Reactive CFD in Engines with a New Unstructured Parallel Solver

M. Zolver[1], D. Klahr[1], J. Bohbot[1], O. Laget[1] and A. Torres[1]

1 Institut français du pétrole, division Techniques d'applications énergétiques,
1 et 4, avenue de Bois-Préau, 92852 Rueil-Malmaison Cedex - France
e-mail: marc.zolver@ifp.fr - diego.klahr@ifp.fr - julien.bohbot@ifp.fr - olivier.laget@ifp.fr - arnaud.torres@ifp.fr

Résumé — **Mécanique des fluides numérique dans les moteurs à l'aide d'un nouveau code parallèle non structuré** — KIFP, un code hexaédrique, parallèle et non structuré est en cours de développement à l'*IFP* depuis deux ans. Certains algorithmes de KMB, code *IFP* basé sur le logiciel KIVA II, ont été implantés dans KIFP. De part le nouveau formalisme non structuré, d'autres algorithmes (volumes finis sur grille décalée, pas de temps fractionné, boucle itérative SIMPLE, convection par sous-cycles, modèles de turbulence et de spray, etc.) ont dû être modifiés. De nouveaux algorithmes, comme le mouvement de maillage par interpolation temporelle conditionnée, l'algorithme permettant de changer de maillage en cours de simulation, les limiteurs de pentes de Van Leer et Superbee dans la phase de convection, le modèle de transferts thermique de Kays et Crawford et le modèle de combustion essence ECFM ont été ajoutés. Les machines superscalaires étant très utilisées et développées, KIFP a été optimisé pour être employé sur ce type de machines. De nombreux ordinateurs et compilateurs ont été testés. Dû à son faible coût d'implantation, le standard de directive OPEN-MP est utilisé pour paralléliser le code. Une accélération de 3 sur 4 processeurs a été atteinte. KIFP est maintenant capable de simuler correctement la combustion tridimensionnelle dans des moteurs automobiles. Comme exemple, nous montrons dans cet article des calculs sur un banc volute stationnaire à pipe hélicoïdale, une admission/compression/combustion dans un moteur à injection indirecte à allumage commandé avec une chambre en toit, et une compression/injection dans un moteur à injection directe Diesel. Des comparaisons avec des résultats expérimentaux pour les cas moteurs (vitesse de rotation et coefficient de *swirl* pour la simulation stationnaire, et des courbes de pressions moyennes pour les simulations instationnaires) ou avec des résultats analytiques pour les cas tests académiques ont été réalisées autant que possible.

Abstract — *Reactive CFD in Engines with a New Unstructured Parallel Solver* — *KIFP, an hexahedral unstructured parallel code is currently developed at* IFP *since two years. Some algorithms of the KIVA-II's version currently used at* IFP *(KMB) are implemented in KIFP. Due to the new unstructured formalism, others algorithms (finite volume on staggered grid, time splitting, SIMPLE loop, sub-cycled advection, turbulent and spray models, etc.) have to be modified. Some new ones, like the conditional temporal interpolation methodology for moving grids, the remapping algorithm for transfering quantities on different meshes during the computation, the Van Leer and Superbee' slope limiters for advective fluxes, Kay and Crawford's heat transfer model and ECFM gasoline combustion model have been added. Super-scalar machines being widely used and developed, KIFP has been optimised for running on this type of machine. Many computer platforms and compilers have been tested. Because of its low cost of implementation, the OPEN-MP standard paradigm is used to parallelize the*

code. A speed-up of 3 on 4 processors is reached. KIFP is now able to well simulate 3D combustion in actual automotive engines: as example, we show in this paper some computations of an helical port on a steady state bench, an intake/compression/combustion in a port fuel injection (PFI) spark ignited engine with a pent roof chamber and a compression/spray injection in a direct injection (DI) Diesel engine. Comparisons with experimental results for engine cases (rotation speed and swirl coefficient for the steady state simulation and mean pressure curves for unsteady ones) or with analytical results for academic test cases are done as much as possible.

INTRODUCTION

Understanding and developing new engine concepts require more and more help from 3D CFD and combustion modeling. Nevertheless, grid generation and computation time remain expensive and time consuming. For years, efforts are directed toward making these key points easier and faster, and it seems that unstructured grids running on parallel computers are efficient (Heel *et al.*, 1998; O'Rourke *et al.*, 1999; Tatschl *et al.*, 2001).

At *IFP*, for many years, a modified version of KIVA II (Amsden *et al.*, 1989), called KMB (Habchi and Torres, 1992; Torres and Henriot, 1996), has been developed and used. Based on structured multi-blocks formalism, it can model multi-valve engines of the last generation. A strong effort has been done improving physical sub-models, like combustion, pollutants and spray models (Torres and Henriot, 1994; Duclos and Zolver, 1998; Beard *et al.*, 2000; Colin *et al.*, 2003). Even if KMB is adapted to engine calculations (Henriot *et al.*, 1999), there are strong constraints like the time taken by structured grid generation in complex geometries or the exclusive use of very expensive vector computers. Also, bad shaped cells and corner cells in complex geometry may limit the numerical accuracy and highly increase the CPU time. To avoid those problems, it has been decided to develop a new hexahedral unstructured version of KMB, which is called KIFP, dedicated to parallel scalar computer platforms.

This paper, first, briefly describes the step by step development methodology of KIFP. More details can be found in Zolver *et al.* (2002). Then, the numerical approaches and developments are shown and discussed: moving grid algorithms, advection, multi-species diffusion, turbulence modeling, law of the wall and heat transfer models, open boundary conditions, gasoline combustion and finally spray models. Then, coding and parallel aspects are described. After showing some comparisons with analytical results, some computations on real automotive engines are done with KIFP and compared, when results are available, with KMB and with measurements: a steady state bench, an intake, compression and combustion strokes in a pent roof spark ignited engine and a compression/injection stroke in a DI Diesel engine. More engine test cases can be found in Zolver *et al.* (2001). At last, CPU performances and parallel speed-up are discussed.

1 NUMERICAL IMPLEMENTATION

1.1 Methodology and Unstructured Aspects

First of all, for developing KIFP, descriptors and their connectivity have to be defined. Like KMB, KIFP keeps the hexahedral description with cell-centered scalars and node-centered vectors. The topological element "face", already present in KIVA II, is more widely used in the new solver. This element can be linked easily to hexahedrons and nodes. It is defined by two or one hexahedrons (fluid or boundary faces) and 4 nodes, nodes 1 to 4 in Figure 1. Sorting cells and nodes indexes for a given face can give the orientation that we need for solving the transport equations. The 8 sorted border nodes, node 5 to 12 in Figure 1, are also registered and used for rising numerical accuracy in convection terms.

Because of the unstructured approach, the connectivity (node, cell and face) must be all computed and registered. Nodes can have any number of neighbours, while a hexahedron can only have 6 ones, respectively to its 6 faces.

One of the guidelines for developing KIFP was to follow KIVA II skeleton and discretisation. ALE (Arbitrary Lagrangian Eulerian) time splitting begins with the source terms (phase A), then follows a full implicit Lagrangian phase (phase B), and finally a sub-cycled explicit Eulerian phase (phase C). The original Semi-IMPLicitE method (Patankar, 1980) used in KIVA II for phase B is retained in its fully implicit version. Also, for each solver in phase B (species diffusion, velocity, internal energy, pressure, etc.) conjugate residual iteration method (O'Rourke and Amsden,
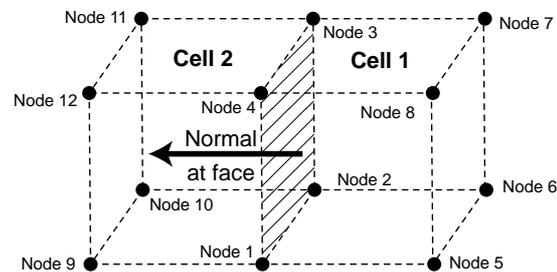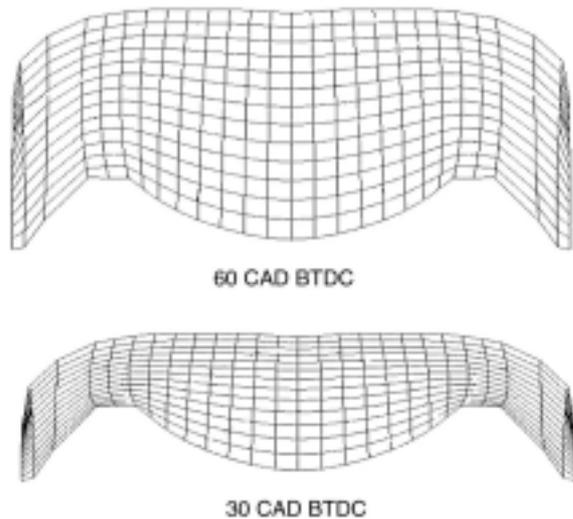

Figure 1

Face definition.

Figure 2

Corrected Laplacian relaxed grids.

1986) is still used. Improving modularity, generic routines, for computing a gradient or a diffusion term for example, are developed and widely used by different parts of the code. SI units are adopted for KIFP.

The methodology for building KIFP is to rewrite step by step all algorithms and closely compare the results with KMB. Moving the grid without the flow is first studied. Then, advection of a scalar and of the momentum is developed. After that, pressure solver is set to deal with inviscid flow. Here, Euler's equations can be solved. Introducing fluid viscosity, the Reynolds' stress tensor and all diffusion terms allow the Navier-Stokes' equations to be solved. Turbulence equations, the standard k-ε ones, are implemented easily thanks to the generic routines. Then, velocity projections at wall and law of the wall allow us to compute fluid dynamic. After the treatment of open boundaries, we have developed an algorithm for creating and destroying cells during the valves and piston movements. Finally source terms due to combustion and sprays are implemented.

1.2 Grid Motion

New algorithms to move a grid without any flow have been investigated. As found in literature, the spring analogy method (Sinha *et al.*, 1998) connects all nodes with springs and relaxes the whole system after any deformation. Nevertheless it proves itself to be too slow for following the piston or the valves motion and also it leads to cells with negative determinant. Also found into the literature, and

already used in KMB, the corrected Laplacian relaxation (Torres and Henriot, 1994) performs the well-known Laplacian for every node's positions weighted by a curvature term. This method allows sharp corners in the grid. It is very attractive but also expensive in terms of computing time because of the relaxation convergence. Figure 2 illustrates a part of a compression stroke in a cut plane through the piston bowl of a simplified GDI engine using this methodology. Another method, which was developed at *IFP*, uses an average of the displacement of every neighbour for a given node. By iterating, new positions can be computed. It gives good results for simple geometry, keeping the whole grid aspect ratio almost constant during the motion.

The temporal interpolation methodology (Duclos and Zolver, 1998) has been adapted from KMB to KIFP. In this method, a target grid at a final angle is fully computed outside the solver by classical grid generators. Then, during the run, the position of a node at a given time is computed by an interpolation between its old position and its target position weighted by its proximity from the piston and the valves. Of course, actual piston and valves laws are followed exactly during the interpolation. As expected, because this algorithm works already well in KMB, good results are observed. A view of the grid of a pent-roof engine during intake stroke is shown in Figure 12 (*see Section 4.2*). Nevertheless, to estimate the weight of the piston and the valves on a given node displacement, full advantages of the structured formalism is used in KMB. In KIFP, an iterative algorithm is needed and therefore the method is slightly more CPU time consuming. Following our experience with temporal interpolation, target grids are needed at least for each maximum geometrical constrain, which means approximately every 60 crank angle degrees (CAD) depending on the complexity of the engine. The first initial grid will be given at intake TDC (which means 360 CAD before combustion TDC) or at valve opening time, then a second one 60 CAD after (where the valves and the piston are very close), a third one at full valve opening time (which is commonly around 100-120 CAD after the opening), a fourth one at intake bottom dead center (BDC) and, at last, a fifth grid at valve closing time. Of course, some grids have to be added in case of exhaust stroke calculations with valves overlap. For compression and combustion only one or two grid might be enough.

Finally, for compression runs, it is possible to use a vertical-scaling algorithm. With this method, the new position of a node is computed from its old one keeping the Z coordinates ratio constant between a top fixed node and a bottom piston-like moving node. These two limiting nodes can be identified for each node of the grid at the beginning of the stroke. In that part of computation, full advantage of unstructured formalism can be taken, filling the roof with fixed hexahedrons while the cylinder is filled by pile-up cells.

1.3 Eulerian Phase

Like in KMB, advective fluxes for scalar (species, internal energy, turbulence related quantities) and vector (momentum) are computed using slope limiters (Leveque, 1992). Using the cells/faces or nodes/faces connectivity as shown in Figure 1, slopes at cells or nodes can be computed. Following our experience in 3D computations with KMB, Superbee's limiter is applied for scalars, and Van Leer's limiter for vectors. The computed slopes allow an estimation of scalar value at a face and momentum value in the middle of an edge. This ensures second-order upwind advection for every scalar and momentum.

1.4 Lagrangian Phase

In a first step, KIFP deals with inviscid fluid, and for that, pressure gradient and implicit pressure solver are needed. Pressure gradient at nodes and pressure forces around faces (Amsden *et al.*, 1989) are obtained from pressure at cell's center and face area vectors. Like KMB, pressure boundary conditions at the walls are linearly interpolated from pressures near the walls. The inversion method for the pressure matrix remains the conjugate residual iteration method (O'Rourke and Amsden, 1986) with a diagonal pre-conditioner. Although memory is not a limiting factor in engine's CFD for today's computers, we have taken care to never use this matrix (and all the others along the phase B) globally but only few elements at each time, which means that KIFP needs not very more memory than KMB. Introducing fluid viscosity makes all the Lagrangian phase complete. Solving mean Reynolds' stress tensor and diffusion of internal energy allows us to compute laminar flow. Velocity, internal energy and pressure solvers give the numerical core of the full implicit loop in the Lagrangian phase, like in KIVA II. One diffusion routine is developed for solving all diffusion terms (internal energy, mass diffusion and standard k-ε turbulence diffusion). Then multi-species and turbulent flows are very easily implemented.

1.5 Open and Walls Boundary Conditions

Because the grid resolution is generally low near the walls, turbulent law of the wall and heat transfer law should be used. Kays and Crawford law (Angelberger *et al.*, 1997) is available in KIFP. Kinetic energy dissipation and heat transfer are directly integrated in the turbulence and internal energy equations. Velocities are corrected and projected on the wall under slip conditions. These projections used in phase B and C are done knowing each orthogonal vector to each face. Three kinds of nodes on the wall are defined. For free nodes no projection is done. For nodes near a geometrical acute corner, a projection should be done following the corner direction. For more general nodes, a projection should be done on a tangential plane at the wall.

In order to solve intake and exhaust strokes for engines, open boundaries were implemented in KIFP following the KMB formalism. Specific flags are set for the inlet and outlet nodes that allow building inlet and outlet face arrays. At inlet, the time dependent mass flow rate, which is often known from 1D analysis or deduced from measured intake pressure, is set. From the entrance area and the mass density downstream, a velocity is deduced. This velocity is set at the input nodes. Time dependant scalar quantities, like species or energy (internal and turbulent) are fixed upstream of the input faces. From that, input fluxed quantities are computed. At outlet, a fixed pressure is set at a given capacity distance. Then all quantities, like velocity at output nodes and scalar fluxes through output faces, can be computed. Pressure waves can also go out when the capacity distance is non-zero. Finally, like in KMB, it is assumed that no diffusion occurs at inlet and outlet.

1.6 Remapping Algorithm

Because grids undergo very strong distortions during the four engine's strokes, it is necessary to add or to remove cells. The methodology used in KIFP is to change the grid at given times during the run. All needed grids can be generated outside the solver by grid generators. The algorithm used, called "remapping", locates each nodes or cell's centers from the new grid inside the old one. Then a set of neighbours in 3D is selected and a matrix interpolation is done on their scalar and vector values. To ensure conservativity, corrections can be applied on the total mass, energy or turbulence quantities. Computations are fully done inside the solver itself during the engine stroke simulation.

Figure 3 shows the remapping algorithm used between unstructured and structured grids. The test case is a diffusion one with a moving piston. Calculation 1 is done completely with the same grid. Calculation 2 undergoes 2 remappings. Results at final time (TDC) on the different grids are quite similar. Figure 12 (*see Section 4.2*) shows some grids used in an intake stroke with 3 remappings.

1.7 Combustion Modeling

Gasoline combustion is available in KIFP. The extended coherent flame model (ECFM) is implemented with a laminar-turbulent transition time correlation for ignition (Duclos and Zolver, 1998; Duclos *et al.*, 1999). This correlation, deduced from experimental data (Baritaud, 1989), is used to determine the time at which a kernel of flame surface density (Σ) is deposited in the mesh. Then a transport equation for Σ is solved with all the production and dissipation terms. Because of Reynolds averaged Navier-Stokes' formalism, only average state is known in a cell. Already coded in KMB, ECFM is implemented in KIFP with the computation of three different states in a given cell: the
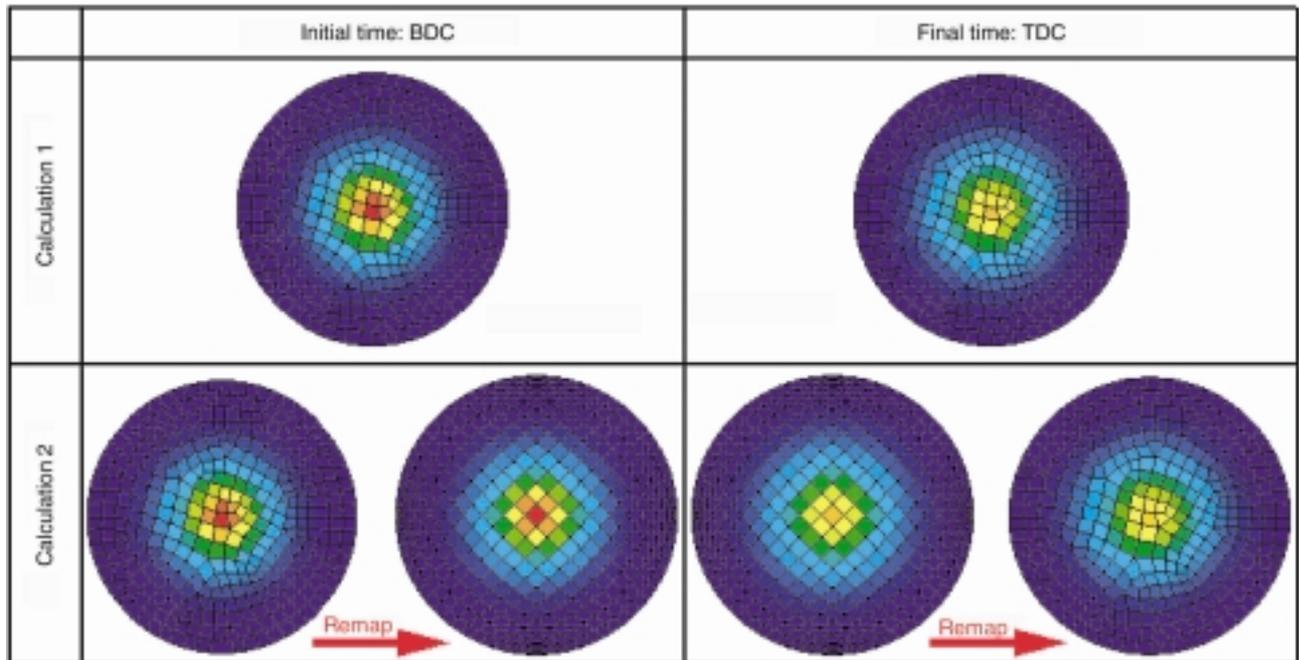
Figure 3

Calculations with and without remappings on a diffusion test.

mean state, the fresh gas state and the burned gas state. This implementation gives a better precision for computing the flame properties, the equilibrium reactions involving CO (Amsden *et al.*, 1989), and the kinetic reactions involving NO following an extended Zeldovitch's mechanism (Heywood, 1988). Building these three states at every time steps for every cells is done by defining three fictitious species, one for the fuel, one for the oxygen and one for the fresh gas enthalpy (Colin *et al.*, 2003). As in KMB, these fictitious species, along with the flame surface density, are diffused and advected.

Indeed, in order to make the use of all species easier in combustion and chemistry, three kinds are defined: the fuel species (for multi-fuel combustion), the chemical species other than fuels (oxidants, neutral species, products of combustion, etc.) and the fictitious species. Pointers with an explicit name and tabulated libraries in International System Units are defined for all of them. Then fully stratified multi-fuel charge can be computed for gasoline engines.

1.8 Spray Modeling

Finally Lagrangian spray modeling is developed in KIFP. KIVA-II methodology (Amsden *et al.*, 1989) is adapted to unstructured mesh. One of the main tasks is to locate efficiently all particles in the Eulerian grid. This is done by using the relative position of a particle to faces and by using
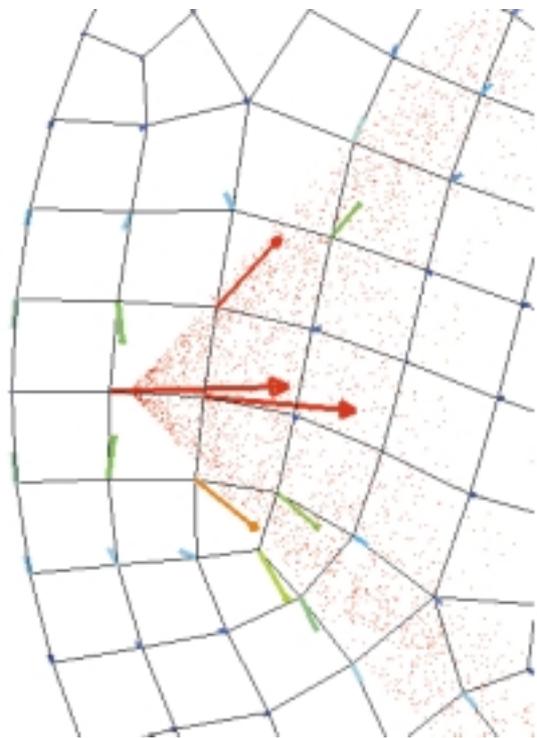

Figure 4

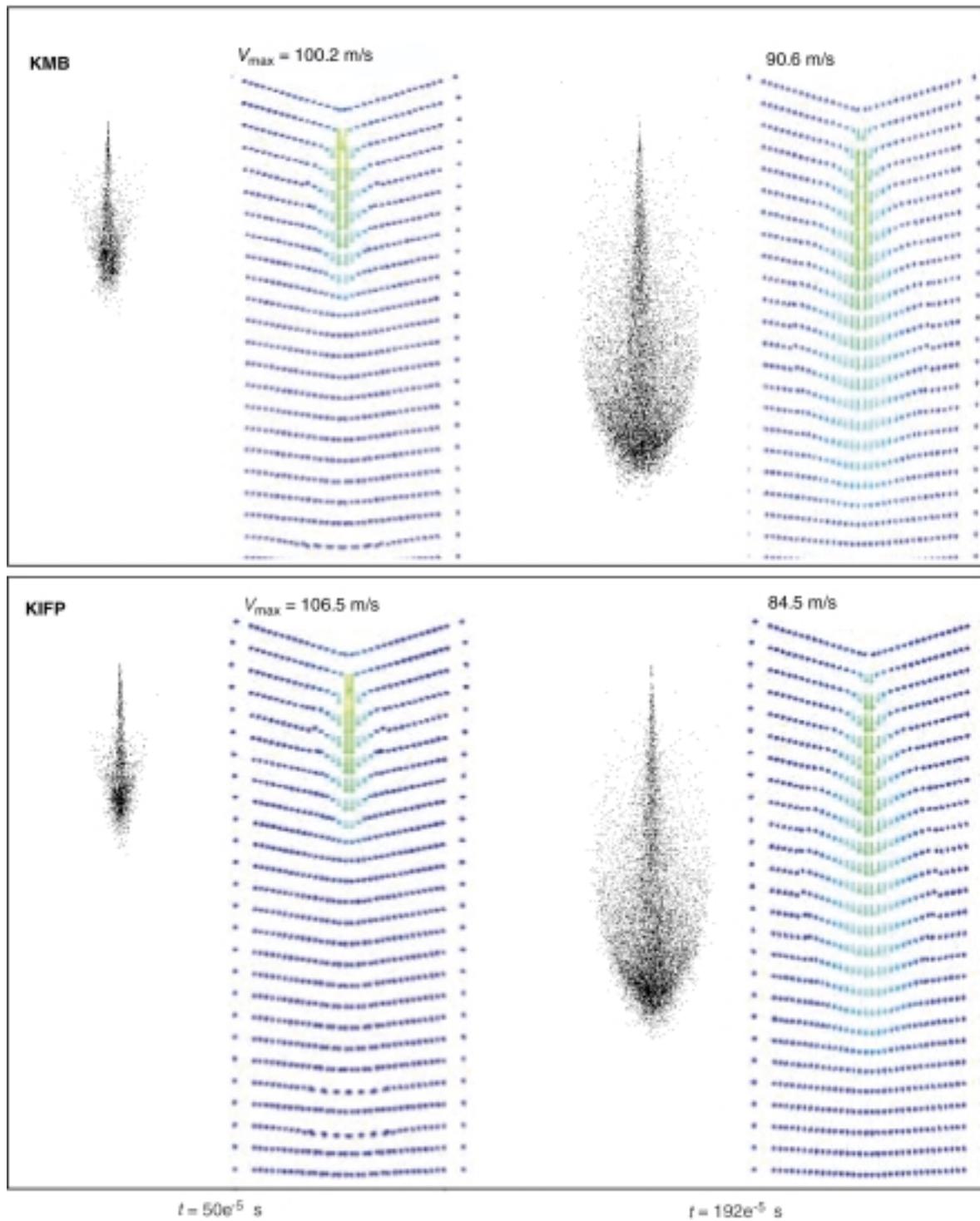Example of a liquid spray and a velocity field on unstructured grid.

Figure 5

Comparisons KMB *vs* KIFP on liquid sprays and velocity fields in a high-pressure cell (P = 61 bar, T = 800 K, P_{inj} = 800 bar) (1 bar = 0.1 MPa).

the face connectivity. This algorithm is the same that the one used for remapping task (*see Section 1.6*). Spray source terms, like injection and evaporation, are implemented in a full multi-nozzle, multi-rate and multi-component way. Break-up and collision are not implemented yet. A χ^2 distribution is used for the particles initial sauter mean diameter. Spray/wall interaction is modelled roughly for the moment, as particles are simply stuck on the wall.

Coupling mass, momentum and energy between Lagrangian and Eulerian phases is done in phase B and is shown working well in Figure 4 on an unstructured grid. A comparison between KMB and KIFP is also shown in Figure 5 for an injection of a Diesel spray in a high-pressure cell without evaporation. Both solvers give the same results.

2 CODING ASPECTS

One of our main goals was to develop a parallel solver efficient on the latest generation of computers at the lowest cost of development. The ratio between performance and development cost decided us to use a data parallelism model which is quite easy to implement using only compiler directives: the OPEN-MP paradigm (www.openmp.org), which is also quite standardized, portable and scalable. These reasons and the global evolution of the scientific computational world require developing on super scalar SMP machines.

As in KMB, we obviously still program using Fortran 90 but taking fully advantage of dynamic allocation and modularity. After a profiling of the sequential version, we note that most CPU time is spent in very few loops in pressure solver and diffusion terms computation. Implementing OPEN-MP directives into them gives a parallelization of about 50% of the code. Parallelization was also developed for the combustion routines. These encouraging results drive us to instrument more routines in order to achieve similar speed-up on the entire code.

Finally, after spending little time on the performance aspects, we obtain a portable code already running on several platforms like SGI O2000, SGI O3000, Compaq ES40 and ES45 and PC under Linux (using PGHPF compiler).
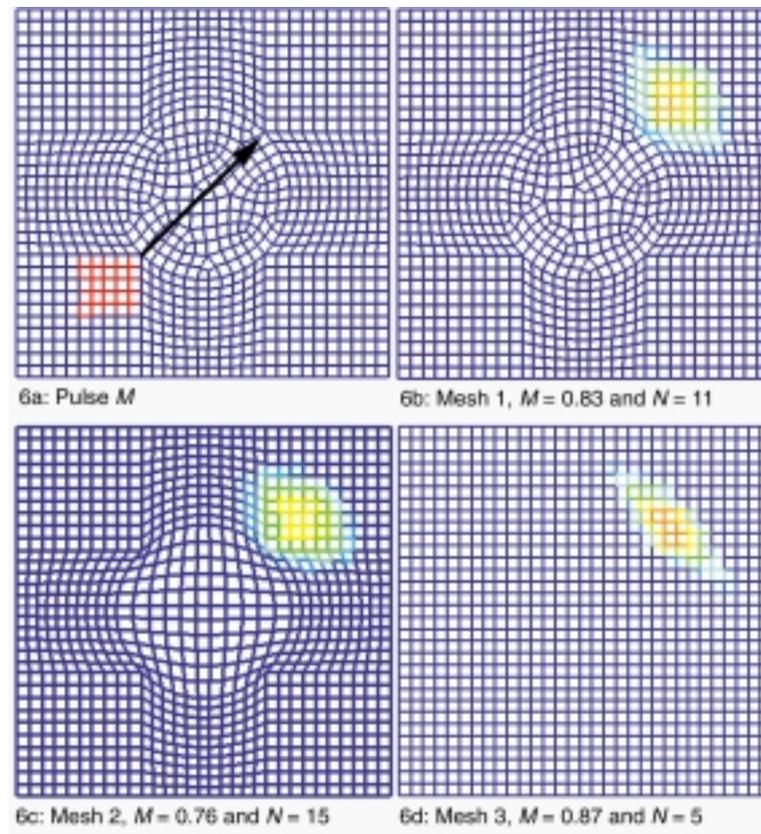

6a: Pulse *M*

6b: Mesh 1, *M* = 0.83 and *N* = 11

6c: Mesh 2, *M* = 0.76 and *N* = 15

6d: Mesh 3, *M* = 0.87 and *N* = 5

Figure 6

Advective transprt of a scalar pulse.

3 RESULTS ON ACADEMIC TEST CASES

3.1 Advection Test

Figure 6a shows a squared scalar pulse, with an initial value (*M*) equals to 1 for 5×5 cells in the lower left side of the mesh, advected in a uniform and constant 45° tilted velocity field with a CFL of 0.2. Three meshes are used. The first one (mesh 1 in *Fig. 6a* and *6b*) is unstructured, mapping a central cylindrical area (which is a common shape found in engine). The second one (mesh 2 in *Fig. 6c*) is a structured mesh mapping the same central area. And the third one (mesh 3 in *Fig 6d*) is a finite difference regular mesh. As seen in these figures, the global shape of the pulse after crossing the central zone is more preserved with mesh 1 than with meshes 2 and 3. For mesh 2, distorted cells induce dissipation of maximum value *M* of the pulse (from 1 to 0.76) and increase the advection sub-cycle number *N* from 11 with mesh 1 to 15. Mesh 3, even if it preserves a better pulse maximum, *M* = 0.87, and has a low sub-cycle number, *N* = 5, completely stretches the pulse's shape perpendicularly to the motion. The unstructured mesh 1 shows a better quality because of the lack of privileged direction in the central cylindrical area. Numerical accuracy is better thanks to the lack of bad shaped cells. The maximum value of the pulse stays high at 0.83, its shape remains more or less squared and the sub-cycle number stays reasonably between the mesh 2 and 3 values.
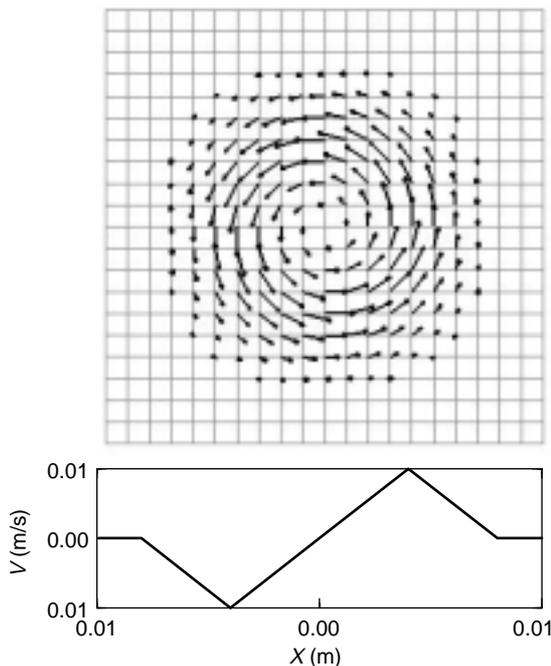

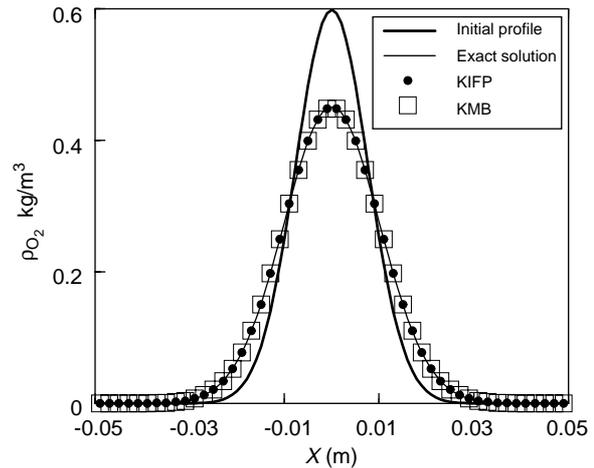
Figure 7

Initial velocity field for the Gresho's case.

Figure 8

Mass diffusion.

3.2 Gresho's Test

When the Lagrangian phase is coded, the inviscid Gresho's case (1990) can be solved. A stationary vortex is set in a 3D cubic box of length 0.02 m. The velocity profile along radius is triangular as shown in Figure 7. During 3 ms of physical time, with a time step of 0.025 s, *i.e.* a CFL of 0.5, a decay of kinetic energy occurs. Thanks to the slope limiters, Superbee and Van Leer, 77% of this energy remains in the box with KIFP, which is close to 74% for CHAD (O'Rourke and Sahota, 1998) when only 62% remains with KMB and its minmod (QSOU) limiter.

3.3 Diffusion Test

Figure 8 illustrates mass diffusion of oxygen mixed with nitrogen and carbon dioxide in a 3D box with $99 \times 9 \times 9$ cells of each. Gaussian profiles for O_2 and N_2 mass fraction are initiated at time t_0 in the *x*-direction, mass fraction being deduced from them. Initial mass fractions in the center of the box, at $x = 0$, are equal for O_2 and N_2, and 0 for CO_2. A strong diffusion coefficient $D = 0.0171$ m²/s is set. Pressure and temperature remain uniform and constant (respectively 105 Pa and 300 K). The analytical solution of this problem is well known and given by the following Equation (1) for O_2:

$$\rho(t, x) = \rho(t_0, x) \sqrt{t_0} \Big/ \sqrt{t} \cdot e^{-\frac{x^2}{4Dt}} \qquad (1)$$

Solving this diffusion problem during 1.3 ms with a time step of 0.01 ms gives good agreement between KIFP, KMB and the analytical solution as shown in Figure 8.

4 RESULTS ON ENGINE CONFIGURATIONS

4.1 Steady State Bench

In order to check open boundary conditions in KIFP, a steady state bench is presented. Around 285 000 cells and 300 000 nodes are used to mesh the geometry. The port is helical, and an outlet replaces the piston. The bore is 0.078 m. The intake duct and the cylinder are elongated and the valve lift is taken at 0.008 m. O-grids are widely used especially in the intake duct, Figure 9, the cylinder and all around the valve as shown in Figure 10. The experimental mass flow rate of 0.038 kg/s is set at the input and a constant pressure of 91 600 Pa is fixed 0.05 m after the output. The time integration done by KIFP is used to achieve the steady state at convergence.
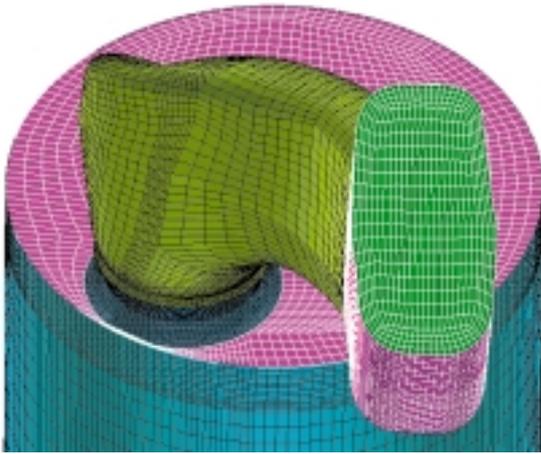
Assuming a solid body rotation in the cylinder, the experimental and computed rotation speeds ω and the swirl coefficients can be compared. The latter is defined as the mean radial velocity $V_r = R\omega$ (where R is half of the bore) over the mean axial velocity V_z. Values are given in Table 1 for KIFP, KMB and the experiment. Although KIFP results are closer from experiment, both codes seems to underestimate the swirl motion generated by the helical manifold. Some reasons may be stated. First, elongating the intake manifold in the computational grid may be not the best way to simulate the ambient room conditions. Second, the measurements were made with a honeycomb inside the cylinder, which is not taken into account in the calculations.

TABLE 1

Rotation speed and swirl coefficient

	ω (rad/s)	V_r/V_z
KIFP	730	3.75
KMB	690	3.60
Experiment	940	5.1

Figure 9

Steady state bench grid with an helical port.
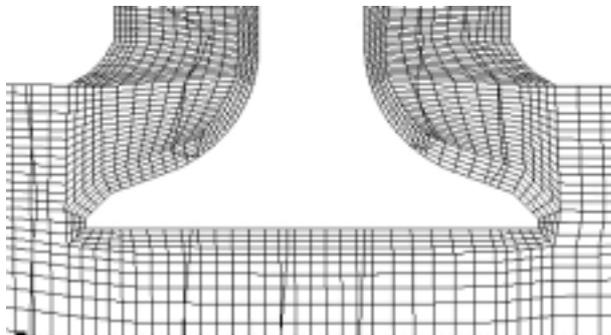
Figure 10

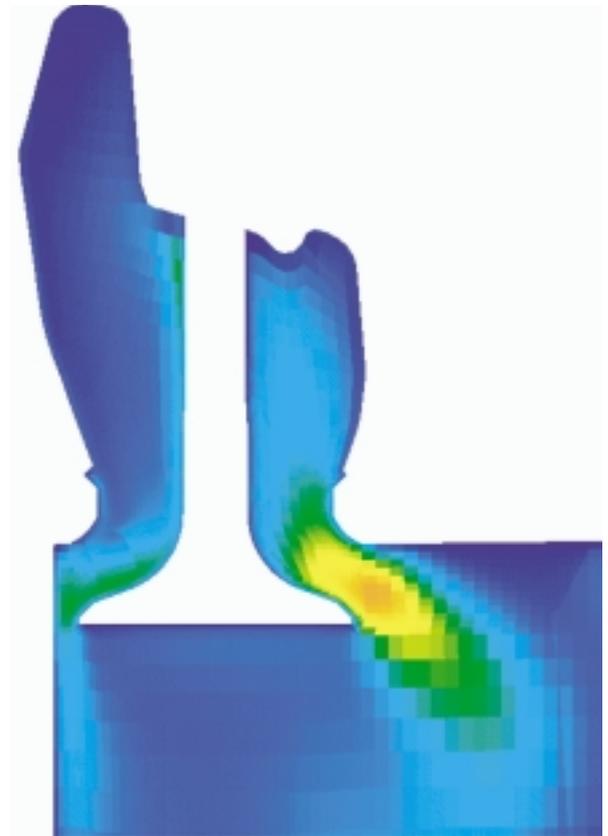O-grid around the valve of the steady state bench.

Figure 11

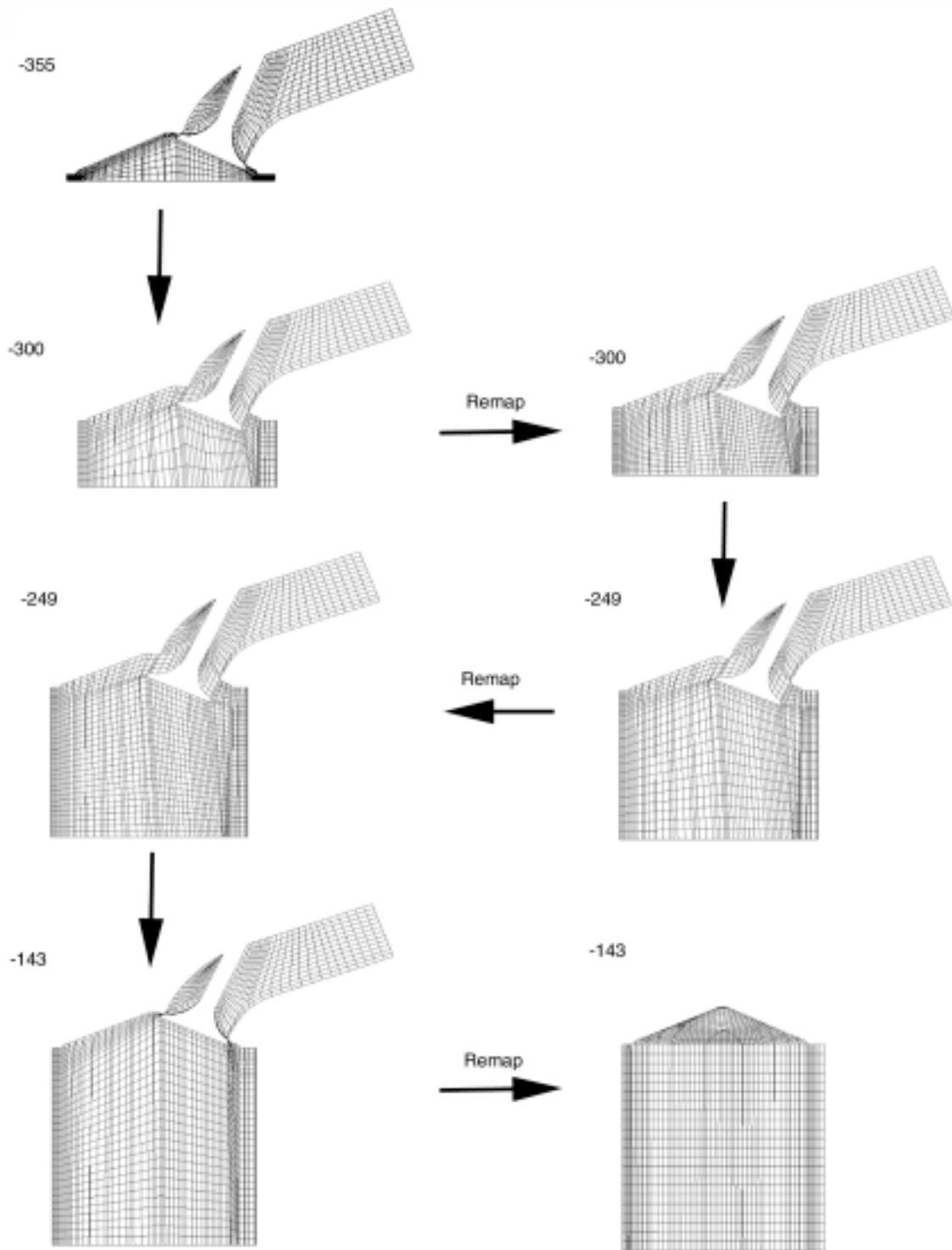Turbulent kinetic field in the steady state bench (max.: 1300 m²/s²).

Figure 12

View of some different meshes used during the intake stroke of a pent roof spark ignited engine.

Figure 11 gives a view of the upper part of the cylinder with a turbulent kinetic energy field. The cut plane goes through the valve and the center of the cylinder. The typical valve jets are observed with a maximum velocity around 155 m/s. As expected turbulent kinetic energy is higher near the highest velocity gradients, that is to say in the maximum shear areas. Of course, these zones are located in the valve curtain between the seat edge in the upper part of the jet and the valve edge in the lower part.

4.2 Intake, Compression and Combustion in a SI Engine

An PFI gasoline pent-roof engine is computed during 395 CAD. Its specifications and the engine operating point are given in Table 2.

TABLE 2

Engine specifications and operating point of the indirect injection gasoline engine (1 bar = 0.1 MPa)

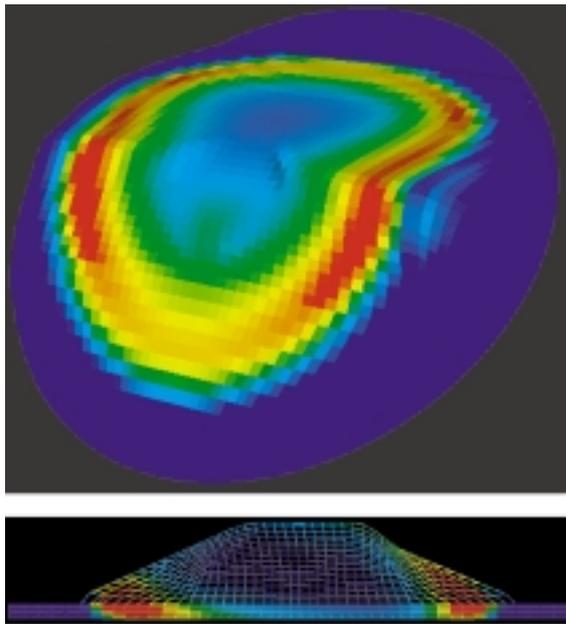| Bore-stroke	0.086 x 0.086 (m)
Compression ratio	10
Rpm	2000
Valve opening/closing time	355/143 CAD BTDC
Full valve opening time	249 CAD BTDC
Intake valve	1
Equivalence ratio	0.7
Fuel	Propane
BMEP	2 bar
Spark timing	40 CAD BTDC

Figure 13

Flame surface density fields at TDC (max.: 700 m^{-1}).

Computation starts from the valve opening time and stops 40 CAD ATDC. The 1D simulated flow rate is set at boundary condition during the intake stroke. Concerning physical results, more details can be found in a previous paper (Zolver *et al.*, 2002) where a part of this intake stroke is presented, but we can say that the resulting flow at IVC is characterize by a tumble motion. We want in this section to focus on the mesh deformation and on the remapping strategy. Figure 12 summarizes quite well all those aspects: eight meshes (one at –355 CAD, two at –300 CAD, two at –249 CAD, one at BTC and two at –143 CAD) are used during the intake stroke, with 3 remappings corresponding to mid valve opening, full valve opening and valve closing times.

After the compression stroke, the combustion one is then modeled. Some typical results of flame surface density fields are shown in Figure 13 (field on the skin of the mesh and within a cutting plane perpendicular to the pent roof): we clearly see the effect of the high tumble flow on the flame front. We plot in Figure 14 the evolution of the mean cylinder pressure, both for experiment and calculation. A good agreement is observed. Nevertheless, ignition seems to occur too late for calculations, which leads to a slight delay for the peak pressure. This should be improved when a new ignition model (Duclos and Colin, 2001) will be implemented in KIFP.
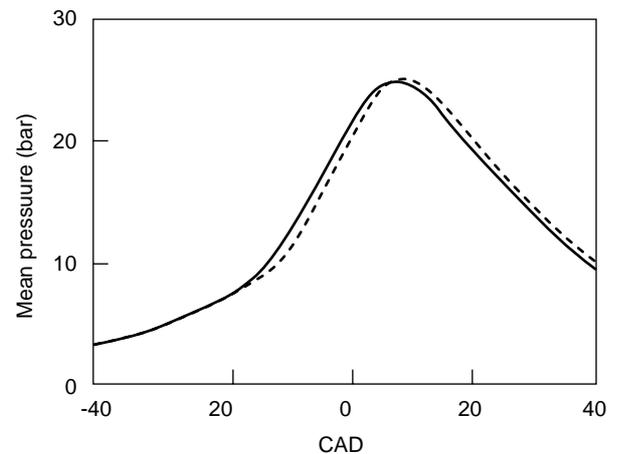

Figure 14

Mean cylinder pressures for experiment and calculation.

4.3 Compression and Injection in a DI Diesel Engine

Engine specifications and the operating point are given in Table 3. Note that combustion is not modeled in this case.

TABLE 3

Engine specifications and operating point of the DID engine

Bore-stroke	0.085×0.088 (m)
Compression ratio	7
Rpm	4000
Injection timing	15.1 CAD BTDC
Injection duration	37.6 CAD
EGR	0%
Lambda value	1.41
Fuel	60% C_7H_{16}/40% $C_{12}H_{26}$
Number of spray holes	6
Total fuel mass	42.86 mg

Calculation starts at 20 CAD BTDC and stops at the end of injection. Initial conditions (swirl, temperature, pressure and turbulent quantities) are given by a previous KMB calculation. 5000 particles of the blend of fuel are injected per hole.

We plot in Figure 15 a view of the unstructured mesh. 34 000 nodes and 30 000 cells are used, which are of course not enough for an optimum accuracy. But this case is only here to note the good behaviour of the code in two phase flow situation. O-grids are used in the piston bowl and near the liner (only in *xy* plane).

Figure 16 is dedicated to the interaction of the swirling flow and the sprays, with a velocity field crossing the piston bowl. The tip of each spray is clearly deviated by the fluid flow. On the other part, the swirling flow is destroyed along the spray, due to the very high spray injection velocities.

Figure 17 shows some fuel vapour fields for the two fuels at TDC. The cutting plane is the same than the previous figure. The heptane maximum density is higher than the dodecane one, due to its higher critical pressure and its higher initial amount. As the liquid phase, we note the effect of swirl motion on the vapour field.
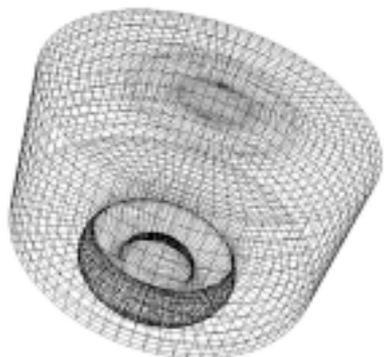
5 CPU PERFORMANCES AND PARALLEL SPEED-UP

5.1 CPU Performances in Sequential Mode

Table 4 shows the characteristics of the different platform we used during this study. We note that both SMP machines (the fourth on the left-hand side of the Table) and PC under Linux are tested. On the right hand side, we put the NEC SX5 vector computer characteristics, which is used usually to run KMB.

In sequential mode, tests were performed in a spark ignited engine configuration (Zolver *et al.*, 2001). Computations start at BDC and finish 40 CAD ATDC. The normalised elapse times for KIFP (left hand side) and KMB (right hand size) are given in Figure 18 for all the computer platforms. We note that the best results are, as expected, for KMB on the expensive but very powerful vector processor of the NEC SX5 (peak performance of 8 GFLOPS). Concerning the two codes, and irrespective of the platform, we observe that KIFP is more adapted to scalar machines than KMB (-17% on CPU time on a SGI O2000). At last, we note that the best performance on a PC under Linux is in the range of the SMP computer performances.

5.2 Parallel Speed-Up

In parallel mode, tests were done in a compression/expansion (5 CAD BTDC to 5 CAD ATDC) in a DI Diesel engine of 103 000 cells. Injection and combustion are not modeled in this test case. Compared to the previous section, the IBM and NEC computers are not tested. Figure 19 gives the speed-up on 2 and 4 processors for all the other platforms. Parallel performances are clearly lower for PC under linux than for modern SMP machines. On this type of platform, a speed-up of 3 on 4 processors is reached.
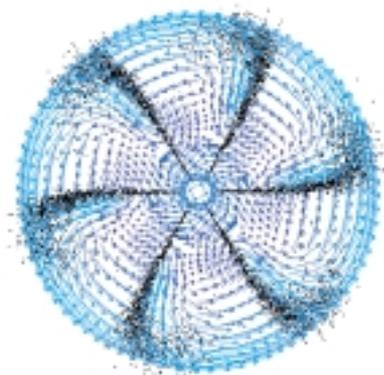

Figure 15

DID engine mesh.

Figure 16

Liquid sprays and velocity field at TDC (max.: 245 m/s; 7782 particles).

TABLE 4

Computer characteristics

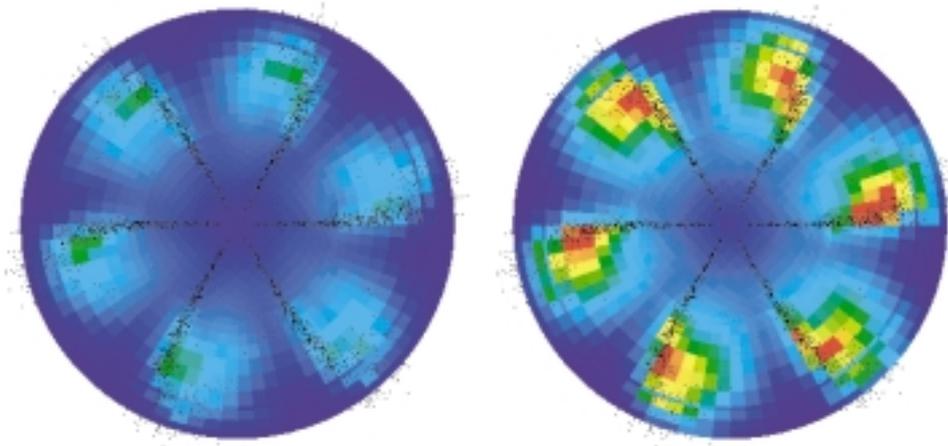|	SGI O2000	SGI O3000	COMPAQ ES45	IBM Power 4	PC 2.2 GHz	PC 1.6 GHz	NEC SX5
Processor	R10K 195 MHz	R14K 600 MHz	EV68 1 GHz	Power 4 1.3 GHz	PIV 2.2 GHz	PIV 1.6 GHz	NEC vector proc.
Nb. procs.	20	16	4	16	2	4	5
Cache size	4 Mo	8 Mo	8 Mo	16 Mo	512 Ko	256 Ko	-
Memory	10 Go	32 Go	8 Go	256 Go	2 Go	1 Go	40 Go
Compiler	MipsPro 7.3	MipsPro 7.3	CPQ F90 5.5	xlf90	PGHPF 3.2	PGHPF 3.2	F90 Rev 212

Figure 17

Liquid sprays and vapour fields for dodecane (left) and heptane (right) at TDC (max.: 2.5 kg/m^3).
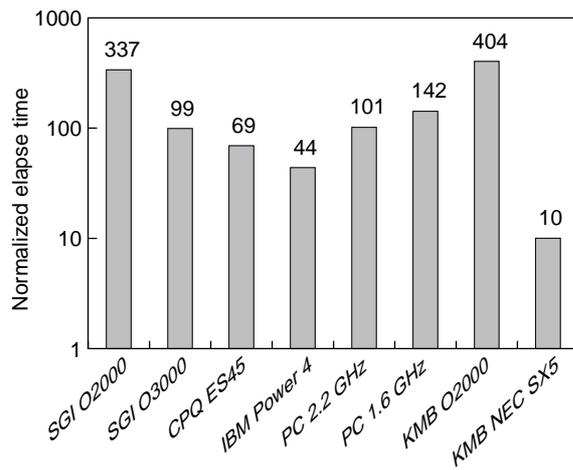

Figure 18

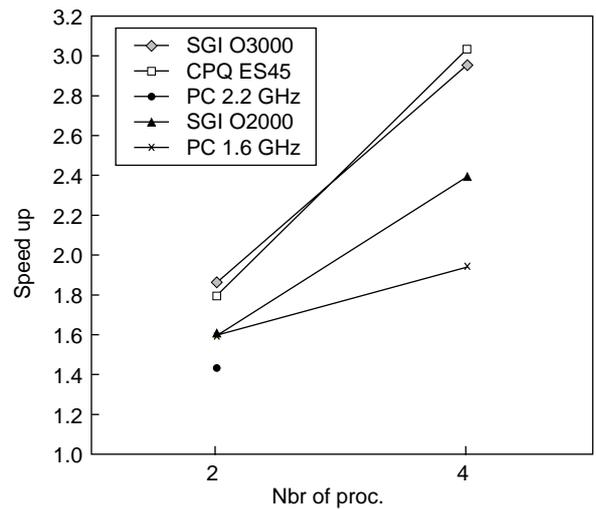CPU performances for KMB and KIFP in sequential mode.

Figure 19

Speed-up fur different computers.

CONCLUSION

KIFP, a new unstructured parallel reactive CFD solver, has been developed and validated by comparisons with analytical, numerical and experimental results.

Academic tests show some improvements in numerical accuracy due to a better mesh configuration. Computations in engines are also done. A steady state bench, an intake/compression/combustion strokes in a PFI spark ignited pent roof engine and a compression/injection strokes in a DI Diesel engine have been shown and successfully compared with experimental values when it was possible. On the latest simulation, an original multi-fuel components calculation is performed. More extensive comparisons will be done in future, especially with more refined grids.

Computations on parallel shared memory superscalar machines give some encouraging results, with a speed-up of 3 for 4 processors on up-to-date super scalar computers. We think that, soon, computing an actual automotive engine with KIFP using 4 processors on the latest SMP computers will be no more time consuming than today with KMB running on an expansive vector computer.

For the next step, the last physical sub-models of KMB, including spray models (break-up, liquid film, spray/wall interaction, etc.), up-to-date combustion and pollutant models (AKTIM ignition model, equivalence ratio fluctuations model, CFM3Z Diesel combustion model, soot model, etc.) will be implemented in the code. We will have to keep a good speed-up on parallel machines by extending the OPEN-MP instrumented parts of the solver. At last, a regular survey of unstructured hexahedral grid generators is necessary for taking full advantages of new products and to facilitate the use of CFD tools for helping in engine development.

REFERENCES

Amsden, A.A., O'Rourke, P.J. and Butler, T.D. (1989) KIVA-II, A Computer Program for Chemically Reactive Flows with Sprays. *Report LA-11560-MS*, Los Alamos National Laboratory.

Angelberger, C., Poinsot and T., Delhaye, B. (1997) Improving Near-Wall Combustion and Wall Heat Transfer Modeling in SI Engine Computations. *SAE Paper 972881.*

Baritaud, T. (1989) Combustion and Fluid Dynamic Measurements in a Spark Ignition Engine: Effects of Thermochemistry and Velocity Fields; Turbulent Flame Speed. *SAE Paper 892098.*

Beard, P., Duclos, J.M., Habchi, C., Bruneaux, G.,Mokkadem, K and Baritaud, T. (2000) Extension of Lagrangian-Eulerian Spray Modeling: Application to High Pressure Evaporating Diesel Sprays. *SAE Paper 2000-01-1893.*

Colin, O., Benkenida, A. and Angelberger, C. (2003) A 3D Modelling of Mixing, Ignition and Combustion Phenomena in Highly Stratified Gasoline Engines. *Oil & Gas Science and Technology – Rev. IFP*, **58**, 1, 47-62.

Duclos, J.M. and Zolver, M. (1998) 3D Modeling of Intake, Injection and Combustion in a DI-SI Engine under Homogeneous and Stratified Operating Condition. *COMODIA 1998.*

Duclos, J.M., Zolver, M. and Baritaud, T. (1999) 3D Modeling of Combustion for DI-SI Engines. *Oil & Gas Science and Technology - Rev. IFP*, **54**, 2, 259-264.

Duclos J-M. and Colin O. (2001) Arc Kernel Tracking Ignition Model for 3D Spark-Ignition Engine Calculations. *Paper (2-25), COMODIA 2001.*

Gresho, P.M. (1990) On The Theory of Semi-Implicit Projection Methods for Viscous Incompressible Flow and its Implementation via a Finite Element Method that also Introduces a Nearly Consistent Mass Matrix, Part 2: Applications. *Int. Journal Of Numerical Methods* for *Fluids,* **11**, 621.

Habchi, C. and Torres, A. (1992) A 3D Multi-Block Structured Version of the KIVA-II Code. *First European CFD Conference Proceedings*, 502-512.

Heel, H., Maly, R., Weller, H.G. and Gosman A.D. (1998) Validation of SI Combustion Model over Range of Speed, Load, Equivalence Ratio and Spark Timing. *COMODIA 1998.*

Henriot, S., Chaouche, A., Cheve, E., Duclos, J.M., Leduc, P., Menegazzi, P., Monnier, G. and Ranini, A. (1999) NSDI-3: A Small Bore GDI Engine. *SAE Paper 1999-01-0172.*

Heywood, J.B. (1988) *Internal Combustion Engines Fundamentals*, Mc Graw Hill Publication, 1988.

http://www.openmp.org

Leveque, R.J. (1992) Numerical Methods for Conservation Laws. *Lectures in Mathematics*, ETH Zürich, Birkhäuser Verlag.

O'Rourke, P.J. and Amsden, A.A. (1986) Implementation of a Conjugate Residual Iteration in the KIVA Computer Program. *Report LA-10849-MS*, Los Alamos National Laboratory.

O'Rourke, P.J. and Sahota, M.S. (1998) A Variable Explicit/Implicit Method for calculating Advection on Unstructured Meshes. *Journal of Computational Physics,* **143**, 312-345.

O'Rourke, P.J., Zhang, S. and Sahota, M.S. (1999) A Parallel, Unstructured-Mesh Methodology for Device-Scale Combustion Calculations. *Oil & Gas Science and Technology – Rev. IFP*, **54**, 2, 169-173.

Patankar, S.V. (1980) *Numerical Heat Transfer and Fluid Flow*, Hemisphere Publishing Corp., Washington D.C.

Sinha, N., Cavallo, P.A., Lee, R.A., Hosangadi, A., Kenzakowski, D.C, Dash, S.M, Affes, H, and Chu, D. (1998) Novel CFD Techniques for In-Cylinder Flows on Tetrahedral Grids. *SAE Paper 980138.*

Tatschl, R., Gabriel, H.P. and Priesching, P. (2001) FIRE – A Genetic CFD Platform for DID Engine Mixture Formation and Combustion Simulation. *11th International Multidimensional Engine Modeling User's Group Meeting at the SAE Congress.*

Torres, A. and Henriot, S. (1994) 3D Modeling of Combustion in Lean Burn Four-Valve Engines: Influence of Intake Configuration. *COMODIA 1994.*

Torres, A. and Henriot, S. (1996) Modeling the Effects of EGR Inhomogeneities Induced by Intake Systems in a Four-Valve Engine, *SAE Paper 961959.*

Zolver, M., Torres, A. and Klahr, D. (2001) CFD and Combustion in Engines with an Unstructured Parallel Solver Based on KIVA. *Paper (1-14), COMODIA 2001.*

Zolver, M., Klahr, D. and Torres, A. (2002) An Unstructured Parallel Solver for Engine Intake and Combustion Stroke Simulation. *SAE Paper 2002-01-1120.*