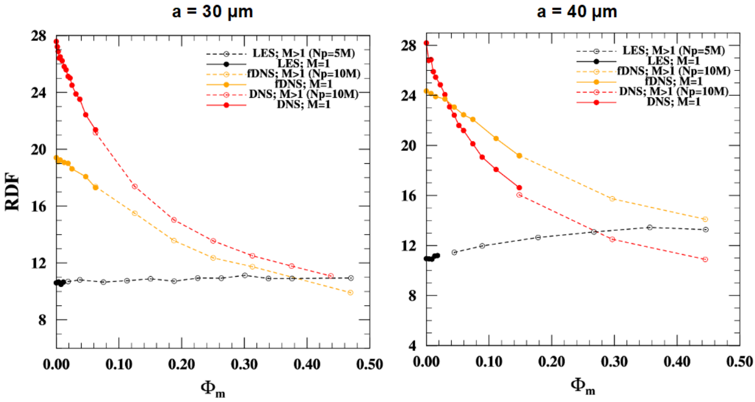
[OWC] Collision Kernel Results (2)



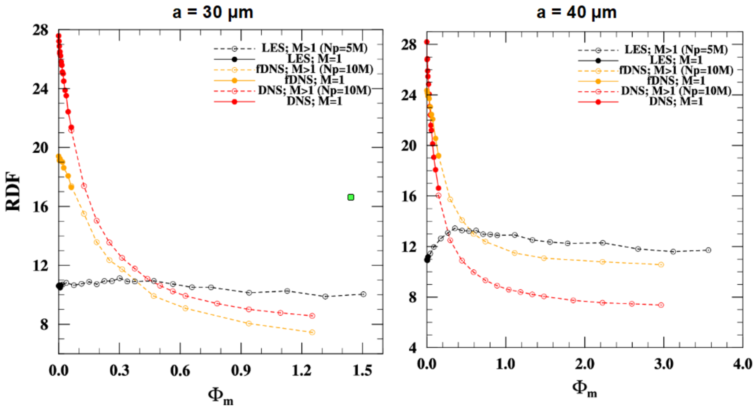
[TWC, No gravity] RDF Results (1)



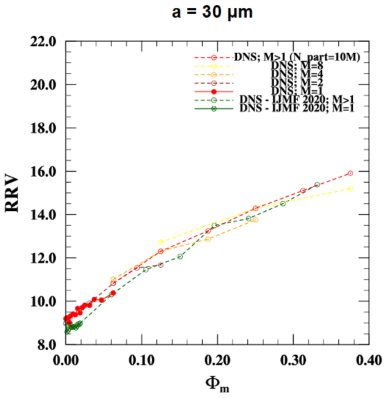
[TWC, No gravity] RDF Results (2)

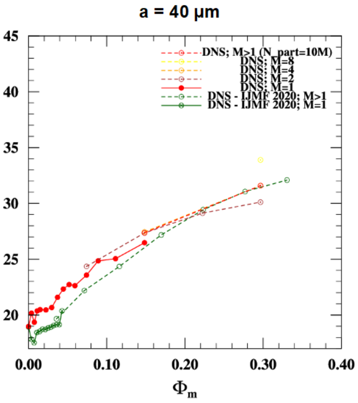


[TWC, No gravity] RDF Results (3)

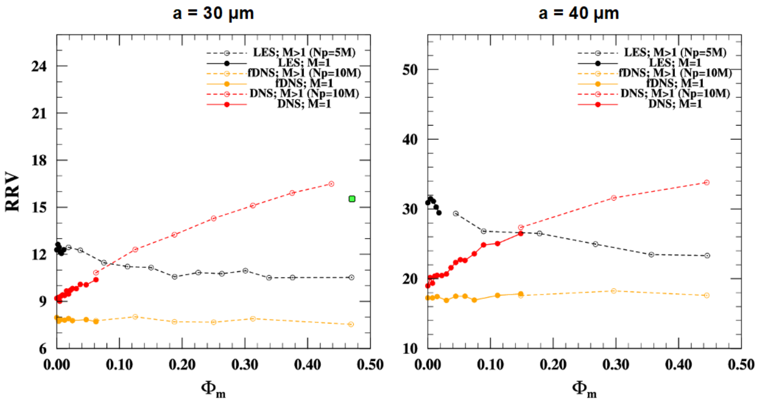


[TWC, No gravity] RRV Results (1)

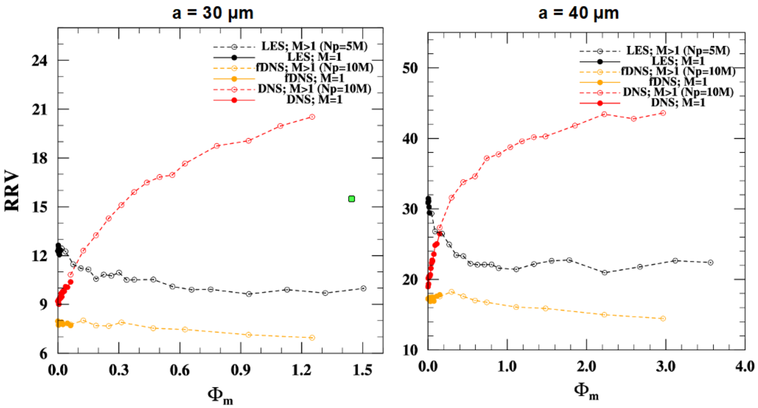




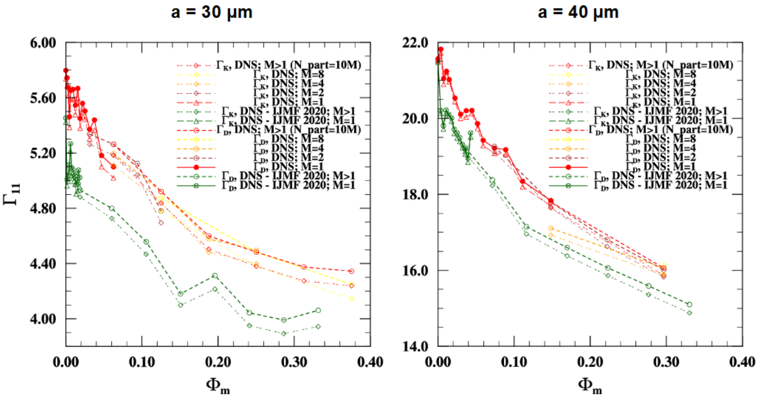
[TWC, No gravity] RRV Results (2)



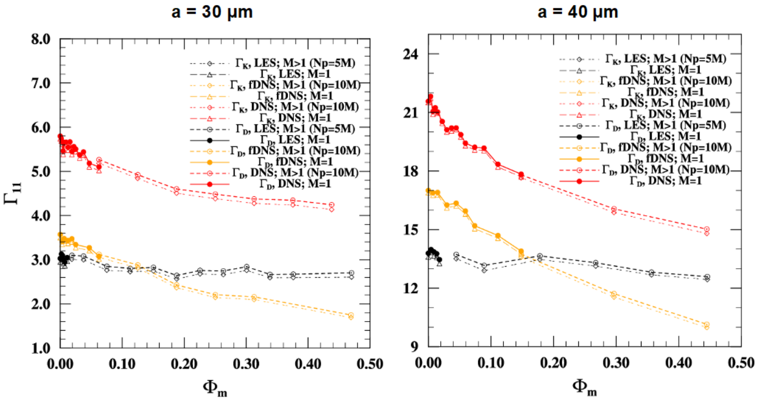
[TWC, No gravity] RRV Results (3)



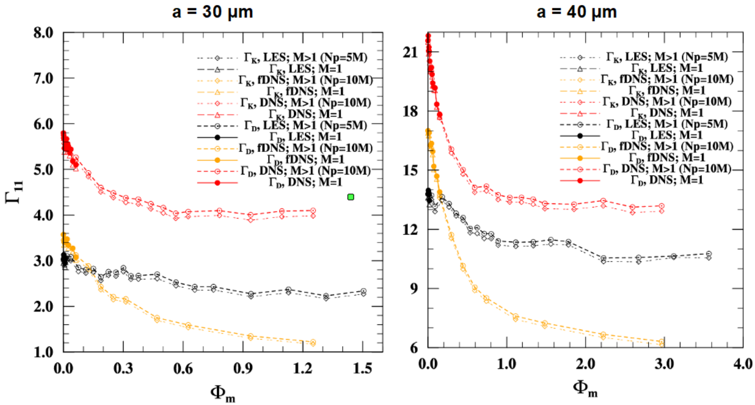
[TWC, No gravity] Collision Kernel Results (1)



[TWC, No gravity] Collision Kernel Results (2)

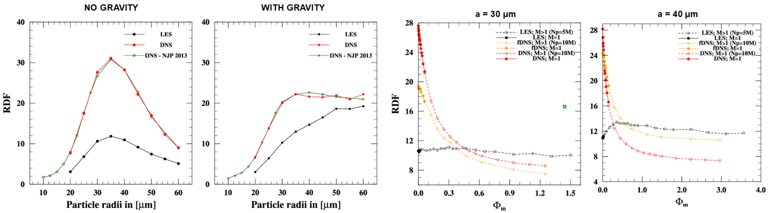


[TWC, No gravity] Collision Kernel Results (3)



Conclusions and Perspectives

DNS vs LES - Accuracy (1)



- initially promising based on OWC results:
  - with gravity - previous results giving quite good match,
  - no gravity - new results with expected deviation from DNS,
- results with TWC give limited reasons for applying LES in that scenario without any adjustments;
- awaiting for TWC results with gravity – whether they give better match;
- not likely to be effective – LES gives not only results with significant value discrepancies, but also presents different to DNS and unexpected trends when increasing mass loading.

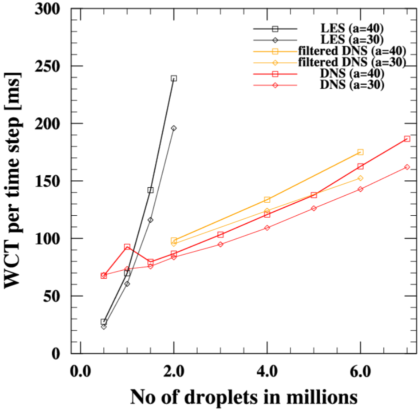
DNS vs LES - Accuracy (2)

- Perspectives:
  - wait for processed TWC results with gravity;
  - try to use different types of subgrid–scale models, or develop new ones that are meant for simulations with particles and two-way coupling (maybe together with specific interpolation scheme in TWC, i.e. alternative version of PNN, somehow coupled with subgrid–scale model).

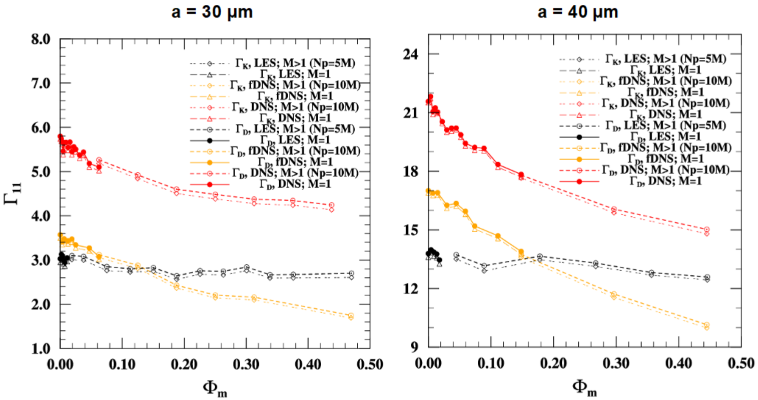
DNS vs LES - Performance (1)

- planned second part of current project / thesis – study performance increase of LES over DNS.
- problem with handling large numbers of particles with smaller number of cores (due to smaller grid, and thus smaller number of computational subdomains; here: DNS – 256; LES – 16);
- possible solution – diffferent parallelization strategy (more focused on dividing particles between more cores).

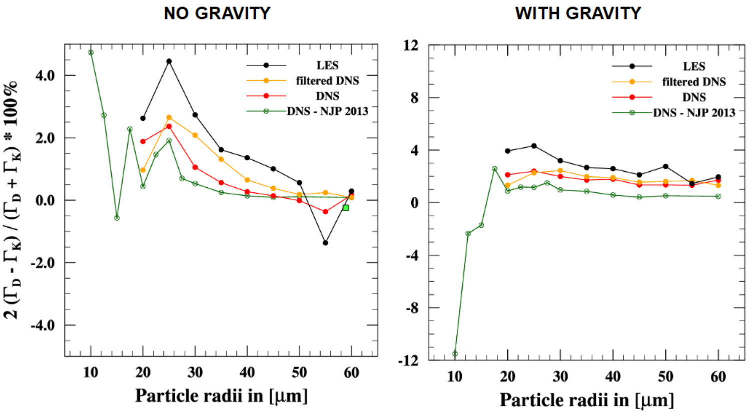
DNS vs LES - Performance (2)



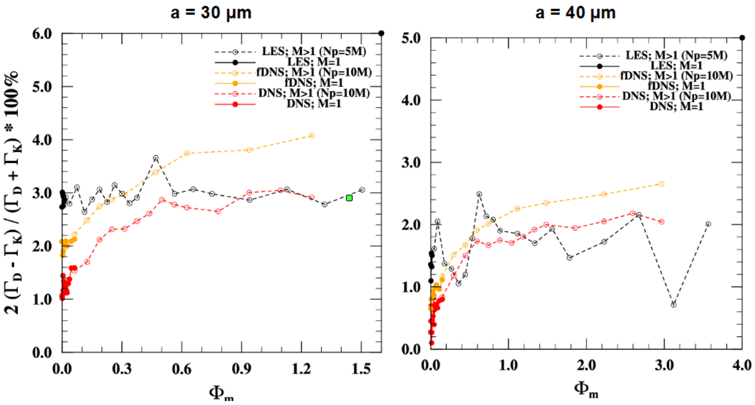
Side Issue – Collision Kernel Differences



Collision Kernel Differences [OWC]



Collision Kernel Differences [TWC, No gravity]



Collision Kernel Differences - Summary

- side issue – spotted accidentaly due to artifacts caused by different scaling in plots;
- (signed) differences between dynamic and kinematic kernels are, in case of TWC simulations, very systematic – kinematic kernels are always underestimated, and relative differences maintain stedy level depending on mass loading (less so in case of LES);
- OWC differences have similar tendencies, but are not as stable as in case of TWC;
- is such systematic difference a symptom of a systematic error that can be addressed and fixed?

References

Key references:

- B. Rosa, H. Parishani, O. Ayala, W. W. Grabowski, L.-P. Wang, **2013**. Kinematic and dynamic collision statistics of cloud droplets from high-resolution simulations, *New Journal of Physics* **15**, 045032.
- B. Rosa, J. Pozorski, **2017**. Analysis of subfilter effects on inertial particles subject to gravity in forced isotropic turbulence, *Journal of Turbulence* **18**, 634-652.
- B. Rosa, J. Pozorski, L.-P. Wang, **2020**. Effects of turbulence modulation and gravity on particle collision statistics, *International Journal of Multiphase Flow* **129**, 103334.

Thank You!

Contact: maciejmanna@gmail.com

Slides available at:

`github.com/xann16/talks/master/cfd/turb-dns-les-1`