Internal combustion engine design on IBM platforms

by F. Papetti
S. Golini
M. Maggiore
S. Succi
P. Gaillard
J.-M. Perez

Computer simulation of fluid flow and combustion in diesel engines is rapidly gaining an increasing popularity within the automotive industry, becoming recognized as a cost-effective tool for cutting design cycle time. This paper shows how an advanced computing environment for numerically intensive applications, entirely based upon IBM platforms, can be profitably exploited within the framework of a joint project with industrial partners, in this case Renault Vehicules Industriels. The computing environment has been applied to the code KIVA-II, a computer program for numerical combustion developed at Los Alamos National Laboratory. Numerical simulations have been performed to assess the capability of the code to correctly reproduce the experimental data. Several features, such as visualization of the fuel spray droplet formation and its evolution in time, and selected scalar fields (velocity components, temperature, and vapor concentration), have proved invaluable for a correct understanding of the various phenomena under examination. In particular, the scientific data visualizer, combined with the power of cooperative processing, has allowed a rapid identification of the most significant parameters that need to be tuned in order to recover good agreement between the simulation and the experimental data.

The great challenge facing the automotive and truck industries for the 1990s is the development of new engines able to drastically cut down pollutant emissions without reducing engine power. Diesel engines are a focus of attention, with research still to be done. The combustion process is subject to improvement, as many experimental studies show. The primary point concerns optimization of the injection process in the combustion chamber, involving many parameters such as geometric data for the injector characteristics and the combustion chamber, the injection rate and duration, and the charge air swirl level. In this framework the automotive industry has become one of the most important users of advanced computers and computer programs. Calculations and simulations provide a more detailed insight into complex physical phenomena than is possible to obtain using experimental methods alone. Programs give realistic simulations of the internal phenomena taking place in the combustion chambers of reciprocating engines.[1] But, while computers and graphics make rapid advances, the software available to researchers and engineers has not kept pace, mainly due to gaps in physical knowledge.

The work done at the IBM European Center for Scientific and Engineering Computing (ECSEC) on

the computer program KIVA-II had two principal targets: first, to participate in validating the code, comparing, at least from a qualitative point of view, the numerical simulations with experimental data provided by Renault Vehicules Industriels (RVI), and to improve some subphysical models, such as the evaporation routine; second, thanks to new advanced tools developed at ECSEC, to provide a solution for engine designers and researchers that is based on a highly interactive package and able to work as an on-line monitor tool on the numerically intensive calculations, as well as a classical postprocessor. In addition to this, the possibility to produce an animated film of the simulations is offered.

This advanced environment is based entirely on IBM platforms; the user is given the choice to work on a mainframe alone, on a workstation (RISC System/6000*), or, in the framework of cooperative processing, mainframe to workstation and workstation to workstation.

Software

Outline of KIVA-II. KIVA-II is a computer program developed at Los Alamos National Laboratory (LANL) for the numerical calculations of transient, two- and three-dimensional chemically reactive fluid flows with sprays.[2]

KIVA-II can solve for:

- Laminar or turbulent flows
- Subsonic or supersonic flows
- Single phase or dispersed two-phase flows

Arbitrary numbers of species and chemical reactions are allowed. A stochastic particle method is used to calculate evaporating liquid sprays, including the effects of droplet collisions and aerodynamic breakups.

KIVA-II solves the unsteady equations of motion of a turbulent, chemically reactive mixture of ideal gases, coupled to the equations for a single component vaporizing fuel spray. The gas-phase solution procedure is based on a *finite volume* method called the ALE (Arbitrary Lagrangian-Eulerian) method.[3] The structured mesh can conform to curved boundaries and can move to follow changes in the combustion chamber geometry. As in the original ALE method, each time step is divided into two phases—a Lagrangian phase and a rezone phase. In the Lagrangian phase the vertices move with the fluid velocity, and there is no convection across cell boundaries. In the rezone phase, the flow field is frozen and remapped or rezoned onto the newly computed mesh. In the Lagrangian phase, implicit differencing is used for all the diffusion terms and the terms associated with pressure wave propagation. Explicit methods are used to calculate convection in the rezone phase; since the convection calculation can be subcycled an arbitrary number of times, any time step restriction associated with the Courant stability condition is removed. The mesh generation allows the computational region to include cupped pistons and domed cylinder heads and to offset these relative to the axis of the cylinder.

The original KIVA-II code can provide three different types of plots at run time.

In addition to giving the user the possibility of performing tape dumps of interesting variables for subsequent graphic postprocessing, the original KIVA-II code can provide three different types of plots at run time: zone (grid) and spray, velocity vector, and contour.

Some plots are simply two-dimensional views through predefined planes (e.g., x-z or overhead views); others are perspective plots. In the latter case, the grid and the spray particles are plotted in a perspective view, while fluid velocity vector and contour plots of selected cell variables are presented in a selected plane. The point of view of the perspective plots must be chosen prior to the run, as well as the planes onto which contour plots are made.

Even though these plotting routines cover the basic needs of many users, it is clear that the use of the code for industrial purposes calls for a significant upgrade of the graphic capabilities, as well as enhanced user-friendliness.

The EVAP2 routine. The fuel droplets absorb heat from the surrounding gas, and release mass to it as they evaporate. It is therefore necessary to introduce a physical model, describing the heat-up and the evaporation of the liquid droplets and some relations that account for the gas-liquid interaction.

In the original code, an Infinite-Conductivity Model is employed for the liquid phase. According to this model, the transmission of heat inside the drop is instantaneous (that is to say that the thermal conductivity of the liquid is infinite), so that the temperature field in the spherical droplet is uniform.

The assumptions of this model lead to very simple mathematical equations, but they may become over-simplified when high temperatures and high pressures are encountered. In this case, the effects due to the thermal conductivity must be taken into account, as the heat-up period is a non-negligible part of the droplet lifetime.

The authors changed[4] the physical model of the liquid phase to obtain a better description of the evaporation process, while trying not to add an excessive computational burden to the program.

The spherical symmetry is retained, but the thermal conductivity of the liquid has its own value, deducted from Reference 5, which results in a so-called Finite-Conductivity Model. The energy equation of the liquid is then the well-known one-dimensional, nonsteady Fourier equation in spherical coordinates

$$\frac{\partial T}{\partial t} = \frac{k_f}{\rho_f c_f} \frac{1}{r^2} \frac{\partial}{\partial r}\left(r^2 \frac{\partial T}{\partial r}\right) \tag{1}$$

with the boundary conditions at the center and surface of the droplet

$$r = 0: \quad \frac{\partial T}{\partial r} = 0$$

$$r = r_s: \quad 4\pi r_s^2 h(T_\infty - T_s) = \dot{m}L + 4\pi r_s^2 k_f \left(\frac{\partial T}{\partial r}\right)$$

where h is deducted from the Nusselt number and accounts for the effects of convection due to the relative motion.

The most important effect of the new model is to generate a temperature distribution along the drop radius: time evolution of this distribution is followed integrating Equation 1 in a purely explicit fashion. Because of this, a new restriction on Δt, based on numerical stability, has been introduced. This restriction on Δt, however, affects only the new routine, EVAP2 and depends on the value of $(\Delta r)^2$ (the distance between two grid points). To prevent Δr (and, consequently, the maximum Δt allowed) from becoming too small, the number of points on the grid is decreased accordingly to the progressive reduction of the droplet radius, so that Δr is, more or less, constant.

In addition to this new evaporation routine, the original RINPUT routine of KIVA-II has been modified to take into account the effective mean area of the injector holes. The area parameter allows the user to simulate more accurately the behavior of the recent electronic pump injector systems. In these new devices, in order to optimize the in-cylinder combustion, a microprocessor controls several parameters of the injection process, such as timing, duration, and law of injection, according to feedback quantities, in order to optimize the in-cylinder combustion.

The computing environment

Thanks to an advanced scientific data visualizer, called 4D-SDV and developed at ECSEC,[6] it is now possible to have a visual appreciation of the complex phenomena simulated by the KIVA-II code. Moreover, a physical submodel causing the original program to fail may be easily recognized. There are currently two ways of working within this new environment: one is classical postprocessing, the other is a more powerful approach, based on the concept of *cooperative processing*. In the latter case, using a tool developed at ECSEC—COOPPROC[7]—provides the user with complete control over the running simulation. While the graphic output is used to report the course of the simulation, it is possible to change some crucial parameters while running the problem, in order to finely tune the simulation run.

This system is completed by a video server that is able to produce animations of the most significant runs.

Postprocessing. The purpose of the graphic postprocessing environment is to visualize static data, i.e., data residing on a mass storage, produced by KIVA-II running on a mainframe under Multiple Virtual Storage (MVS), virtual machine (VM), or Advanced Interactive Executive/370* (AIX/370*) operating systems.

A static version of the 4D-SDV tool can be used for the visualization, either on a graphic workstation directly attached to the mainframe (IBM 6090 or IBM 5080), or on a RISC System/6000 workstation.

In the latter case, the data are transferred via TCP/IP (Transmission Control Protocol/Internet Protocol) network protocol.

Mainframe environment: MVS. The output data are written onto a disk file (namely a sequential MVS data set) at selected time steps by a set of new output routines that have been added to the original code.

The following are typical data:

- The fully-deformable three-dimensional computational grid
- The velocity magnitude field
- The in-cylinder temperature field
- The fuel vapor concentration field
- The spray droplets (coordinates, radii, surface temperatures)

Note that the third and fourth items are interpolated to the vertices of the computational grid starting from the values at cell centers, where they are originally defined. To this purpose, an interpolating routine (interp) has been explicitly added to the original code.

The maximum number of the fields that can be displayed is currently set to 20.

The new subroutines are:

- wdata: Allows the user to write out data for later visualization. Its calling frequency can be controlled by the standard flags present in the input file (NCFILM, CAFILM, TWFILM).
- wgeom: Writes the cartesian coordinates of the grid vertices
- wvelo: Writes the velocity modulus field
- wtemp: Writes the temperature field
- wconc: Writes the fuel vapor mass fraction field, as well as the different gaseous species mass fraction fields
- wdrop: Writes the spray droplets coordinates, radius, and temperature

These routines also convert data from EBCDIC to ASCII format.

Workstation environment. Four programs allow the user to visualize the results of the KIVA-II run:

1. getfile: A script that performs automatic file transfer program (ftp) to MVS in order to get
   - The unformatted data file
   - The *include* file containing dimension parameters
2. prepost: A FORTRAN program that reads the unformatted data file coming from MVS and produces several files, in a format suitable for 4D-SDV

   In the previous example, the files could be
   - kiva*nn*.xyz, the grid corresponding to frame *nn*
   - velo.kiva*nn*, the velocity defined on the grid *nn*
   - temp.kiva*nn*, the temperature
   - conc.kiva*nn*, the vapor concentration
   - drop.kiva*nn*, the droplets at frame *nn*
3. setprof: A FORTRAN program that, given one of the names generated by prepost for the grid data file (e.g., kiva*n*, *n* being the sequential number of the frame to be visualized), generates the appropriate kiva.profile file for 4D-SDV
4. 4D-SDV: The visualization program

In order to visualize the data frames, the user should execute getfile and prepost in sequence.

At this point an appropriate profile for the 4D-SDV is needed. This file, e.g., kiva.profile, is automatically generated by the program setprof. The user is then requested to enter the name of the grid data file at a selected frame (e.g., kiva3 for the third frame), as well as the grid dimensions. This information is contained in the file INFORUN, which is also prepared by prepost. At this point, upon invocation of 4D-SDV, the visualization of

the data pertaining to the selected frame will result.

These pictures yield an immediate appreciation of the global heterogeneous environment where the vaporization takes place, as discussed in the following text. Combined with the analysis of the evolution of some integral quantities over the whole combustion chamber, the static visualization is a very valuable tool to identify the local phenomena that should be modified or suppressed to proceed with the optimization.

Cooperative processing

In a cooperative processing environment, functionally distinct processes, typically numerical simulation and data visualization, are assigned to different computing devices. The paradigm upon which this environment is based is the client/server model. Client applications request services from a server process.

Within this framework, the results computed by the "number cruncher" device (client) running KIVA-II can be continuously displayed on an advanced graphic workstation (server). In our case, the client application is running on an Enterprise System/3090* (ES/3090*) Vector Facility under the AIX/370 operating system (but MVS or VM are equally available), while the server application is running on a RISC System/6000 workstation. KIVA-II exchanges data and information with the visualization program by means of the communication library supplied by the COOPPROC tool. This library contains a series of C routines that are invoked at run time by both client and server source codes. All the interprocess communications are handled by two processes running in the background.

In order to put the COOPPROC tool in operation, a number of files are required on both client and server sides.

On the client side, the working directory contains two files to be customized by the user:

- whichws: Contains the network address of the machine to be connected and the name of the running application
- inputpara: Contains the parameters that can be changed at run time in the client application

On the server side, the following files are required:

- whichhost: Contains the network address of the machine to be connected and the name of the running application
- inputpara: Contains the parameters that can be changed in the client application KIVA-II and a flag indicating if conversion between 370 and IEE488 floating-point representation mode is required
- kiva.profile: Contains the profile needed by 4D-SDV and information on the number and type of fields to be displayed
- kiva.input (optional): A command file that sets several options in the graphic tool, such as lights (type and positions), roto-translations, etc.

During a typical working session, the user opens a window on the host machine (client) and runs both KIVA-II and the visualization program (server).

According to a flag present in both the files whichhost and whichws, two modes of operation are available. In one case, the two applications are launched at the same time in such a way that the time evolution of KIVA-II simulation is followed from the beginning.

Alternatively, the client is started and the server can be dynamically attached or detached at any subsequent time, giving the user the possibility to monitor the client application now and then, leaving the workstation free. This is useful particularly during the most intensive runs.

Figure 1 refers to a typical working session on a RISC System/6000. On the top left the client window is shown, which reports the KIVA-II transmission activity. On the bottom right is the window displaying the server activity. On the bottom left is a widget (*feedback*), which allows one to select the control parameter (transmission frequency, frames retrieving. . .). In the top center is the physical output; in this case it represents the RVI case geometry, with the cupped piston (yellow) and the injector (red). The other windows are the soft dials controlling the 4D-SDV activity. Special attention is to be devoted to the window at the left of the graphic output. It allows the user to change interactively the color table to

Figure 1 A cooperative processing working session

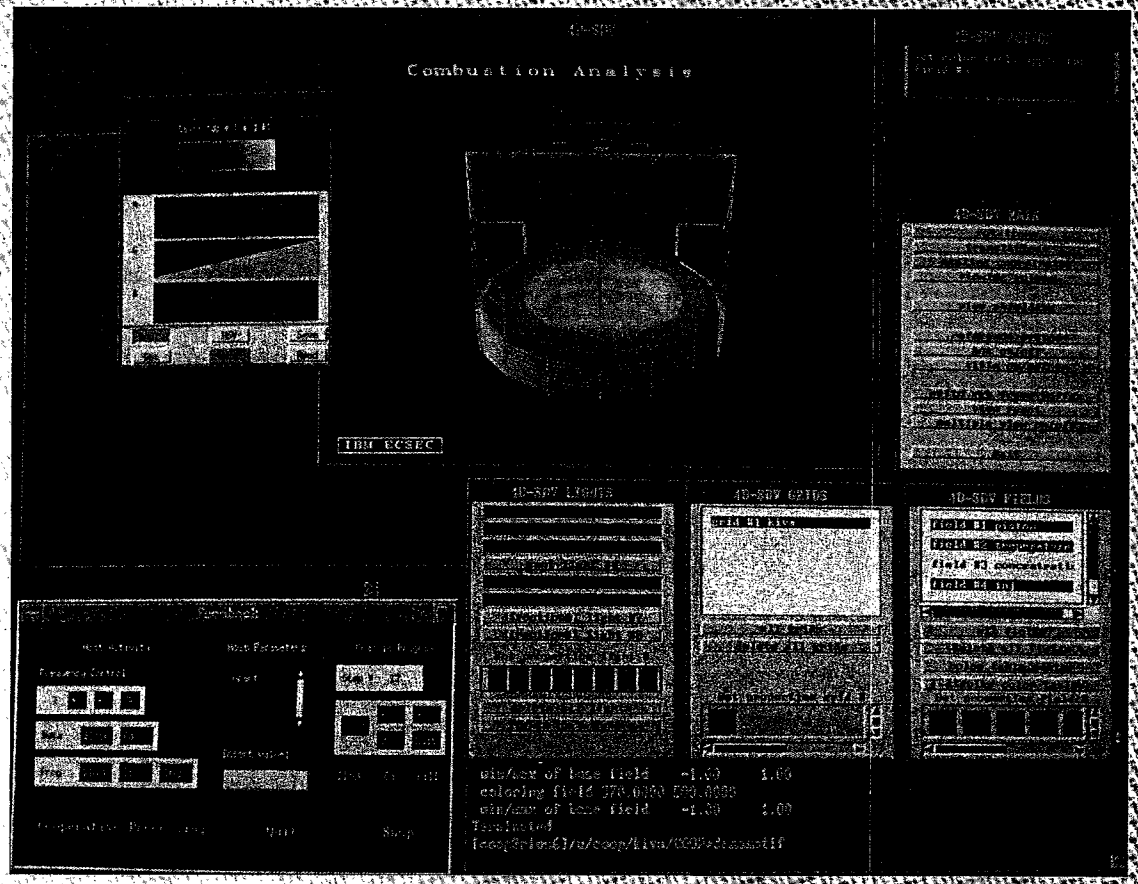
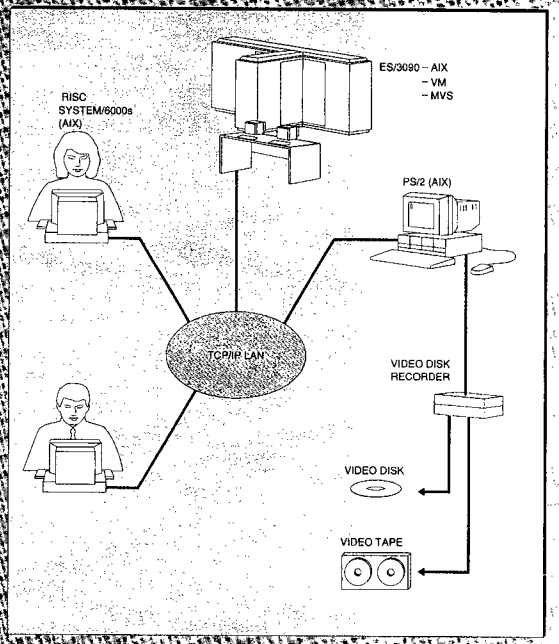


Figure 2 The computing environment



LIFT, PLEASE

Figure 3 Case A: comparison between computed and measured fuel volumetric flow during injection

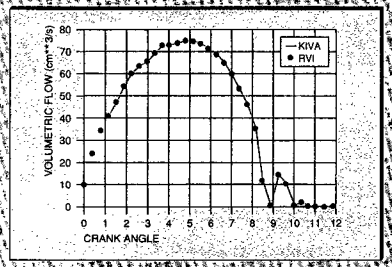


Figure 4 Case A: comparison between computed and measured average pressure near top dead center (TDC)

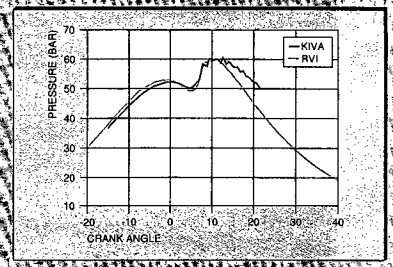


Figure 5 Cases A and B: computed NO mass near TDC

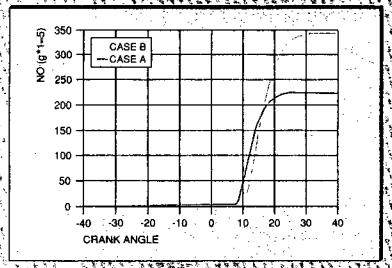


Figure 6 Cases A and B: computed average temperature near TDC

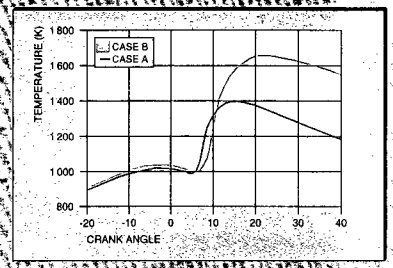


Figure 7 Case C: comparison between computed and measured fuel volumetric flow during injection

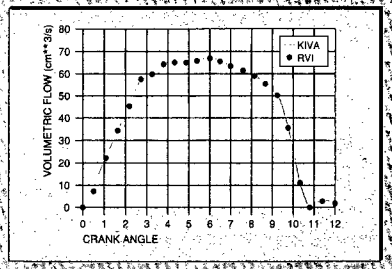


Figure 8 Case C: comparison between computed and measured average pressure near TDC

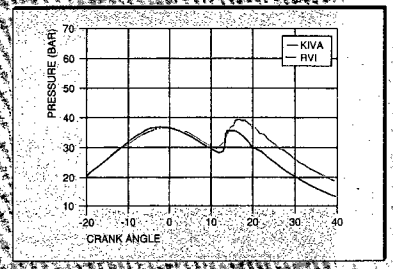


Figure 9 Fuel droplets in the combustion chamber

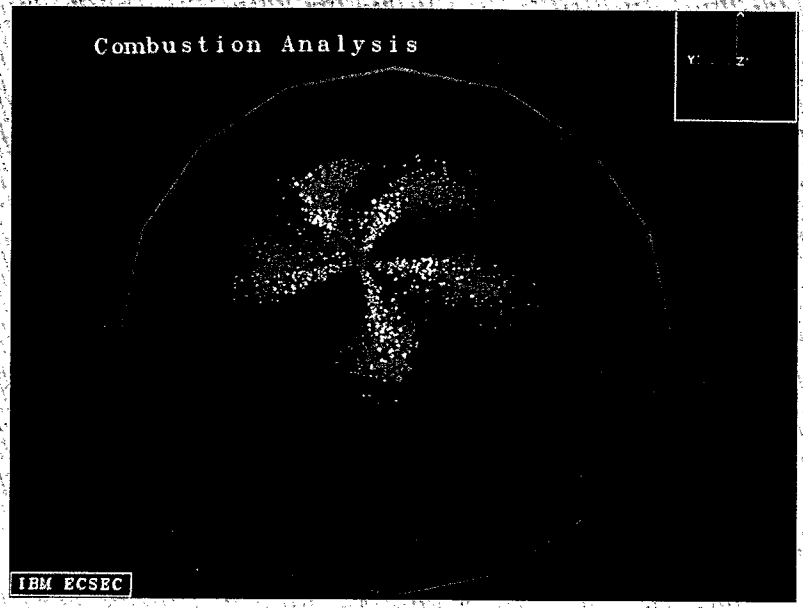


Figure 10 Slicing on the temperature field

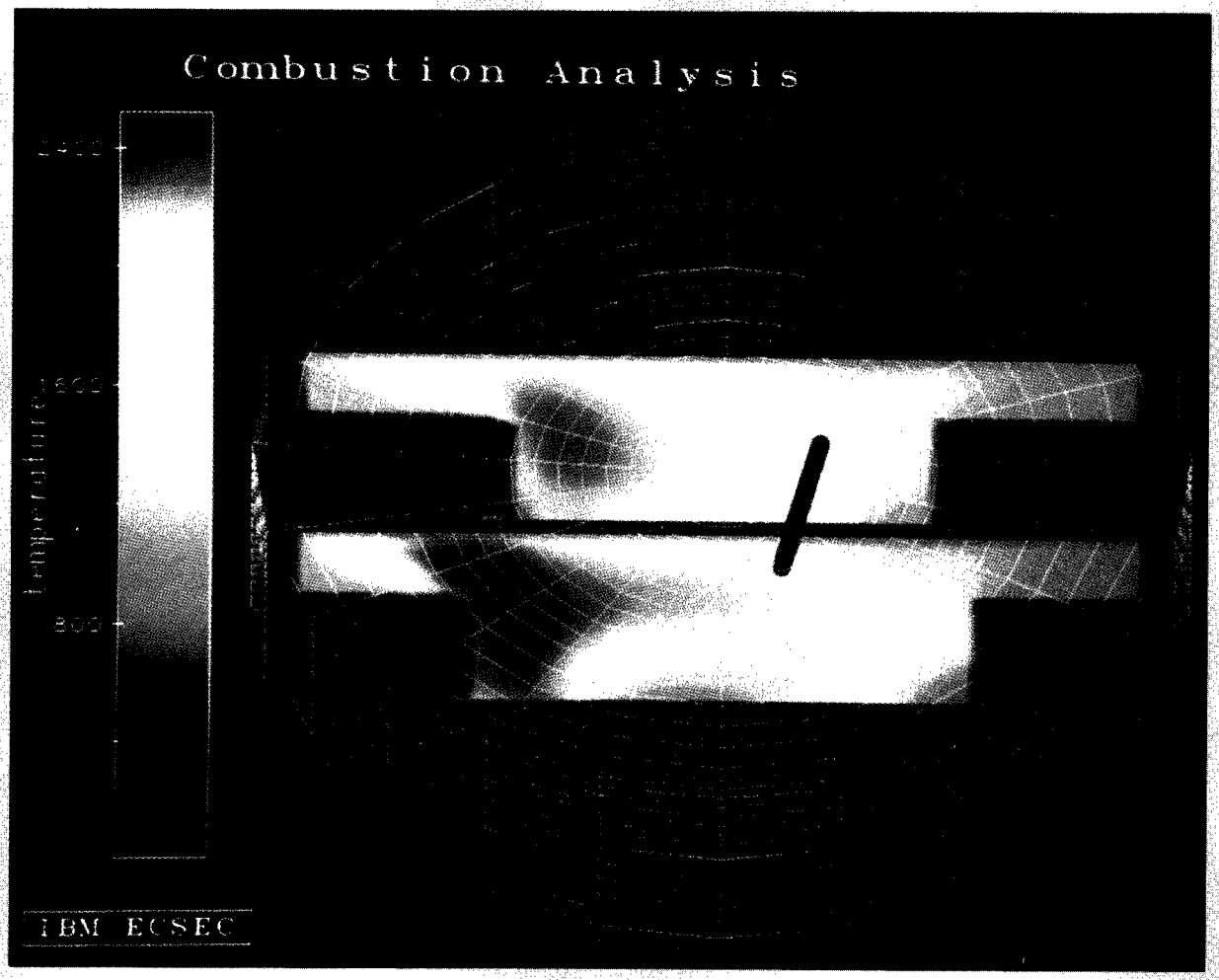
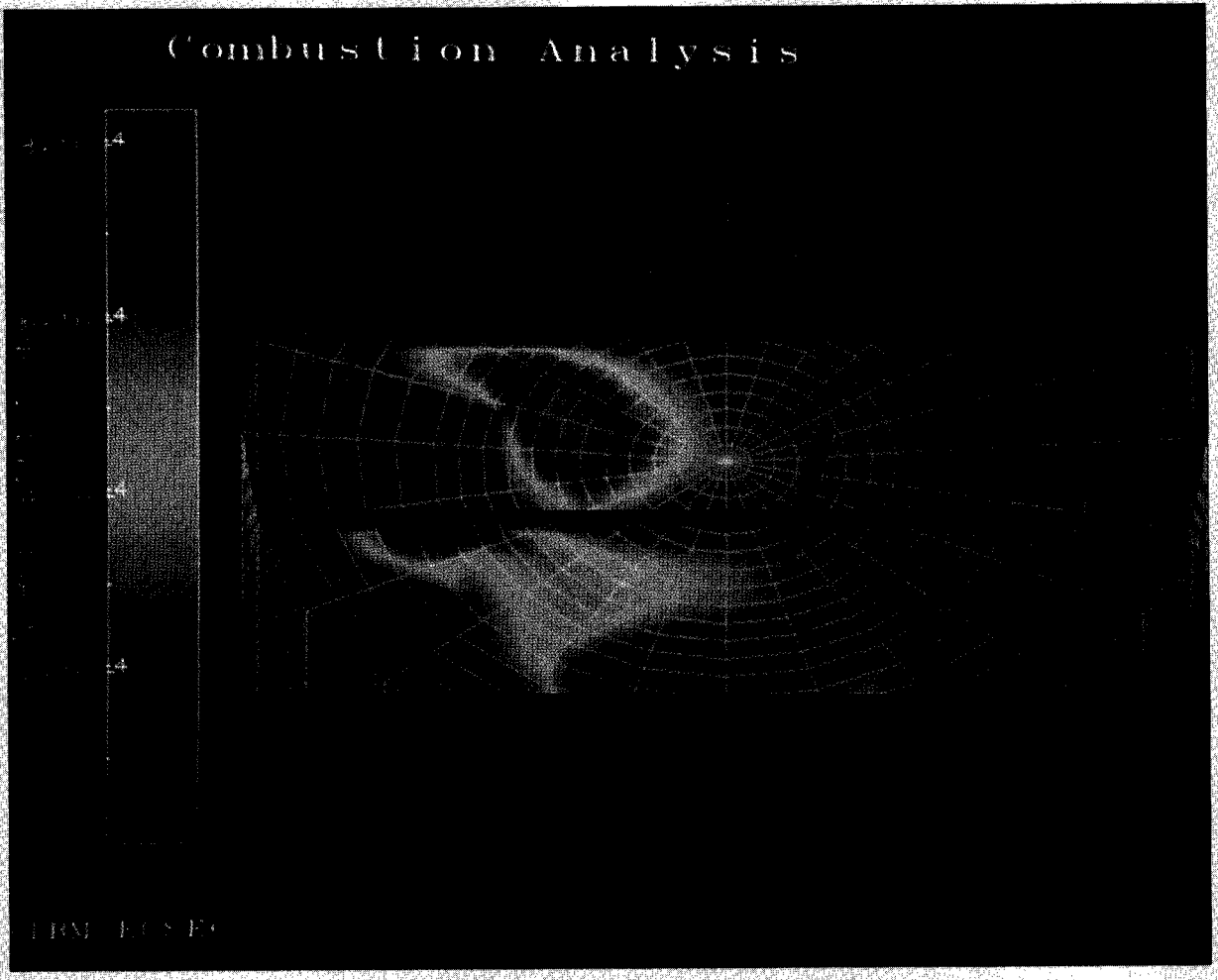


Figure 11 Slicing on the NO mass fraction field



be applied to a given field and to save/retrieve edited tables.

The user can perform a series of actions such as the following:

- Powerful graphic manipulations, such as roto-translations, clipping, slicing, use of light sources, etc.
- Freezing-resuming the KIVA-II run and/or the graphic program
- Changing previously selected parameters of KIVA-II (steering)
- Varying the frames frequency (in terms of cycles)
- Retrieving a number of previous frames (time history)

The video server. Videotapes are a new way of communicating the results of scientific and technical simulations. But professional video equipment is expensive, and recording using videotapes is a sequential procedure. Availability of direct-access writable optical laser videodisks, together with a multiuser video server is the solution. An easy-to-use video server for the production of video animations and archival of images has been developed at ECSEC.[8] With this tool, the results of some simulations running on the mainframe under MVS are immediately turned into animations. To this purpose a high-performance graphic workstation, such as the IBM 6090, had to be attached to the mainframe. During the KIVA-II run, some calls to 4D-SDV with the option "write rgb frame buffer" resulted in a request to the video server to record selected frames. The data were transferred through the local area network (LAN) to the Personal System/2* (PS/2*) driving the videodisks and running the video server itself. At a later time, animations were enriched with caption panels and transferred to a common videotape, with different playing speeds.

Hardware

The high performance of this computing environment takes advantage of the variety of IBM machines available. The complete scheme is shown in Figure 2 and consists of a mainframe (IBM ES/3090 VF), several RISC System/6000 machines and a PS/2. All of them are connected via a TCP/IP LAN. On the mainframe operating system, independence (AIX, MVS) is guaranteed. The mainframe acts as a *client*, while the RISC machines can be operated as *servers* or *clients* as well.

Special care was taken to have the advanced combustion analysis computer environment running on a wide spectrum of hardware configurations—from the minimal one to the most complete one.

Results

Much effort has been spent during the simulations for the purpose of tuning the KIVA-II on a specific problem, i.e, to reproduce the behavior of an experimental engine.

The engines used by the research laboratories for visualization purposes have a transparent quartz window for the optical inspection, commonly placed on the top of the combustion chamber or in the place of a flat piston bowl. In a reciprocating engine, this transparent access window would soon become dirty with the deposits from the combustion products and lubricant. To avoid this problem, no lubricating oil is used, but the engine is operated for only a few cycles in each experiment. As a result, the piston and the cylinder head and walls have an initial temperature from preheating, i.e., the engine is a "tepid" engine.

The RVI experimental data referred to ten different situations, in which the engine was operated at different speeds (revolutions per minute, or rpm), with different injection laws (in terms of timing and duration of the injection, velocity history, and amount of injected fuel). The data consisted of a series of physical measured quantities, such as average pressure inside the combustion chamber, and a series of derived quantities, such as the injection velocity law. A film showed the injection and the subsequent combustion processes taking place in the cylinder by means of a high-speed cinematography technique, over 5000 frames per second. Some photographs derived from the film frames were also made available by RVI.

Overall comparison with experimental data. The first step taken to reproduce the experimental data was to prepare a suitable input file for KIVA-II. The real piston geometry was implemented along with the coordinate and the orientation of a single multihole injector. The piston had a cylindrical bowl, slightly offset along the x-direction. The injector had five nozzles and was inclined with

respect to the z-axis. The problem has dimensions $17 \times 24 \times 24$.

The strategy adopted in order to test the behavior of the KIVA-II code under real conditions was to start from a standard direct injection (DI) diesel engine test case and to move step-by-step toward the experimental conditions of the RVI engine.

Attention was focused on one of the ten RVI experiments as a first target (case *A*); in this experiment the engine was run at 1200 rpm. Several simulations were performed both on the mainframe and on the RISC System/6000 workstation.

Much emphasis was put on the reproduction of the real injection law. The code provides a fixed-area nozzle injector model. For this case, subroutine RINPUT calculates the total fuel mass predicted by the velocity table (given in the input file) using the sums of the nozzle areas, and corrects the velocity profile up or down by the ratio of mass desired to mass predicted. In addition to this, the effective mean areas provided by RVI were also used to construct an area table similar to the velocity one, the assumption being that all the nozzles behaved in the same way.

This model seems to fit the experimental data very well, as shown in Figure 3.

The most significant break-through in the reproduction of the experiment was achieved during a cooperative processing session in which we noticed that the simulated fuel jets closely resembled the real ones (as seen in the RVI videotape) if the *breakup* submodel was turned on some time after the injection beginning. The breakup is a physical mechanism by means of which, under certain conditions, the fuel droplets are split in smaller droplets. This influences the evaporation stage and thus the entire combustion process. The delay was calculated from the Hiroyasu[9] formula and the parameter in KIVA-II controlling the breakup subroutine was changed at run time.

A qualitative comparison between the experiment and the calculation can be made by plotting the average pressure inside the engine vs the crank angle, assumed to be zero at top dead center (TDC), that is at the end of the compression stage. Figure 4 shows how the code behaves quite well until the last stage of the combustion phase. The divergence between the two curves is probably due to a loss of energy caused by a poor modeling of the heat exchange process during the compression and expansion strokes, but also mainly by the turbulent combustion model used in this initial KIVA-II version.

The physical and numerical parameters tuned for case *A* were also used for a similar case at 1200 rpm, which had different initial conditions in terms of pressure and temperature and a different injection law (case *B*).

From the designer point of view, the distribution of a pollutant such as *NO* during and after the combustion stage is crucial information, and it is not easy to obtain from visualizing experiments. Figure 5 shows the computed *NO* mass for cases *A* and *B*. The greater amount of *NO* produced in case *B* is due to the higher temperature reached during the combustion, in this case (Figure 6).

A different simulation at 1800 rpm (case *C*) is also presented. Figure 7 shows the computed and experimental volumetric flows, while Figure 8 shows the average pressure curves. The same discrepancies as cases *A* and *B* during the combustion stage are also encountered.

Visual inspection. The power of the scientific data visualizer allows one to follow the initial setup of the simulations. In fact, one could test almost instantaneously the correct position of the injector with respect to the cylinder geometry and, above all, the correct disposition of the five holes, which are given in the KIVA-II input file in the uneasy form of angles.

A comparison between experimental photographs and color hard copies of the workstation screen during the calculations is not so straightforward. The high-speed photos show the evolution of the flame during the combustion, while we use the computed scalar fields (i.e., temperature, species mass fraction, etc.) for visualization. Nevertheless a different kind of information can be extracted from the computer graphics. In fact the vapor fuel concentration field, i.e., the vaporized fuel ready to burn, can be displayed as well as CO_2, NO, CO, etc. species mass fraction and can be correlated with the displays of turbulent kinetic energy and temperature in the gas phase.

As an example, several fuel droplets are displayed in a top view of the combustion chamber

along with the injector, about 5° crank angle after injection (Figure 9). The droplets are colored according to their surface temperature; the swirl motion inside the piston bowl is visible as well.

An interesting comparison can be made between different scalar fields sliced according to the same plane. For instance, Figure 10 shows the temperature field sliced in the y-z plane (at 20° crank angle after TDC), while Figure 11 refers to the *NO* mass fraction field, sliced in the same plane. It is apparent that the concentration of the pollutant *NO* is maximum where the temperature is also maximum.

Animations. The static two- or three-dimensional display is a first reduction of data that permits one to identify the critical zones and to perform quantitative analysis. The animation extends the data synthesis in four dimensions. Some accuracy is lost, but the user gains the physical feeling of the three-dimensional transient phenomena. The animations of the droplets evolution can be directly compared to the high-speed film.

Several animations of the most effective runs were generated in the environment shown in Figure 2. The flexibility of this system allowed the rotation of the moving piston and the display of different iso-surfaces, also making use of the transparency feature.

The animations showed how bad the original spray/wall interaction submodel is. In fact, as spray droplets impinge onto the piston face or the cylinder wall, they stick on it and evaporate from there. Since in the RVI experiments the injector is set off the z-axis, one could observe a major number of droplets accumulating on the piston bowl internal face nearest to the injector. As a result, much of the evaporating fuel is seen to be localized near the wall, instead of being equally spaced in the combustion chamber. This could be another reason for the poor modeling of the combustion stage.

Future perspectives

Physical submodeling: spray dynamics. The efficiency of the diesel combustion process is strongly influenced by the spray dynamics. This field is still the object of much research work, since the atomization mechanism is not very well understood.

Among other phenomena, such as the spray dispersion and initial atomization, the ignition delay, and the soot particle formation, the interaction of spray with walls is very poorly modeled in KIVA-II. The spray–wall interaction is of major interest to predict the global combustion efficiency, the hydrocarbons contained in the exhaust gases, and therefore the "pressure vs time" curves.

The first target is the spray–wall interaction submodel in the absence of combustion. Depending on the value of a physical parameter (nondimensional Weber number), the model should contain either the formation of a liquid film on the wall, which could evaporate, or a droplet bounce associated with fragmentation.

In the framework of a collaboration with the Istituto Motori of the National Council of Research (CNR) in Naples, experimental data on diesel spray dynamics will be used to validate the new models.

Further steps will be made in the study of influence of gas density, ambient temperature, and pressure on the jet dispersion.

Distributed processing. There is a wide consensus in the users' community that high-performance scientific and technical computing could be enhanced by a physically distributed (albeit perhaps logically-shared) memory.

Among the various possibilities currently being explored, an architecture that looks very appealing from the point of view of cost-effectiveness is the one based upon a distributed *cluster* of superscalar RISC workstations. Within this architecture, each machine takes care of a given subportion of the global problem and exchanges data with other computational nodes through a physical interconnect via message-passing techniques.

Crucial to the success of such an architecture is the availability of fast interconnects able to minimize the waste of useful CPU cycles during the data transmission.

Taking advantage of the structured mesh used by KIVA-II, preliminary studies are being made at IBM

ECSEC in order to improve the computational efficiency of the code on such a distributed architecture.

Acknowledgments

We kindly acknowledge M. Bernaschi and G. Richelli of IBM ECSEC for their collaboration in customizing the cooperative processing tool to KIVA-II. We are also indebted to M. Briscolini, P. Santangelo, and F. Valentini of IBM ECSEC, whose work allowed us to interface KIVA-II with the scientific visualization tool. Finally, our thanks go to G. Chiatti of Rome University "La Sapienza" Dipartimento di Meccanica e Aeronautica, for the work done on the new vaporization model.

*Trademark or registered trademark of International Business Machines Corporation.

Cited references

1. P. Pinchon, "Modelling of Fluid Dynamics and Combustion in Piston Engines," *Proceedings of the International Symposium COMODIA*, Kyoto (September 3–5, 1990).
2. A. A. Amsden, P. J. O'Rourke, and T. D. Butler, *KIVA-II: A Computer Program for Chemically Reactive Flows with Sprays*, Los Alamos National Laboratory Report LA-11560-MS (May 1989).
3. C. W. Hirt, A. A. Amsden, and J. L. Cook, "An Arbitrary Lagrangian-Eulerian Computing Method for All Flow Speeds," *Journal of Computational Physics* **14**, 227–253 (1974).
4. G. Chiatti, S. Golini, M. Maggiore, F. Papetti, and S. Succi, "Improving the Vaporization Model of KIVA-II in an Advanced Computing Environment," submitted to *International Journal of Heat and Mass Transfer* (1992).
5. J. B. Maxwell, *Data Book on Hydrocarbons*, Robert E. Krieger Publishing Co., Huntington, New York (1977), p. 213.
6. M. Briscolini, P. Santangelo, and F. Valentini, *4D-SDV: A Short User's Guide*, internal report IBM ECSEC (July 1991).
7. M. Bernaschi and G. Richelli, *COOPPROC: A Tool for Cooperative Processing*, internal report IBM ECSEC (July 1991).
8. M. Briscolini and P. Santangelo, "Animation of Computer Simulations of Two-Dimensional Turbulence and Three-Dimensional Flows," *IBM Journal of Research and Development* **35**, No. 1/2, 119–139 (January/March 1991).
9. H. Hiroyasu, T. Kadota, and M. Arai, "Supplementary Comments: Fuel Spray Characterization in Diesel Engines," *Combustion Modelling in Reciprocating Engines*, J. N. Mattavi and C. A. Amann, Editors, Plenum Press (1980), pp. 369–408.

Accepted for publication July 17, 1992.

Francesco Papetti *IBM European Center for Scientific and Engineering Computing, viale Oceano Pacifico 173, 00144 Rome, Italy (electronic mail: papetti@romvmnic.vnet.ibm.com).* Dr. Papetti is a system engineer in the Engineering Applications group at the European Center for Scientific and Engineering Computing (ECSEC). He received his degree in physics in 1988. In 1989-1990 he was at the Laser Technologies Laboratory of ENEA Energy Research Center (Frascati-Rome), where he worked on the realization of an infrared laser sensor and a laser range finder, both dedicated to robotics applications in hostile environments. At the end of 1990 he joined IBM ECSEC, where he works in fluid dynamics research projects, namely numerical combustion. Since April 1991 he has been involved in the joint project with Renault Vehicules Industriels (RVI) on diesel engine combustion analysis. He is currently working on modeling the dynamics of diesel sprays impacting on hot surfaces and subsequent heat-transfer phenomena, in collaboration with the Istituto Motori of the National Council of Research in Naples. He has authored more than 20 papers in the fields of lasers, optics, and computational fluid dynamics.

Stefano Golini *Dipartimento di Meccanica e Aeronautica, Universita' degli Studi di Roma "La Sapienza," via Eudossiana 18, 00184 Rome, Italy.* Dr. Golini is currently a post-doctoral Fellow at the University of Rome "La Sapienza," where he received his degree in mechanical engineering in July 1990. His thesis research was the simulation of a direct injection gasoline engine. From April 1991 to March 1992 he was a visiting researcher at IBM ECSEC, where he studied problems concerning the vaporization of fuel drops and was involved in the joint project with Renault Vehicules Industriels (RVI) to test the KIVA-II code on an experimental engine. From March 1992 to September 1992 he has been at the Istituto Motori of the National Council of Research in Naples, where he has studied the propagation of front flame in spark ignition engines.

Marco Maggiore *IBM European Center for Scientific and Engineering Computing, viale Oceano Pacifico 173, 00144 Rome, Italy.* Mr. Maggiore received his degree in mechanical engineering from the University of Rome in 1985. In 1986 he was at the Department of Electronical Engineering at the same university, where he developed methods to describe closed chains spatial mechanisms used by robots. In the same year he joined the IBM Scientific Center of Rome, where he worked on a project involving an artificial intelligence system for automatic knowledge acquisition. In the following year he moved to IBM ECSEC, where he is a researcher in the field of computational fluid dynamics. He has authored about 20 papers in the field of mechanics and computational fluid dynamics.

Sauro Succi *IBM European Center for Scientific and Engineering Computing, viale Oceano Pacifico 173, 00144 Rome, Italy.* Dr. Succi received his doctorate in nuclear engineering from the University of Bologna. Subsequently he spent a year in the Department of Fast Breeder Reactor of the National Committee of Nuclear Energy (CNEN) in Bologna, and a year at the Max Planck Institut für Plasmaphysik in Garching, Germany, working on Montecarlo simulation codes for thermonuclear plasmas. Subsequently he moved to the Plasma Physics Center of the Ecole Polytechnique Federale in Lausanne, Switzerland, from which he obtained a Ph.D. working on finite-element codes for plasma turbulence. In 1986 he joined IBM ECSEC, where he is currently leading a number of

projects in the area of computational fluid dynamics on high-performance IBM computing platforms.

Philippe Gaillard *Renault Vehicules Industriels, Engine Research and Development Division, DER 1, 69800 Saint-Priest, France.* Dr. Gaillard is a research engineer in the truck diesel engine field. He was graduated as an engineer in energetics from the Institut National des Sciences Appliquees of Lyon in 1975. He started his career in the development of three-dimensional computational fluid dynamics codes devoted to predict the diffusion of pollutants in the atmosphere. He received his Doctor-Engineer diploma from the Claude Bernard University of Lyon in 1978. During his one-year military service, he taught aerothermodynamics at the Ecole de l'Air of Salon de Provence. In 1979, he joined the Ecole Centrale de Lyon as an assistant professor in fluid mechanics and thermodynamics. In 1983 he joined Renault Vehicules Industriels and initiated computational fluid dynamics codes development for industrial engine applications. From 1984 to 1989 he was the Renault representative at Engine Modeling Consortia such as the Joint Research Committee and the Groupement Scientifique Moteurs. He is now in charge of research on engine cooling, engine breathing and supercharging, and combustion modeling.

Jean-Michel Perez *Volvo Truck Corporation, Power Train Division, Department 26644 BC3, S-40508 Gothenburg, Sweden.* Dr. Perez is a research engineer in the truck diesel engine field. He received a Master of Science degree from Tolouse University in 1986 and a post-graduate diploma in physics/energetics from the Polytechnic Institute of Grenoble in 1987. He started his career in the heat transfer department of the Nuclear Research Center of Grenoble, where he worked on anisotherm diffusion of fine particles, with an application to fouling of heat exchangers. He received his doctor's degree from the Institut National des Sciences Appliquees of Lyon in 1990. The same year, he joined Renault Vehicules Industriels (RVI) to work on advanced engine research. In 1992 he joined Volvo Truck Corporation (VTC) as part of the RVI/VTC engineering exchange program. Today, he is involved in engine emission projects.

Reprint Order No. G321-5497.