NEPTUNE_CFD High Parallel Computing Performances for Particle-Laden Reactive Flows

H. Neau*†, J. Laviéville‡, O. Simonin*†

* Université de Toulouse ; INPT, UPS ; IMFT ; Allée Camille Soula, F-31400 Toulouse, France

† CNRS ; Institut de Mécanique des Fluides de Toulouse ; F-31400 Toulouse, France

‡ EDF Recherche & Développement ; DMFEE; F-78401, Chatou, France

neau@imft.fr

Keywords: High Performance Computing, Parallel computation, Multiphase flows, Euler multi-fluid approach

Abstract

This paper presents high performance computing of NEPTUNE_CFD V1.07@Tlse. NEPTUNE_CFD is an unstructured parallelized code (MPI) using unsteady Eulerian multi-fluid approach for dilute and dense particle-laden reactive flows. Three-dimensional numerical simulations of two test cases have been carried out. The first one, a uniform granular shear flow exhibits an excellent scalability of NEPTUNE_CFD up to 1024 cores, and demonstrates the good agreement between the parallel simulation results and the analytical solutions. Strong scaling and weak scaling benchmarks have been performed. The second test case, a realistic dense fluidized bed shows the code computing performances on an industrial geometry.

Nomenclature

Roman symbols

CPU	Central Processing Unit
Ef	Efficiency
Ef_{weak}	Weak scaling efficiency
g	Gravitational constant ($\mathrm{m.s^{-2}}$)
H_{bed}	Bed height (m)
HPC	High Performance Computing
MPI	Message Passing Interface
P_g	Mean gas pressure ($\mathrm{N.m^{-2}}$)
q_p^2	Mean particle agitation ($\mathrm{m^{-2}.s^{-2}}$)
Sp	Speedup
T_{ref}	Job execution time on one node (8 cores)
T_n	Job execution time on n nodes ($n \times 8$ cores)
$U_{k,i}$	Mean velocity of phase k ($\mathrm{m.s^{-1}}$)
$u'_{k,i}$	Fluctuating velocity of phase k ($\mathrm{m.s^{-1}}$)

Greek symbols

α_k	Volume fraction of phase k ($-$)
μ_g	Gas viscosity ($\mathrm{kg.m^{-1}.s^{-1}}$)
ρ_k	Density of phase k ($\mathrm{kg.m^{-3}}$)

Subscripts

g	Gas
p	Particle
i	i^{th} component of a vector

Introduction

Dilute and dense particle-laden reactive flows are used in a wide range of industrial applications such as coal combustion, catalytic polymerization or uranium fluoridation. A few years ago, three-dimensional numerical simulations of such flows, especially fluidized beds, were restricted to coarse meshes due to limitation of computing power (CPU clock speed, memory, hard drive). The recent improvements on high performance computing overcome this limitation. For hardware, the frequencies of the processors have remained almost flat in the last years. The development of multi-core technology and the increase of memory cache size and of memory and other interconnect frequencies have led to very highly parallel and efficient computing platforms. For software, this computational power has been exploited according to the parallelization of codes (message passing). Hence, we are able to solve previously intractable problems. The strong development of HPC allows the use of finer and larger meshes and the increase of numerical simulation accuracy.

Since a few years, three-dimensional realistic numerical simulations of industrial configurations are realized with unsteady Eulerian multi-fluid modeling approach for monodispersed or polydispersed reactive particle mixture (Randrianarivelo et al. (2007)). This approach

has been developed and implemented by IMFT (Institut de Mécanique des Fluides de Toulouse) in the NEPTUNE_CFD V1.07@Tlse version. NEPTUNE_CFD is a unstructured parallelized multiphase flow software developed in the framework of the NEPTUNE project, financially supported by CEA (Commissariat à l'Énergie Atomique), EDF (Électricité de France), IRSN (Institut de Radioprotection et de Sûreté Nucléaire) and AREVA-NP. NEPTUNE_CFD V1.07@Tlse. This code allows to take into account complex phenomena: particle mixture, particle-fluid interaction, particle-particle and particle-wall collisions, heat and mass transfers and chemical reactions.

In this study, three-dimensional numerical simulations of two test cases have been carried out: a uniform granular shear flow and a dense gas-solid fluidized bed at industrial scale. These simulations enable to evaluate high computing performances of NEPTUNE_CFD on three clusters with different technologies and architectures with different message passing libraries. Parallel structuring of the computational procedure, speedups, efficiency and their dependence upon the size of the problem, and the number of cores are discussed.

NEPTUNE_CFD overview

NEPTUNE_CFD is dedicated to calculating multiphase or multi-field flows, at the local scale and in geometries that may be complex. The software has the following main characteristics and functions:

- flow systems processed:
  - 1 to 20 fluid fields (phases or fields),
  - processing of water/steam flows with actual thermodynamic laws;
- numerical methods:
  - meshes with all types of cell (element), non-conforming connections,
  - "cell-center" type finite volume method,
  - calculation of co-localized gradients with reconstruction methods,
  - distributed-memory parallelism by domain splitting;
- physical models:
  - interfacial momentum transfer terms,
  - interfacial energy transfer terms,
  - turbulence,
  - head losses and porosity,
  - additional scalar;
- architecture:

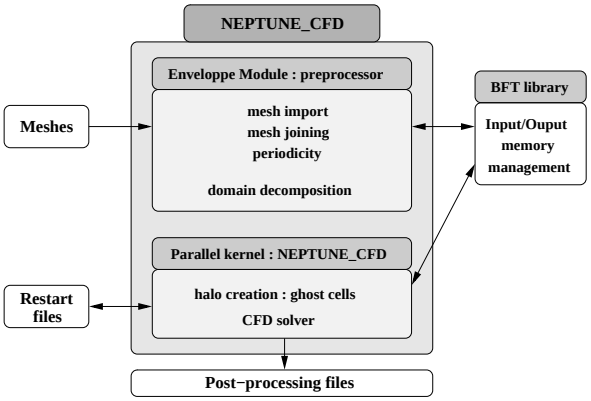


Figure 1: NEPTUNE_CFD modules.

  - interfacing with the Code_Saturne Enveloppe module for management of the preprocessing operations on the mesh, of the parallelism and of the post-processing files,
  - written in Fortran 77 (the majority) and C (ANSI 1989) (procedural programming),
  - ready for Unix and Linux platforms.

The method of parallel computing used by NEPTUNE_CFD is known as domain decomposition, in which the geometry and associated fields are broken into pieces and allocated to separate cores. The process of parallel computation involves: decomposition of mesh and fields, running the application in parallel, post-processing. The parallel running can use public or not implementations of the message passing library (MPI).

NEPTUNE_CFD is composed of two main elements (Fig. 1):

- the Kernel module is the numerical solver,
- the Enveloppe module is in charge of mesh data reading, mesh pasting allowing non matching meshes, domain decomposition for parallel computing using METIS (Karypis and Kumar (1999)), definition of periodicity boundary conditions and post-processing.

NEPTUNE_CFD also relies on one compulsory library (under LGPL license): BFT (Base Functions and Types) for memory and input/output management as well as specific utilities.

Mathematical models and numerical methods

The unsteady Eulerian multi-fluid approach is derived from joint fluid-particle PDF equation allowing to derive the transport equations for the mass, momentum and agitation of particle phases (Simonin (2000)).

In the proposed modeling approach, separate mean transport equations (mass, momentum, and fluctuating kinetic energy) are solved for each phase and coupled through inter-phase transfer terms.

In the following when subscript $k = g$ we refer to the gas and $k = p$ to the particulate phase. The mass balance equation is:

$$\frac{\partial}{\partial t}\alpha_k\rho_k + \frac{\partial}{\partial x_j}\alpha_k\rho_k U_{k,j} = 0, \quad (1)$$

where α_k is the k^{th} phase volume fraction, ρ_k the density and $U_{k,i}$ the i^{th} component of the velocity. In (1) the right-hand-side is equal to zero since no mass transfer takes place.

The mean momentum transport equation takes the following expression:

$$\alpha_k\rho_k\left[\frac{\partial U_{k,i}}{\partial t} + U_{k,j}\frac{\partial U_{k,i}}{\partial x_j}\right] = -\alpha_k\frac{\partial P_g}{\partial x_i} + \alpha_k\rho_k g_i \quad (2)$$
$$+I_{k,i} + \frac{\partial}{\partial x_j}\left[-\alpha_k\rho_k\left\langle u'_{k,i}u'_{k,j}\right\rangle + \Theta_{p,ij}\right],$$

where $u'_{k,i}$ is the fluctuating part of the instantaneous velocity of phase k, P_g is the gas pressure, g_i the i^{th} component of the gravity acceleration and $I_{k,i}$ the mean gas-particle interphase momentum transfer without the mean gas pressure contribution. Finally $\Theta_{k,ij}$ is for $k = g$ the molecular viscous tensor and for $k = p$ the collisional particle stress tensor. For details on diluted flows we refer to Simonin (1991), and on fluidized beds we refer to Balzer et al. (1995), Gobin et al. (2003).

The partial differential equations are discretized with a second-order centered scheme and the solution is time-advanced by a first order scheme.

The model and the numerical method are adapted to the handling of n-phases (in fact n-fields), including the single phase frame. Models and Eulerian transport equations represent an extension of the classical two-phase flow model developed in the frame of the ESTET-ASTRID Project and include water-steam closure laws and correlations.

The algorithm, based on original elliptic fractional step method (see Méchitoua et al. (2002) and Méchitoua et al. (2003)) that leads either to use linear solvers or direct nphas.x.nphas matrix inversion. The main interest of the method is the so-called "alpha-pressure-energy" step that ensures conservativity of mass and energy and allows strong interface source term coupling. Mass, momentum and energy equations are coupled with the help of a pressure correction equation, within the iterative "alpha-pressure-energy" step. The algorithm allows density variation according to pressure and enthalpy during the computation.

The momentum balance equations are solved with a semi-implicit method. They are splited in fractional steps: explicit balance, velocity implicit increment prediction, "alpha-pressure-energy" implicit increment prediction, final velocity correction.

The "Alpha-Pressure-Energy" step stops after the mass conservation sub-step, when the volume conservation holds. The user can adapt the criterion parameter ε_{vol}, but this one remains very severe as it applies to a maximum value over the whole domain:

$$\max_{I\in NCEL}[1 - \sum_k \alpha_k(I)] < \varepsilon_{vol}. \quad (3)$$

The standard value of ε_{vol} is 10^{-5}.

Because of implicit formulation and three-dimensional unstructured meshes, we use iterative solvers:

- for the pressure: conjugated gradient or bi-conjugate gradient stabilized (bi-cgstab),
- for volume fraction: biconjugate gradient stabilized or jacobi,
- for velocity: jacobi.

Parallel computer description

The numerical simulations have been carried out on 3 clusters:

- C1: SGI Altix ICE 8200 based on Intel Xeon Quad Core processors (Harpertown technology),
- C2: SGI Altix ICE 8200 EX based on Intel Xeon X5560 Quad Core processors (Nehalem EP technology),
- C3: cluster based on AMD Shangaï Quad Core processors.

The complete description of clusters is given in table 1. These clusters among the most efficient in France are based on different technologies, on different file systems and on different interconnects. The aim is to evaluate NEPTUNE_CFD parallel performances on different architectures about speedup and efficiency and not to compare cluster HPC (elapsed time). For clusters C1 and C2, we test 2 implementations of MPI : Intel MPI and SGI MPT (C1a, C1b, C2a and C2b) and we use Intel Fortran and C compiler. Numerical simulations on cluster C3 are carried out with GNU fortran and C compiler and OpenMPI. All clusters were in production.

Table 1: Cluster description.

Cluster	C1a	C1b	C2a	C2b	C3
Computational center	CINES		CALMIP		Renault F1 Team
Model	SGI Altix ICE 8200		SGI Altix ICE 8200 EX		HP
Number of cores	12288		2816		1024
Number of nodes	1536		352		128
Processors/nodes	2		2		2
RAM/core (GB)	4		4.5		4
Peak performance (TFlop/s)	147		31.5		9.4
Processor	Intel Xeon E5472		Intel Xeon X5560		AMD Shangaï
Processor frequency (GHz)	3.0		2.8		2.3
Cores/processor	4		4		4
L2 cache	2x(6 MB per 2 cores)		4*256KB		4*512KB
L3 cache	—		8MB per 4 cores		6MB per 4 cores
Network fabric	Infiniband		Infiniband		Infiniband
File system	Lustre		Lustre		LFS
C compiler	Intel icc11.1		Intel icc11.1		gcc4.1.2
Fortran compiler	Intel ifort11.1		Intel ifort11.1		gfortran4.1.2
MPI	Intel MPI 3.2.2	SGI MPT1.25	Intel MPI3.2.2	SGI MPT1.25	OpenMPI1.2.6

Simulation of a three-dimensional granular uniform shear flow

Test case overview

The purpose of the test case is a three-dimensional isothermal granular shear flow. For such a simple configuration, the mathematical model has a analytical solution which can be compared with the numerical solution given by NEPTUNE_CFD.

The computational domain is a cubic box of length $H_{cube} = 0.01$ m (Fig. 2). The powder and gas material properties are given in table 2. The particulate phase is monodispersed with diameter 350 μm.

The analytical solution with 0D approach gives the following values:

- particle fluctuating kinetic energy:

$$q_p^2 = 2.66.10^{-5}\ \mathrm{m}^2 \cdot \mathrm{s}^{-2},$$

- particle kinetic viscosity:

$$\mu_{k_p} = 6.21.10^{-4}\ \mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1},$$

- particle collisional viscosity:

$$\mu_{c_p} = 9.88.10^{-4}\ \mathrm{kg} \cdot \mathrm{m}^{-1} \cdot \mathrm{s}^{-1}.$$

The gas phase is laminar and the particle agitation model is accomplished by solving a system of two transport equations: one for the particle turbulent agitation and one for the fluid-particle covariance. According to a large particle to gas density ratio only the drag force is taken into account. It must be printed out that the gravity

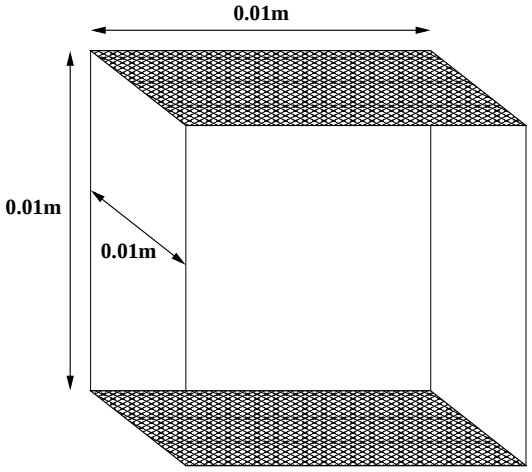


Figure 2: Sketch of the three-dimensional granular uniform shear flow.

is also neglected in this case. Finally for this test case we solve 10 equations (mass balance, momentum transport, particle turbulence).

Two different meshes have been employed to evaluate the performances: 1,000,000 cells from 1 up to 32 nodes corresponding to 8 up to 256 cores and 10,503,459 cells from 16 up to 128 nodes corresponding to 128 up to 1024 cores. The Fig. 3 shows the standard mesh composed of 1,000,000 hexaedra with uniform cell size ($\Delta x = \Delta y = \Delta z = 0.1\,\mathrm{mm}$). The second mesh is also a regular mesh of 10,503,459 hexaedra (219 cells per direction).

Configuration is detailed in Fig. 4. Three kinds of boundary conditions are used:

Table 2: Powder and gas properties.

	Gas	Particles
Density ($\mathrm{kg \cdot m^{-3}}$)	1	1,000
Viscosity$\times 10^5$ ($\mathrm{Pa \cdot s}$)	1	-
Pressure (bar)	1	-
Diameter ($\mu\mathrm{m}$)	-	350
Solid volume fraction	-	0.3
Restitution coefficient	-	0.8

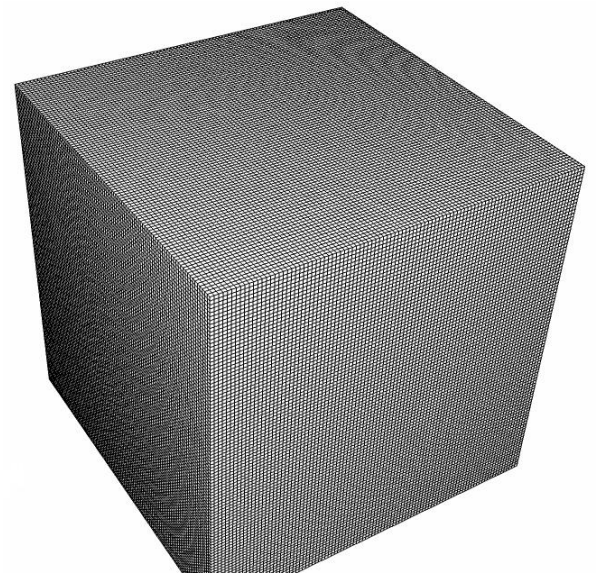


Figure 3: Mesh containing 1,000,000 hexaedra.

- right-side and left-side, inlet-outlet condition for particles and gas with imposed velocity:

$$U_x = 20 * (Y - H_{cube}/2),$$

- front-side and back-side, symmetry condition for particle and gas
- top-side and bottom-side, moving wall:

$$U_{wall} = 20 * (Y_{wall} - H_{cube}/2).$$

Moving wall is a no-slip wall boundary for particle and gas:

$$\begin{cases} U_{g,i} = U_{p,i} = U_{wall} \\ \dfrac{\partial q_p^2}{\partial n} = 0. \end{cases} \quad (4)$$

Inlet boundary condition values to particle fluctuating kinetic energy and fluid-particle fluctuating velocity covariance are:

$$\begin{cases} q_p^2 = 7.5.10^{-5}\ \mathrm{m^2 \cdot s^{-2}} \\ q_{fp} = 1.10^{-4}\ \mathrm{m^2 \cdot s^{-2}}. \end{cases}$$

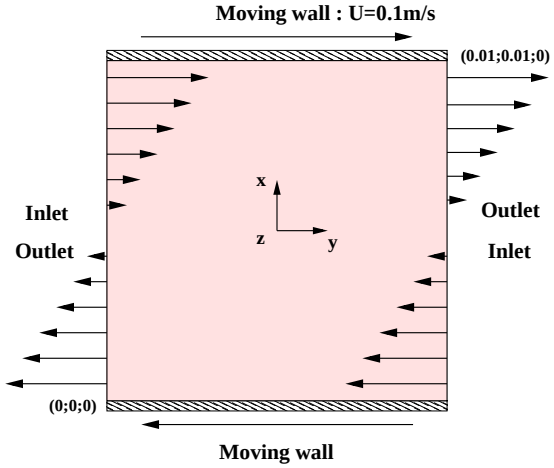


Figure 4: Schematic of boundary conditions.

Table 3: Analytical and numerical result comparison.

	Analytical	NEPTUNE_CFD	
		minimum	maximum
α_p (-)	0.30	0.29989	0.30071
q_p^2 ($\mathrm{m^2 \cdot s^{-2}}$)	$2.66.10^{-5}$	$2.6432.10^{-5}$	$2.6639.10^{-5}$
μ_{k_p} ($\mathrm{kg/ms}$)	$6.21.10^{-4}$	$6.1872.10^{-4}$	$6.2116.10^{-4}$
μ_{c_p} ($\mathrm{kg/ms}$)	$9.88.10^{-4}$	$9.8784.10^{-4}$	$9.9085.10^{-4}$

An adaptative time step is used (computed from Courant and Fourier criteria), typically $\Delta t = 5.10^{-4}$ s.

The following iterative solvers have been selected: jacobi for the velocity, conjugated gradient for the pressure and bi-cgstab for the volume fraction. The criterion parameter ε_{vol} of "Alpha-Pressure" step is fixed to 10^{-6} and the maximum number of cycles into "Alpha-Pressure" step is 50.

Results and discussion

After a transient phase, numerical simulations converge to stationary state. The Fig. 5 shows temporal evolution of the solid volume fraction between 0.5 s and 2.0 s.

Three-dimensional unsteady numerical simulation results agree with analytical solutions (table 3). The accuracy of numerical results is independent of parallelization and of core number. NEPTUNE_CFD performances are evaluated on a restart simulation of 1,000 additional iterations after transient step. The "CPU_TIME" intrinsic fortran sub-routine has been used to measure CPU times per core of the main routines of code, the CPU times per iteration per core and the effective CPU time per core of the simulation. The measurements of different times are repeated 3 times for

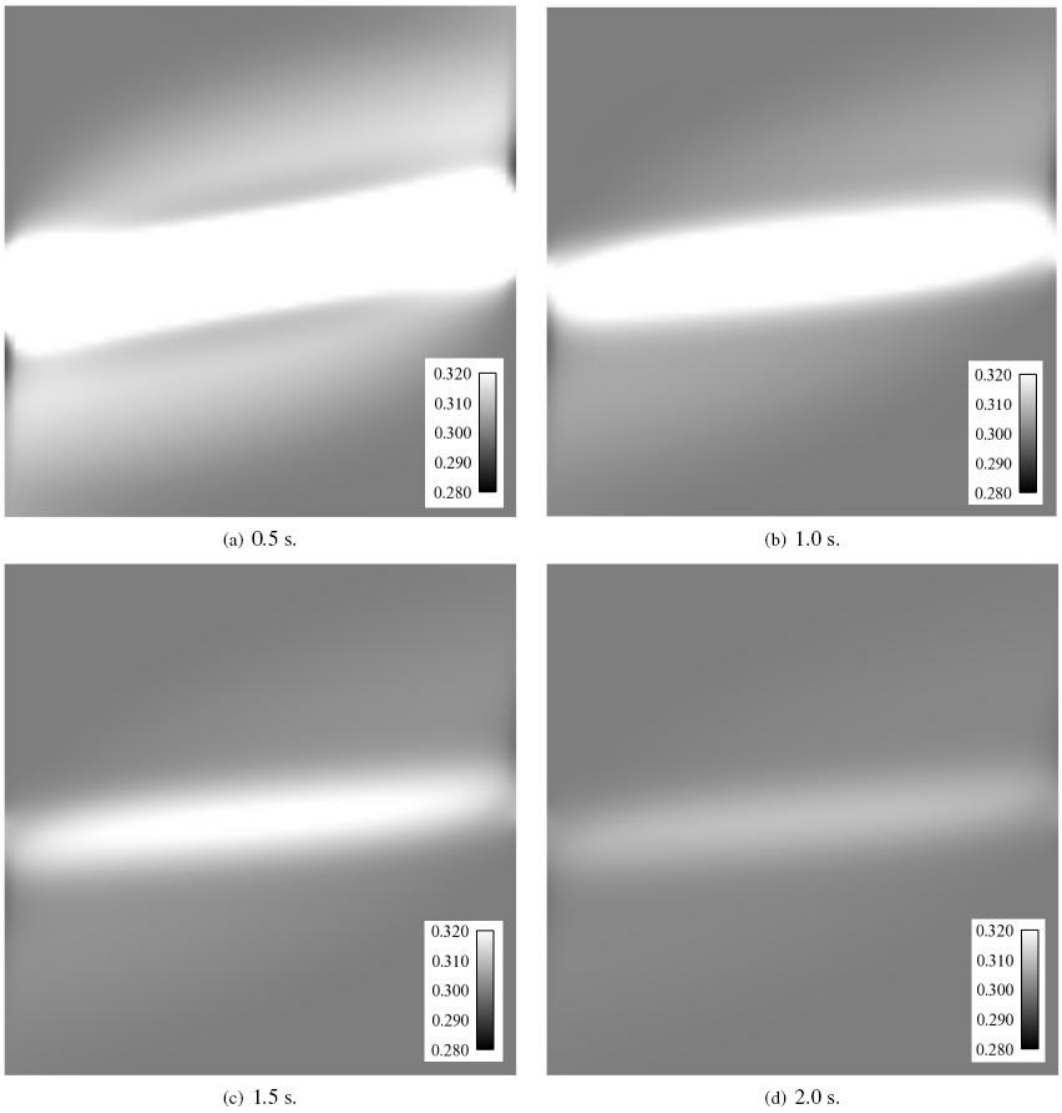


Figure 5: Temporal convergence of solid volume fraction between 0.5 s to 2.0 s.

each case to remove accidental and statistical error. No significant difference of measured time is observed. The following analysis is based on the averaged values of the 3 different runs.

In a first time, we focus on the strong scaling which is defined as how the solution time varies with the number of cores for a fixed total problem size. Secondly, we are interested in the weak scaling that is to say how the solution time varies with the number of processors for a fixed problem size per processor (strong scaling is about same problem size in shorter time and weak scaling is about larger problem size in same time). It is hard to achieve a good strong-scaling at larger process counts since the communication overhead increases in proportion to the number of processes used.

Table 4: Effect of core number on effective core CPU time for 1,000 iterations with the standard mesh.

number of cores	Effective CPU time/core (s.)				
	C1a	C1b	C2a	C2b	C3
8	35,621	35,325	12,150	12,462	15,469
16	14,999	14,936	5,628	5,639	7,087
32	5,959	5,877	2,598	2,567	3,254
64	1,571	1,323	1,200	1,127	1,548
128	803	751	563	583	895
256	580	757	449	485	1,269
512	498	—	493	—	—

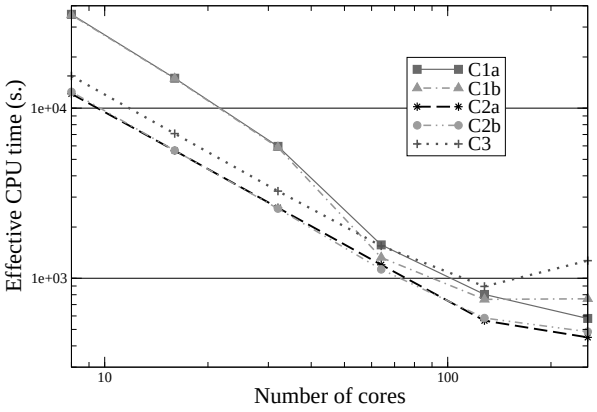


Figure 6: NEPTUNE_CFD effective CPU time per core (standard mesh - 1,000 iterations).

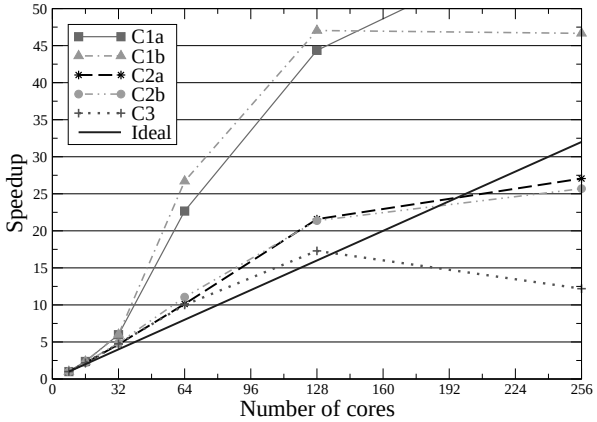


Figure 7: NEPTUNE_CFD speed-up with respect to T_{ref} of each cluster (standard mesh - 1,000 iterations).

The table 4 shows the effect of core number on effective CPU time per core with the standard mesh. Computation elapsed time is near to this time. The speedup is defined as the ratio between the elapsed time to execute a program on one node (ref = 1 node = 8 cores) and on a set of concurrent n nodes ($n \times 8$ cores) and the efficiency is defined as the ratio between speedup and n:

$$Sp = \frac{T_{ref}}{T_n} = \frac{T_{8cores}}{T_n}$$
$$Ef = \frac{Sp}{n} \quad (5)$$

where T_{ref} is the effective core CPU time of the parallel job on one node (8 cores) and T_n is the effective core CPU time of the parallel job on n nodes ($n \times 8$ cores).

NEPTUNE_CFD effective CPU time per core, speedup and efficiency of the calculation are depicted in Fig. 6, 7 and 8 respectively. The continuous line without symbol indicates linear speedup. The Fig. 7

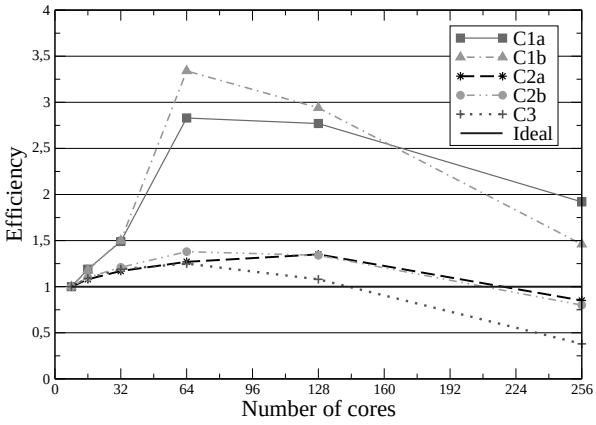


Figure 8: NEPTUNE_CFD efficiency with respect to T_{ref} of each cluster (standard mesh - 1,000 iterations).

demonstrates that NEPTUNE_CFD exploits the parallelism very efficiently. From 16 up to 128 cores (2 up to 16 nodes), all of the speedup is greater or equal to the linear speedup (say, super-linear speedup) and the efficiencies are also greater or equal to 1.0 (Fig. 8).

However, in Fig. 6, we can observe very long effective CPU time on C1a and C1b for weak core number (up to 32). Ratio between C1 and C2 CPU time is about 1.3 for high core number and about 3 for weak core number. This effect may be attributed to Harpertown technology and cache limitation. The overtime for effective CPU time per core using 8 cores implies speedup and efficiency over-estimate. Using 128 cores, a speedup of 45 is not realistic, linear speedup is 16.

To avoid this problem, we define "absolute" speedup and efficiency with respect to T_{ref} of C2a (ref = 1 node = 8 cores) as following:

$$Sp_{abs} = \frac{T_{ref_{C2a}}}{T_n} = \frac{T_{8cores_{C2a}}}{T_n}$$
$$Ef_{abs} = \frac{Sp}{n} \quad (6)$$

where T_{ref} is the execution time of the parallel program on one node (8 cores) on C2a and T_n is the elapsed time of the parallel program on n nodes ($n \times 8$ cores) on the different clusters.

When we use this kind of definition about speedup, the speedup values are less interesting than the evolution of speedup per cluster as core number.The changing slope indicates the optimum core number. The Fig. 9 and 10 show the evolution of absolute speedup and efficiency. On all clusters, parallel performances are very good from 8 up to 128 cores. On C1, we find again low performances for few cores. Over 128 cores, the efficiency degradation is significant especially for C3 where effective CPU time increases. Moreover for C1 and C2,

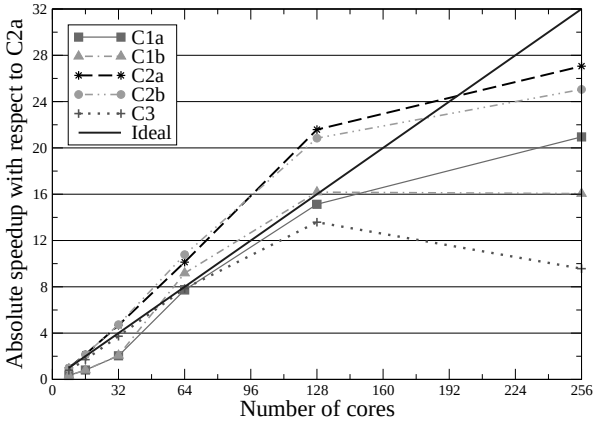


Figure 9: NEPTUNE_CFD absolute speed-up with respect to T_{ref} of C2a (standard mesh - 1,000 iterations).

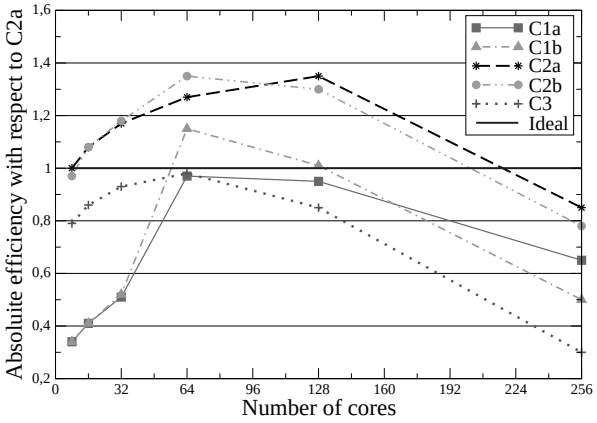


Figure 10: NEPTUNE_CFD absolute efficiency with respect to T_{ref} of C2a (standard mesh - 1,000 iterations).

we can note a MPI implementation effect. SGI MPT seems more efficient for several cores and Intel MPI is better for numerous cores.

The degradation for high core number is mainly caused by the load unbalancing and the communication overhead. Restitution time of a simulation is divided into computation time, communication and waiting time (send/receive/barrier MPI) and I/O time. In this case, I/O time is almost zero. The increase of core number corresponds to a decrease of cell number per core that is to say to a decrease of computation time per core. Moreover, more cores implies more communications between cores. When communication time increases quicker than calculation time decreases, the core limit number is reached. Thus there is an optimal number of cores for a mesh and a set of equations.

For this numerical simulation of a three-dimensional granular uniform shear flow solving 10 equations, NEPTUNE_CFD performances are excellent on all clusters

Table 5: Effect of core number on core CPU time for 500 iterations with the refined mesh on C2a.

number of cores	CPU time/core (s.)
8	109,220
16	55,205
32	28,877
64	14,707
128	7,351
256	3,721
512	1,780
768	1,220
1024	962

while cell number per core is greater than 10,000 that is to say while the core number is lower or equal to 128 for a mesh of 1,000,000 cells. On Nehalem cluster, performances are excellent up to 196 cores. It is noteworthy that the restitution time decreases up to 256 cores (5,000 nodes per core).

NEPTUNE_CFD scalability is excellent on all platforms, with linear speedup on Harpertown and Shangaï technologies and super-linear speedup on Nehalem technology.

To improve NEPTUNE_CFD performance evaluation, we perform numerical simulations with the refined mesh (10,503,459 hexaedra) using from 1 up to 128 nodes (8 up to 1024 cores). As CPU times are important, these numerical simulations have been realized only on cluster C2a (the most recent cluster) with Intel MPI implementation. Performances are evaluated on a restart simulation of 500 additional iterations after transient phase. Time measurements are still done with the "CPU_TIME" intrinsic fortran sub-routine and each run is repeated 3 times.

The table 5 shows effect of core number on effective CPU time per core with the refined mesh. The speedup and the efficiency are defined as previously.

NEPTUNE_CFD effective CPU time per core, speedup and efficiency of the calculation are depicted in Fig. 11, 12 and 13 respectively. The solid line represents the linear speedup. The Fig. 12 demonstrates that NEPTUNE_CFD exploits the parallelism very efficiently even for large core number. The speedup is super-linear up to 768 cores and linear for 1,024 cores and efficiency are also greater or equal to 1.0 (Fig. 13). Parallel performances are very good up to 128 nodes. As in previous case, performance degradation begins when communications become preponderant to compute. We can assume that with this mesh speedup should be low using 2,048 cores .

This refined mesh study confirms that NEPTUNE_CFD performances are excellent on Nehalem

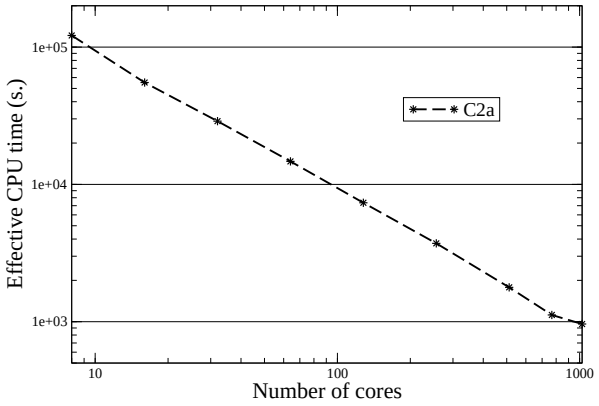


Figure 11: NEPTUNE_CFD effective CPU time per core (refined mesh - 500 iterations).

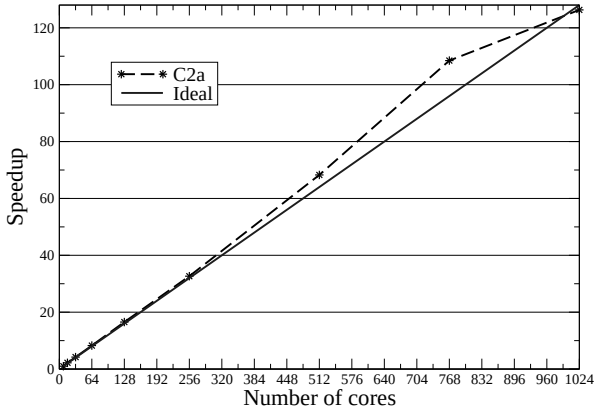


Figure 12: NEPTUNE_CFD speed-up with ref = 1 node (refined mesh - 500 iterations).

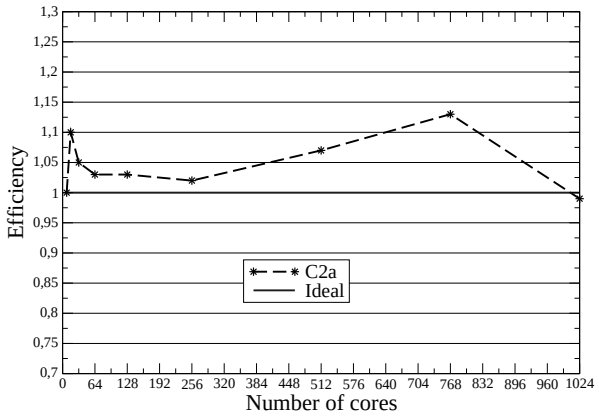


Figure 13: NEPTUNE_CFD efficiency with ref = 1 node (refined mesh - 500 iterations).

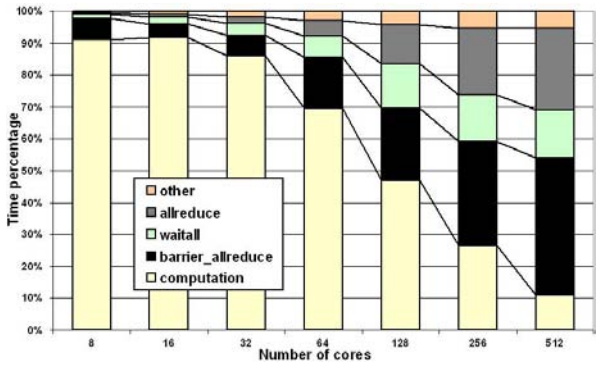


Figure 14: MPINSIDE profiling on C2a (standard mesh - 1,000 iterations).

cluster (C2) while cell number per core is higher than 10,000 that is to say for a mesh of 10,503,459 cells a core number lower or equal to 1,024.

Hence, NEPTUNE_CFD strong scaling is excellent up to 1,024 cores in condition of respect of minimum cell number per core.

To understand these performances and time divide according to core number, NEPTUNE_CFD profiling is necessary. We use MPINSIDE to evaluate MPI communications and PerfSuite to get NEPTUNE_CFD timing informations.

MPINSIDE increases slightly the elapsed time of numerical simulations but gives the right percentages of elapsed time used by computation and by the different MPI process. The main MPI function times and computational time are depicted in Fig. 14 and 15 for the standard and the refined meshes for different core number.

Using 8 cores, with the two meshes, computation time is over 90 % of effective core CPU time. Barrier before MPI Allreduce time (b_allred) is weak and MPI Allreduce time (allred) and Waitall time are almost nil. Other MPI job times (overhead, send, receive, ...) are negligible. Increasing core number, barrier before MPI Allreduce time, MPI Allreduce time and Waitall time increase and computational time decrease. Using 64 cores with the standard mesh, computational time is only about 70 % of core CPU time and using 512 cores computational time is about 10 %. With the refined mesh, using 256 cores, computational time is only about 70 % of core CPU time and using 1024 cores computational time is about 25 %.

MPINSIDE gives informations about MPI function samples. Using 16 cores on refined mesh, during 500 iterations, barrier before MPI Allreduce number is about 55,904,240. Using 1,024 cores, barrier before MPI Allreduce number is about 3,574,704,128. Indeed, NEPTUNE_CFD iterative solvers require a lot of MPI communications. Thus, a lot of messages are exchanged between the cores many times in every iteration. For a high

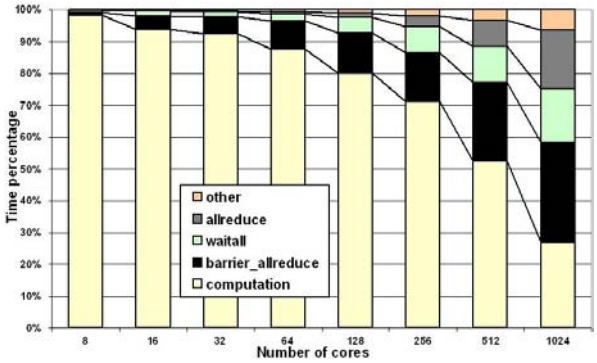


Figure 15: MPINSIDE profiling on C2a (refined mesh - 500 iterations).

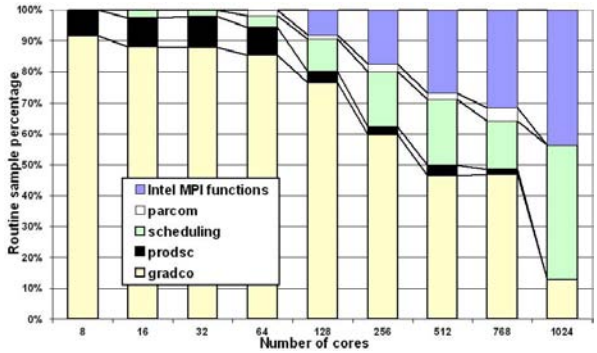


Figure 16: PerfSuite routine profiling on C2a (refined mesh - 500 iterations).

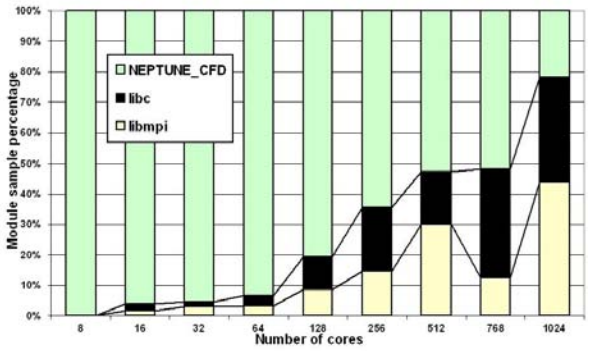


Figure 17: PerfSuite module profiling on C2a (refined mesh - 500 iterations).

Table 6: NEPTUNE_CFD integrated profiling : details of one iteration with the refined mesh using 16 cores on C2a.

step	CPU time (s.)
post-processing	0.00
initialization	1.35
velocity prediction	5.41
"alpha-pressure"	109.70
gradco routine	102.92
jacobi routine	1.61
bi-cgstab routine	1.19
Total time for iteration	120

core number, communication time is important. MPINSIDE confirms that MPI communications due to iterative solvers and "alpha pressure" step will be the main limitations to NEPTUNE_CFD massively parallel computation.

PerfSuite is used to know the most frequently asked routines and functions (sample profiling). The Fig. 16 shows the distribution of call for the main "routines" as a function of core number. The term "routines" here corresponds to Intel MPI functions, function "parcom" (call of MPI function in fortran code), scheduling, routine "prodsc" (fortran scalar product), routine "gradco". For low core number, percentage of NEPTUNE_CFD routine sample for "gradco" and "prodsc" is about 90 %. For high core number (above 256), percentage of Intel MPI functions and scheduling is significant and become predominant for 1,024 cores.

PerfSuite enables to follow the called number to the code, the “libc” library and the Intel MPI library (Fig. 17). The increase of core number reduces strongly the percentage the NEPTUNE_CFD call number.

This sample profiling can be compared with a time profiling included in NEPTUNE_CFD. This profiling gives per iteration per core, CPU time and main algorithm step time. MPI process time are included into routines calling. In table 6, we can see that with the refined mesh and using 16 cores, one iteration requires a core CPU time of 120 s. meanly spent by the "alpha-pressure" step (109 s) and this one is meanly spent by the conjugate gradient (103 s). For the same iteration but this time using 1,024 cores, the core CPU time of 2.14 s. is meanly spent by the "alpha-pressure" step (1.88 s) which is meanly spent by conjugate gradient (1.76 s).

Between 80 % and 90 % of effective CPU time is spent in the step of matrix-inversion (iterative solvers).

A complementary aspect of performance evaluation is weak scaling. To respect a cell number per core of 20,000 we have created seven regular meshes to evaluate performance from 16 cores up to 512. The table 7 details the different meshes according to the core number. NEPTUNE_CFD performances are evaluated on a restart simulation of 2,500 additional iterations after transient step on C2a.

In the Fig. 18, the weak scaling of NEPTUNE_CFD is shown. The effective CPU times per core are almost constant. A weak scaling efficiency Ef_{weak} can be de-

Table 7: NEPTUNE_CFD weak scaling : different meshes according to core number on C2a.

hexaedra number	Core number	Cell per core
166,375	8	20,797
328,509	16	20,532
658,503	32	20,578
1,295,029	64	20,235
2,515,456	128	19,652
5,177,717	256	20,225
10,503,459	512	20,515

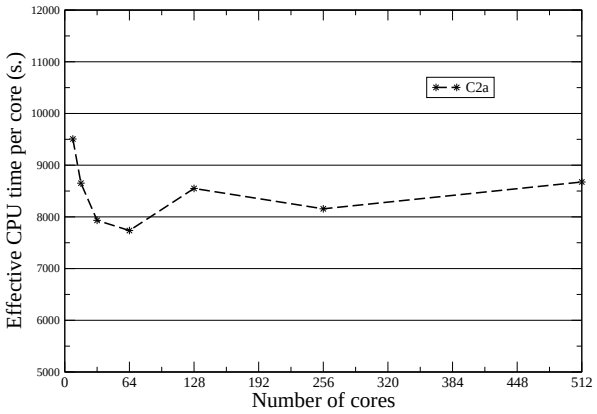


Figure 18: NEPTUNE_CFD effective CPU time per core for 2,500 iterations (20,000 cells per core)

fined as following:

$$Ef_{weak} = \frac{T_{ref}}{T_n} = \frac{T_{8cores}}{T_n} \quad (7)$$

(8)

where T_{ref} is the elapsed time of the parallel job on one node with smallest mesh and T_n is the elapsed time of the parallel job on n nodes with the respective meshes. The Fig. 19 shows that this weak scaling efficiency is very good up to 512 cores (larger than 1).

This test case of granular shear flow exhibits accuracy of NEPTUNE_CFD results and excellent scalability of the code for large enough problem sizes.

Simulation of a industrial gas-solid fluidized bed

The purpose of this test case is a three-dimensional instationnary dense fluidized bed at industrial scale. NEPTUNE_CFD V1.07@Tlse is dedicated to this kind of numerical simulations Gobin et al. (2003).

The industrial reactor is composed of a cylinder and a bulb at the top. The reactor is about 30 meter high and 5

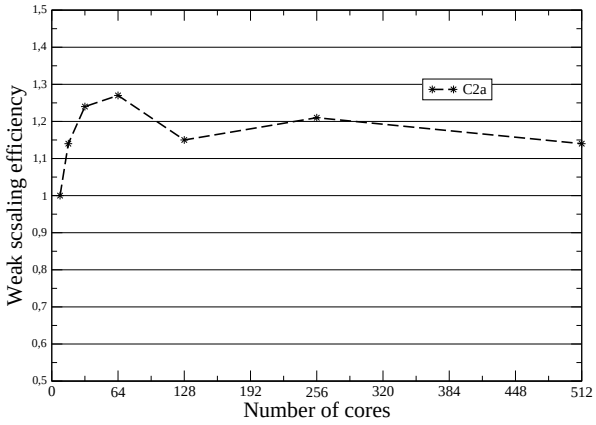


Figure 19: NEPTUNE_CFD weak scaling efficiency for 2,500 iterations (20,000 cells per core)

Table 8: Powder and gas properties of industrial reactor.

	Gas	Particles
Density ($\mathrm{kg/m^3}$)	21.11	850
Viscosity$\times 10^5$ (Pa.s)	1.54	-
Pressure (bar)	20	-
$\mathrm{V_f}$ ($\mathrm{m.s^{-1}}$)	0.67	-
Mean diameter (μm)	-	1,600
Solid mass (kg)	-	80,000
Restitution coefficient	-	0.9

meter wide as shown by Fig. 21. This bulb is an hemispherical dome with two smokes-tacks for gas outlet.

The powder properties and operating points are given in table 8. In the industrial process, the particle phase is polydispersed however the numerical simulations have been carried out with monodisperse particle having a median diameter.

For the gas turbulence modeling, we use a standard $k-\varepsilon$ model extended to the multiphase flows accounting for additional source terms due to the interfacial interactions. For the dispersed phase, a coupled transport equations system is solved on particle fluctuating kinetic energy and fluid-particle fluctuating velocity covariance. According to large particle to gas density ratio only the drag force is accounted as acting on the particles. Finally, we solve 12 equations (mass balance, momentum transport, gas and particle turbulence).

The three-dimensional mesh is shown by Fig. 20. The mesh, based on O-grid technique, contains 3,150,716 hexaedra with approximately $\Delta x = \Delta y = 30\ \mathrm{mm}$ and $\Delta z = 90\ \mathrm{mm}$.

At the bottom, the fluidization grid is an inlet for the gas with imposed superficial velocity corresponding to the fluidization velocity $V_f = 0.67\ \mathrm{m.s^{-1}}$. For the particles this section is a wall. At the top of the fluidized

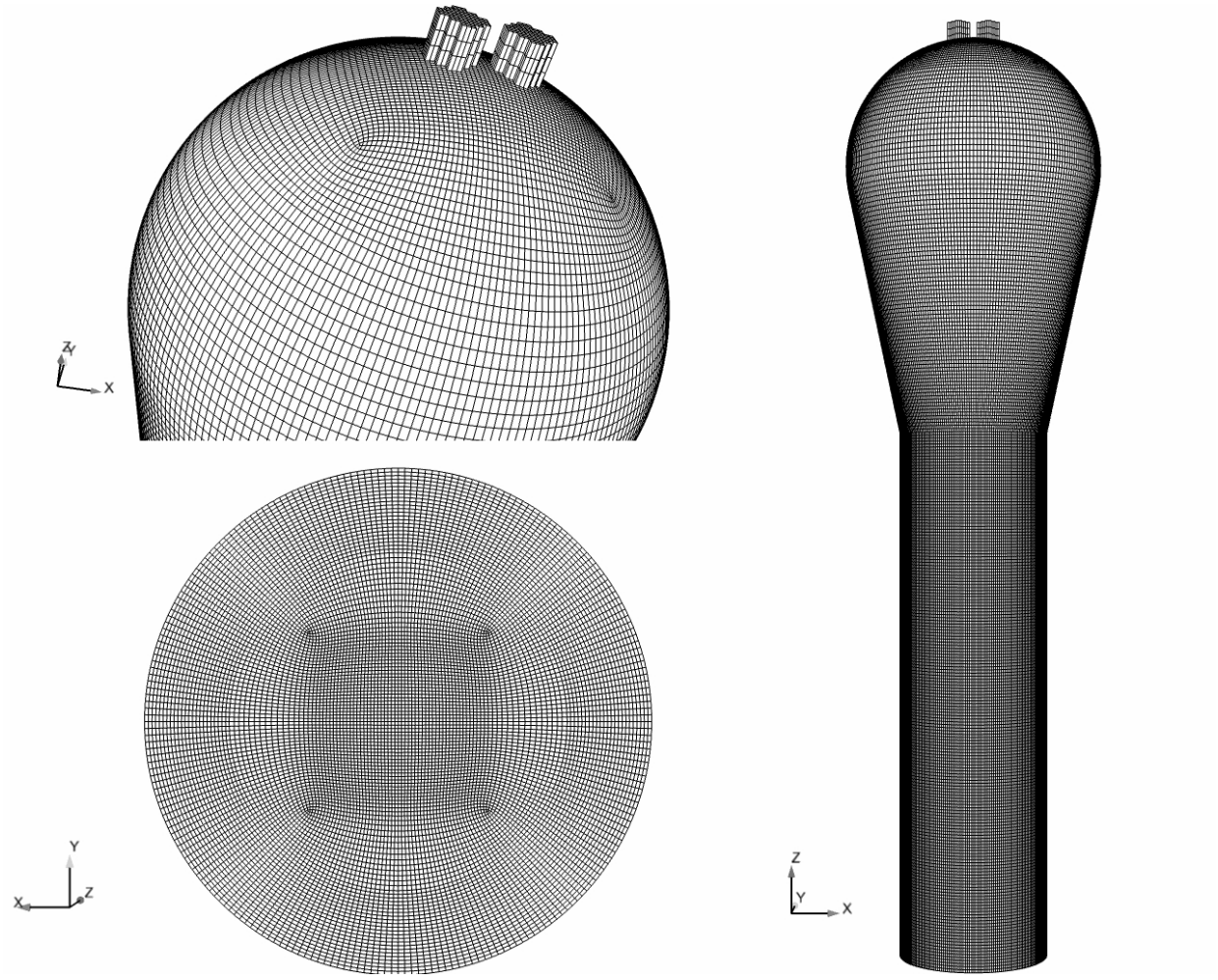


Figure 20: Three-dimensional mesh containing 3,150,716 hexaedra.

bed reactor, we defined two free outlet for both the gas and the particles. The wall-type boundary condition is friction for the gas and slip for the particles.

$$\begin{cases} U_{p,n} = 0 \\ \dfrac{\partial U_{p,\tau}}{\partial n} = 0 \\ \dfrac{\partial q_p^2}{\partial n} = 0, \end{cases} \tag{9}$$

An adaptative time step is used (computed from Courant and Fourier criteria), typically $\Delta t = 1.10^{-3}$s.

The following iterative solvers have been selected: jacobi for the velocity, conjugated gradient for the pressure and bi-cgstab for the volume fraction. The criterion parameter ε_{vol} of "Alpha-Pressure" step is fixed to is 10^{-6} and the maximum number of cycles into "Alpha-Pressure" step is 50.

Results and discussion

The Fig. 22(a) shows the automatic domain decomposition realized by METIS on the industrial reactor.

The Fig. 22(b) shows instantaneous solid volume fraction at 16.5 s.

NEPTUNE_CFD performances are evaluated on a restart simulation of 250 additional iterations after a transient phase corresponding to the destabilization of the fluidized bed (15 s.). Numerical simulations have been carried out only on cluster C2a (Nehalem architecture with Intel MPI).

As for the shear flow, the "CPU_TIME" intrinsic fortran sub-routine has been used to measure the different times per core.

The table 9 shows the effect of core number on effective CPU time per core.

As described above, we calculate speedup and efficiency. NEPTUNE_CFD effective CPU time per core, speedup and efficiency of this calculation are depicted

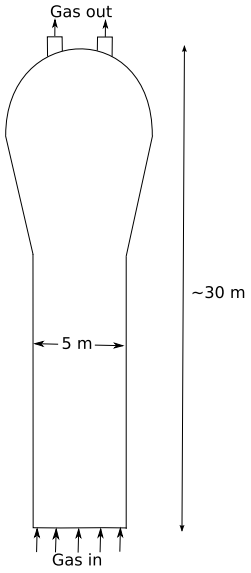


Figure 21: Sketch of the industrial polymerization reactor.

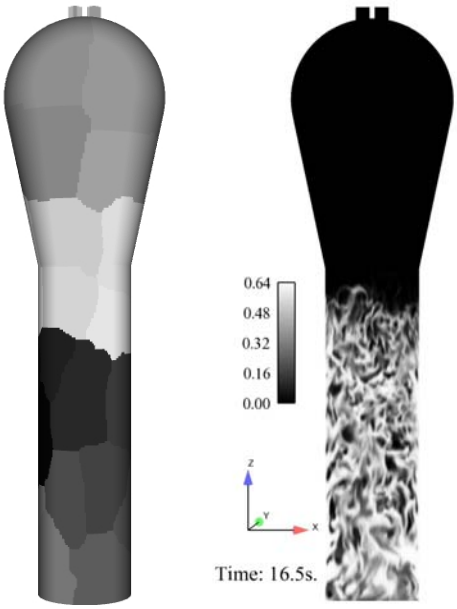
Table 9: Effect of core number on core CPU time on C2a (industrial fluidized bed).

number of cores	CPU time/core (s.)
8	68,479
16	40,885
32	19,233
64	8,625
128	4,129
256	1,707
384	1,275
512	1,173
768	1,190
1024	1,220

in Fig. 23, 24 and 25 respectively. The continuous line without symbol indicates linear speedup and ideal efficiency. The Fig. 24 demonstrates that NEPTUNE_CFD exploits the parallelism very efficiently. From 16 up to 384 cores (48 nodes), the speedup is super-linear and efficiencies are also greater than 1.0 (Fig. 25).

Parallel performances are very good up to 384 cores, the restitution time decreases up to 512 cores. Over this value, an increase of core number is not interesting but restitution time does not increase significantly. The increase of communication time is balanced by the decrease of computational time.

On the same way, in this case realistic industrial dense fluidized bed solving 12 equations, NEPTUNE_CFD HPC is excellent on Nehalem cluster (C2) while cell



(a) (b)

Figure 22: Industrial fluidized bed. (a) Automatic domain decomposition using Metis. (b) Instantaneous solid volume fraction field.

number per core is higher than 10,000 that is to say for a mesh of 3,150,716 hexaedra a core number inferior or equal to 384. If we consider the "cost" of the numerical simulation about CPU time, the optimum performances are obtained with 256 cores.

This test case illustrates excellent NEPTUNE_CFD scalability performances on a industrial geometry. Strong scaling is good for large enough problem sizes (up to 384 cores with this mesh).

Conclusion

Three-dimensional numerical simulations of two different cases of dense gas-particle flows have been performed to evaluate NEPTUNE_CFD parallel performances. The first one exhibits accuracy of numerical results and demonstrates high NEPTUNE_CFD parallel computing performances up to 1,024 cores. The second one, a realistic fluidized bed shows NEPTUNE_CFD efficiency and scalability on industrial geometry. The results show linear speed-up using Harpertown and Shangaï technologies and super-linear speedup with Nehalem technology. For the two test cases, a minimum of 10,000 cells of mesh per core is needed and 20,000 is recommended. A next step of evaluation of NEPTUNE_CFD performances will focus on effect of phase number (fluids) and of solved equation number on the minimum cell number per core. On the other, we will

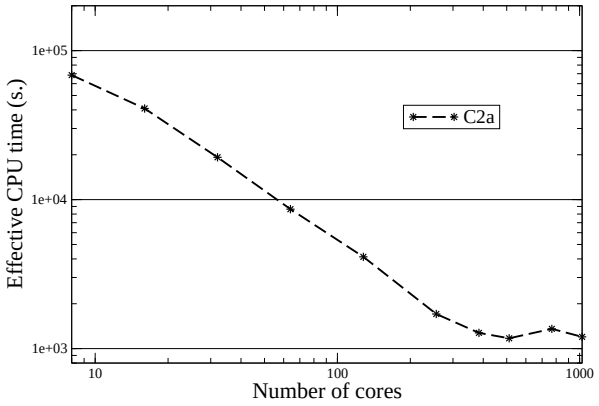


Figure 23: NEPTUNE_CFD effective CPU time per core on C2a (250 iterations) on an industrial reactor.

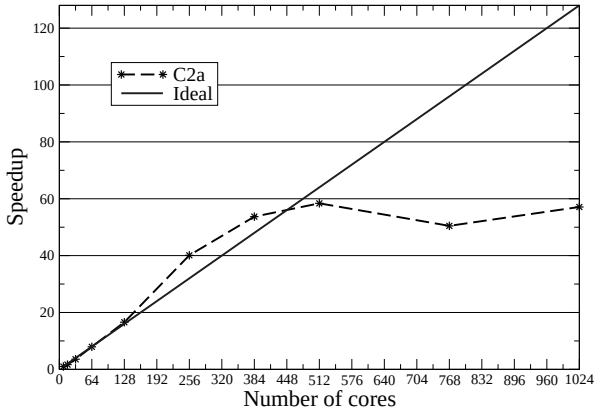


Figure 24: NEPTUNE_CFD speed-up with ref =1 node on C2a (250 iterations) on an industrial reactor.

evaluate the performances using several thousands of cores such as realized on the Open Source CFD code: Code_SATURNE by Sunderland et al. (2008).

Moreover, the profilings have shown that the most important part of CPU time is spent in the step of matrix-inversion by the iterative approach. The next implementation of multi-grid in NEPTUNE_CFD should improve strongly the global performances of the code.

Acknowledgments

This work was granted access to the HPC resources of CALMIP under the allocation P0111 (Calcul en Midi-Pyrénées). This work was also granted access to the HPC resources of CINES under the allocation 2010-026012 made by GENCI (Grand Equipement National de Calcul Intensif).

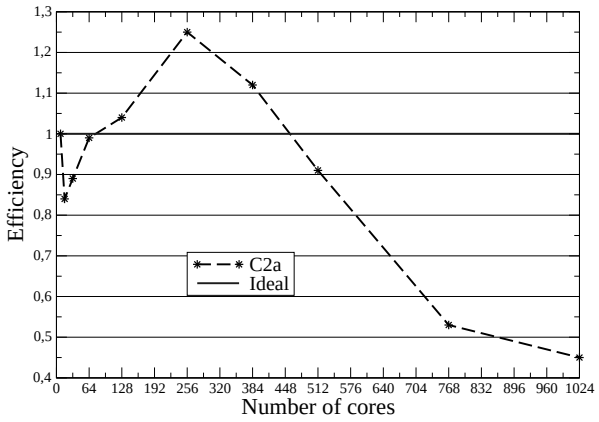


Figure 25: NEPTUNE_CFD efficiency with ref = 1 node on C2a (250 iterations) on an industrial reactor

References

G. Balzer, A. Boelle, and O. Simonin. Eulerian gas-solid flow modelling of dense fluidized bed. *FLUIDIZATION VIII, Proc. International Symposium of the Engineering Foundation, J.F. Large and C. Laguérie (Editors)*, pages 409–418, 1995.

A. Gobin, H. Neau, O. Simonin, J.R. Llinas, V. Reiling, and J.L Sélo. Fluid dynamic numerical simulation of a gas phase polymerization reactor. *Int. J. for Num. Methods in Fluids, 43*, pages 1199–1220, 2003.

G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing, Vol. 20, No 1*, pages 359–392, 1999.

N. Méchitoua, M. Boucker, S. Mimouni, S. Pigny, and G. Serre. Numerical simulation of multiphase flow with on elliptic oriented fractional step method. *Third International Symposium on Finite Volumes for Complex Applications, Porquerolle France*, 2002.

N. Méchitoua, M. Boucker, J. Laviéville, J.M. Hérard, S. Pigny, and G. Serre. An unstructured finite volume solver for two-phase water/vapour flows modelling based on elliptic oriented fractional step method. *NURETH 10, Séoul, Corée du Sud*, 2003.

T. Randrianarivelo, H. Neau, O. Simonin, and F. Nicolas. 3d unsteady polydispersed simulation of uranium hexafluoride production in a fluidized bed pilot. In *Proc. 6th Int. Conf. on Multiphase Flow, Leipzig (Germany), paper S6_Thu_A_46,*, 2007.

O. Simonin. Prediction of the dispersed phase turbulence in particle-laden jets. *Gas-Solid Flows-1991, ASME FED, Vol. 121*, pages 197–206, 1991.

O. Simonin. Statistical and continuum modelling of turbulent reactive particulate flows. *Theoretical and Experimental Modeling of Particulate Flows. Lecture Series 2000-06, Von Karman Institute for Fluid Dynamics, Rhode Saint Genèse, Belgium*, 2000.

A.G. Sunderland, M. Ashworth, N. Li, C. Moulinec, Y. Fournier, and J. Uribe. Towards petascale computing with parallel cfd codes. *Parallel CFD 2008, Lyon France*, 2008.