ANL/RAE/CTR-03-1

Modeling and Analysis of Transient Vehicle Underhood Thermo-Hydrodynamic Events Using Computational Fluid Dynamics and High Performance Computing

by Paul Froehle, Adrian Tentner, Carey Wang

ARGONNE NATIONAL LABORATORY
TRANSPORTATION
TECHNOLOGY R&D CENTER

Operated by The University of Chicago,
under Contract W-31-109-Eng-38, of the

United States Department of Energy

February 2003

Argonne National Laboratory, with facilities in the states of Illinois and Idaho, is owned by the United States Government and operated by The University of Chicago under the provisions of a contract with the Department of Energy.

Disclaimer

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor The University of Chicago, nor any of their employees or officers, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of document authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof, Argonne National Laboratory, or The University of Chicago.

Available electronically at http://www.doe.gov/bridge

Available of a process fee to U.S. Department of
Energy and its contractors, in paper, from:

U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831-0062
Phone: (865) 576-8401
Fax: (865)576-5728
Email: reports@adonis.osti.gov

Modeling and Analysis of Transient Vehicle Underhood Thermo-Hydrodynamic Events using Computational Fluid Dynamics and High Performance Computing

Paul Froehle, Adrian Tentner, Carey Wang

Transportation Technology R&D Center
Argonne National Laboratory

February 2003

Table of Contents

Page

Modeling and Analysis of Transient Vehicle Underhood Thermo-Hydrodynamic Events using Computational Fluid Dynamics and High Performance Computing

Paul Froehle, Adrian Tentner, Carey Wang

1. Introduction

This work has explored the preliminary design of a Computational Fluid Dynamics (CFD) tool for the analysis of transient vehicle underhood thermo-hydrodynamic events using high performance computing platforms. The goal of this tool will be to extend the capabilities of an existing established CFD code, STAR-CD, allowing the car manufacturers to analyze the impact of transient operational events on the underhood thermal management by exploiting the computational efficiency of modern high performance computing systems.

In particular, the project has focused on the CFD modeling of the radiator behavior during a specified transient. The 3-D radiator calculations were performed using STAR-CD, which can perform both steady-state and transient calculations, on the cluster computer available at ANL in the Nuclear Engineering Division. Specified transient boundary conditions, based on experimental data provided by Adapco and DaimlerChrysler were used. The possibility of using STAR-CD in a transient mode for the entire period of time analyzed has been compared with other strategies which involve the use of STAR-CD in a steady-state mode at specified time intervals, while transient heat transfer calculations would be performed for the rest of the time. The results of these calculations have been compared with the experimental data provided by Adapco / DaimlerChrysler and recommendations for future development of an optimal strategy for the CFD modeling of transient thermo-hydrodynamic events have been made. The results of this work open the way for the development of a CFD tool for the transient analysis of underhood thermo-hydrodynamic events, which will allow the integrated transient thermal analysis of the entire cooling system, including both the engine block and the radiator, on high performance computing systems.

2. Overview of the Scientific/Technical Approach

The key objective of this work is to develop a CFD - based analytical methodology that can efficiently and accurately predict the transient thermal response of an automotive cooling system and underhood thermal environment under severe operating conditions such as vehicle acceleration while climbing a long steep slope. Such an analytical capability does not presently exist and automotive designers are required to rely on expensive and time-consuming experimental programs to assess the vehicle underhood conditions during transients. The CFD code selected for the development of the new transient computational capabilities is STAR-CD, a leading CFD code widely used in the automotive industry worldwide, which can model complex thermal-hydraulic phenomena on parallel computing platforms. The computing platform used is a cluster computer with 60 nodes, using the Linux Operating System.

To evaluate the accuracy of STAR-CD calculations and validate the new computational methods developed during this work we relied on experimental data obtained during transient radiator experiments performed by DaimlerChrysler. The experimental data was provided by Adapco and DaimlerChrysler. The experimental setup, radiator geometry, and transient experiments performed are described in Section 3. Several selected transient experiments were analyzed using the STAR-CD code. For this analysis we used a STAR-CD computational radiator model developed initially at DaimlerChrysler. The computational radiator model used in this work is described in Section 4.

Although modern computational fluid dynamics (CFD) codes such as STAR-CD have the capability to solve the transient coupled thermal-fluids problem associated with underhood air flow, coolant flow through the radiator and engine block, and conjugate heat transfer within the engine, the computational requirements for solving such a coupled problem with millions of computational zones during transient events that can span hundreds and even thousands of seconds is prohibitive, even with today's massively parallel high performance computing systems. An assessment of the computational time required to model transient events using the standard STAR-CD transient capabilities was performed during this work and the results are presented in Section 5.

From an analysis perspective, however, it may not be necessary to solve this complex system in a tightly coupled mode for transient thermo-hydrodynamic events. The thermal-hydrodynamic assessment of air, coolant and engine block/component temperatures requires a coupling between interacting phenomena through boundary conditions. The response times of the hydrodynamic phenomena are significantly shorter than the thermal response times. This suggests that the integrated system may be viewed as a set of hydrodynamic and thermal transient phenomena that can be analyzed independently over appropriate time intervals and coupled only at selected times through appropriate boundary conditions. The key systems and phenomena being addressed in this proposal are the underhood air flow and the radiator performance during an engine coolant temperature transient. Two methods were developed which allow the computationally efficient modeling of the coupled thermal-hydrodynamic processes which determine the radiator behavior during transients: a) a combination of transient temperature calculations assuming fixed air and coolant velocities and periodic steady state calculations which adjust the velocity field to reflect current conditions, and b) a transient temperature calculation which assumes that the coolant and air velocities change in magnitude during the transient, but remain similar to the velocity fields calculated during the last steady state calculation. A number of new models and capabilities, necessary for the correct modeling of transient thermal-hydraulic phenomena have also been developed. These include the heat exchanger capability to accurately handle variable coolant flow rate situations using measured radiator performance curves, and the ability of the heat exchanger model to calculate the metal temperature in each computational cell and account for the thermal inertia effect of the radiator mass during transients. These methods and models mentioned above are described in Section 6.

The methods and models developed during this work have been tested and validated through comparison with the experimental data described in Section 3. The results of these experiment analyses are described in Section 7. The software implementation of the new methods and models is described in Section 8. Recommendations for further development and validation of the methods and models developed in this pilot project are presented in Section 9.

3. Experiment Description

The experimental data used to validate the methods and models developed during this work was provided by Adapco and DaimlerChrysler. Both steady-state and transient radiator experiments were performed by DaimlerChrysler in order to characterize the radiator response under various thermal-hydraulic conditions. A selected radiator was installed in the DaimlerChrysler experimental facility as shown in Fig. 3.1. The radiator experiment was instrumented allowing the measurement of the inlet air flow rate and temperature, the outlet air average temperature, the inlet coolant flow rate and temperature, and coolant outlet temperature. An initial series of steady-state experiments was performed to determine the steady-state radiator performance curves. In these experiments the coolant flow was varied in the range 0.5 - 3.5 Kg/s. For each coolant flow rate value a series of experiments with air flow rates between 0.1 - 3.3 Kg/s was performed. The heat transfer information obtained from these runs has been used in the STAR-CD heat exchanger input file, as described in Section 6.

Four series of transient experiments were performed to explore the transient response of the radiator under various conditions: a) transient inlet coolant temperature, b) transient coolant flow rate, c) transient air flow rate, and d) transient coolant and air flow rate. During the experiments, the selected transient parameters were changed in time in a prescribed manner. The values of all characteristic thermal-hydraulic values of the radiator were measured and recorded at fixed time intervals of 2.3 s during the experiments. The length of the experiments varied between 500 and 1000 s. In this preliminary work we have selected one example run from the series a, b, and c for analysis. The characteristics of the selected experiments and the results of the STAR-CD analyses are presented in Section 7.

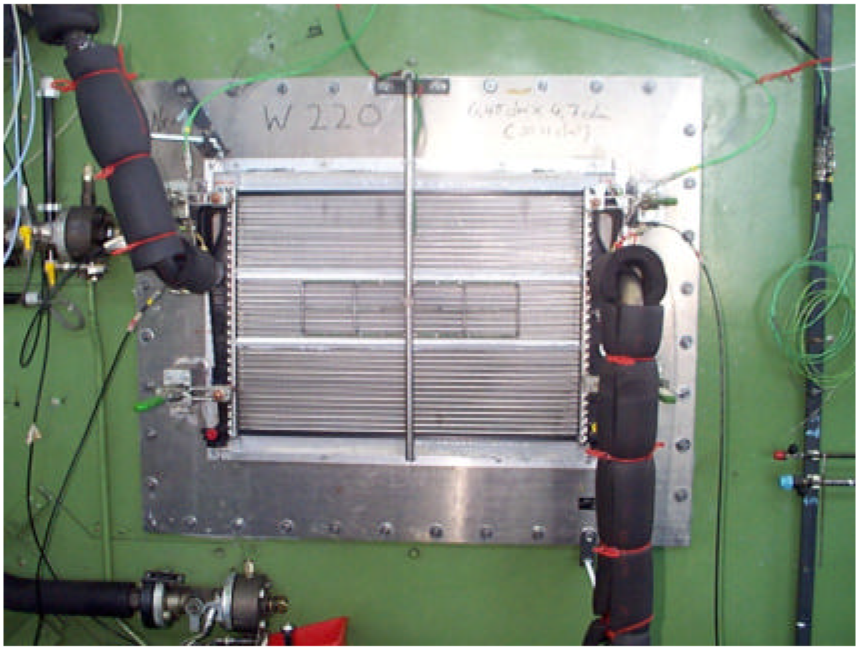


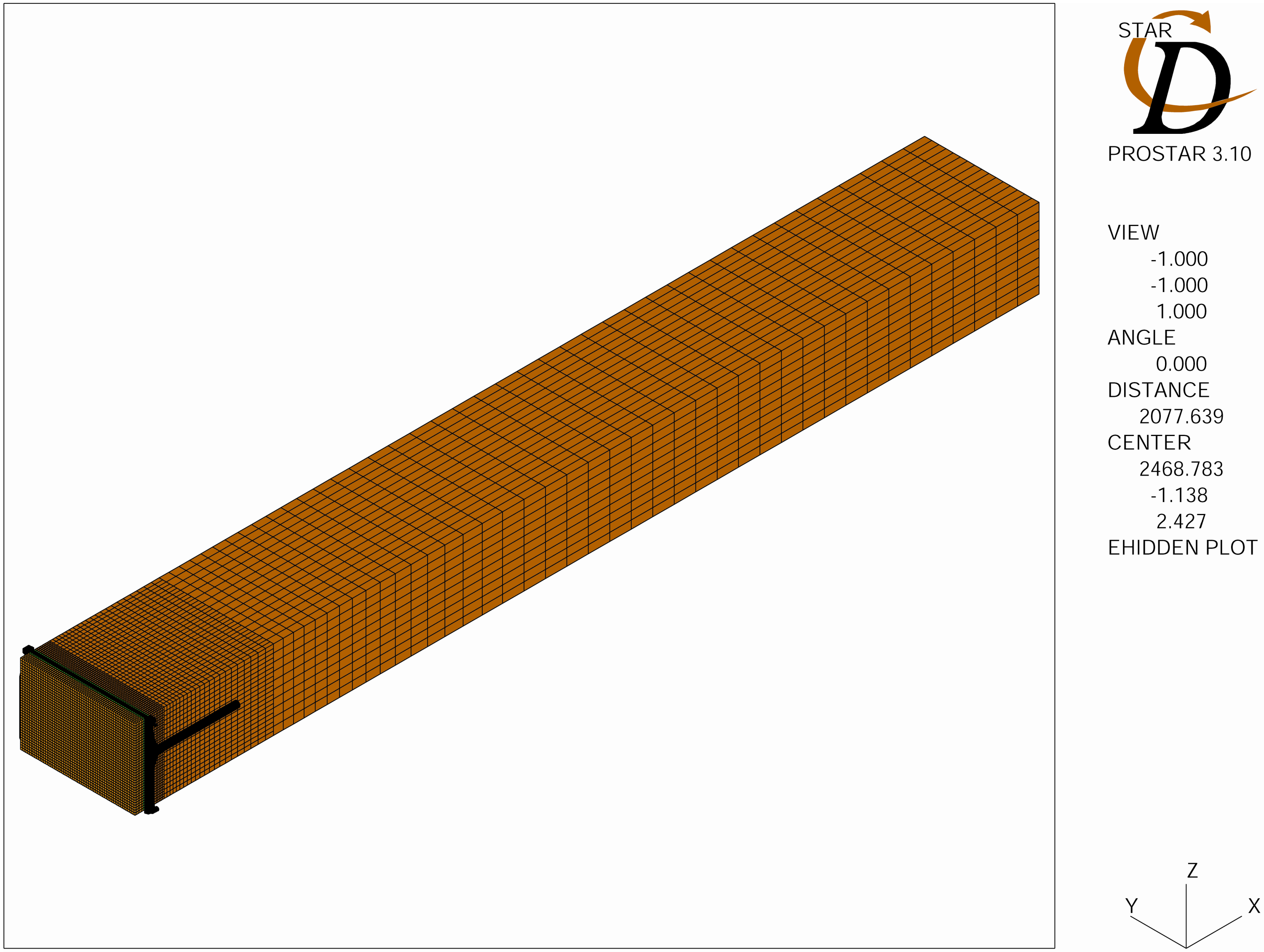
Fig. 3.1. Radiator Experimental Facility.

4. Computational Model Description

The computational STAR-CD model describing the radiator experimental setup was provided by DaimlerChrysler. It consists of 217,150 fluid cells. This model only has a radiator component, which was represented by 14,208 air-coolant cell pairs. The mesh configurations used in the STAR-CD calculations are presented in Figs. 4.1 and 4.2. In this model, the left, right, top and bottom surfaces are assumed to be slip surfaces. The vertical faces at the front and back side of the model are assumed to be the inlet and outlet regions of the air stream, respectively. The inlet pipe of the coolant flow is clearly shown near the front face of this model. The high Reynolds number turbulence flow model is used for both air and coolant. Buoyancy and temperature effects are also considered.

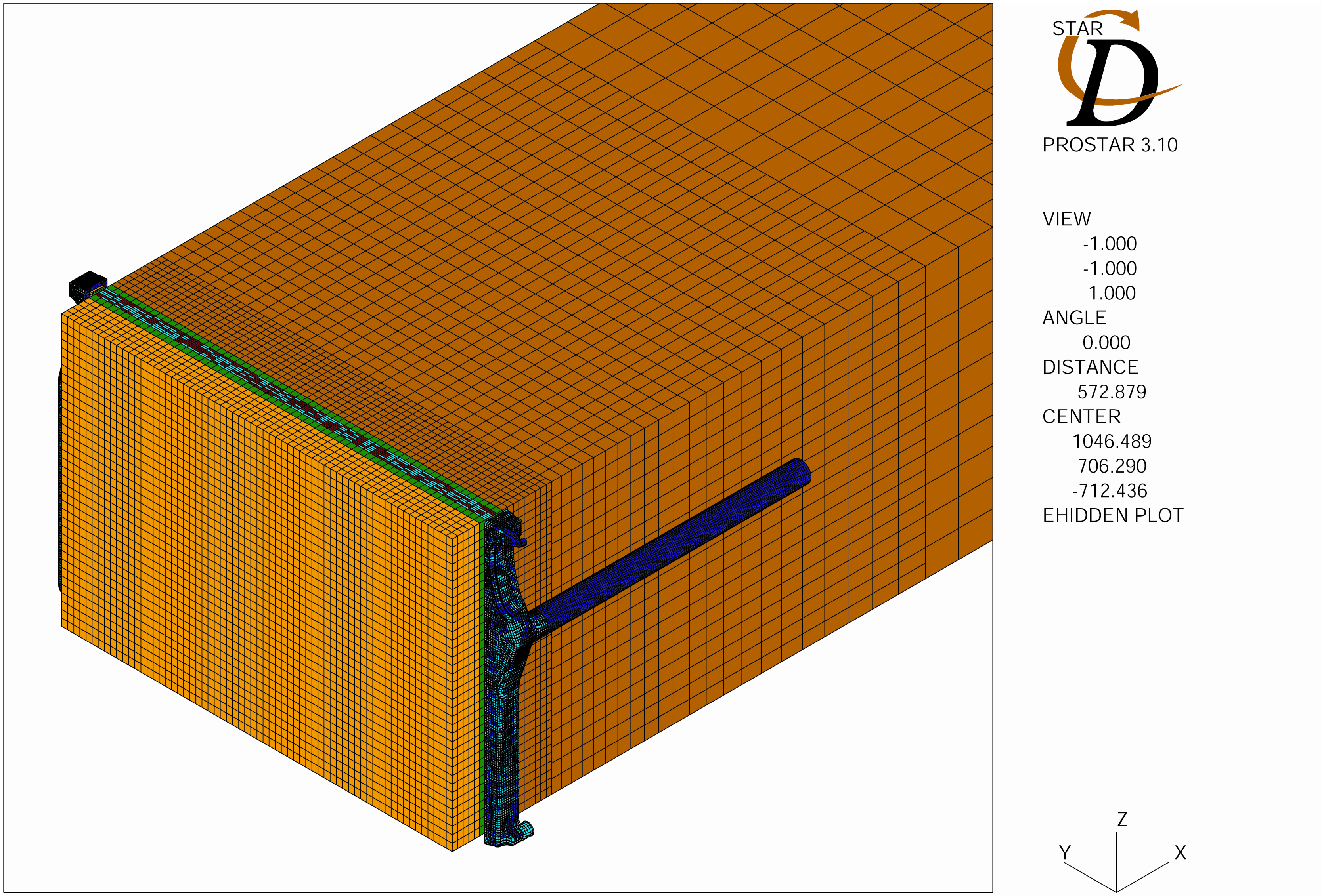
In the analysis, both air and coolant flows are considered. Both streams are modeled explicitly in STAR-CD by defining two overlapping meshes with different material types (e.g. radiator air-side cells of material type 1 overlapping radiator coolant cells of material type 2).The radiator is model as "slab" of cells, with little or none of the internal geometric details (i.e. fins, tubes) included, as shown in Fig. 4.3. Pressure losses through the air and coolant sides of the radiator were simulated by treating the respective fluid streams as porous media. The porous medium constants were obtained from test data of pressure drop versus flow rate for the radiator.

In the STAR-CD calculations two options can be used to specify the heat transfer data in the HX input file, hxinput.dat. In the first option, the coolant mass flow rate is constant. The HX test data are input under keyword UADAT which specifies air mass flow rates versus overall local heat transfer rate between the air and the coolant. In the second option, both mass flow rates of air and coolant can be varied. The HX test data are specified under the keyword QDATA. The input data involve the air and coolant mass flow rates, inlet temperatures of air and coolant, and the total heat transfer. For the second option, STAR-CD determines the heat transfer coefficients through a quadratic curve fitting method, rather than the linear interpretation method. A new option, specified under the keyword UCDAT has been developed during this work and is described in Section 7. This option allows the user to input a series of air flow rate data for each specified coolant flow rate, and then interpolates linearly between the specified performance curves. This ensures full consistency with the performance curve data, eliminating the need for the quadratic curve fitting.



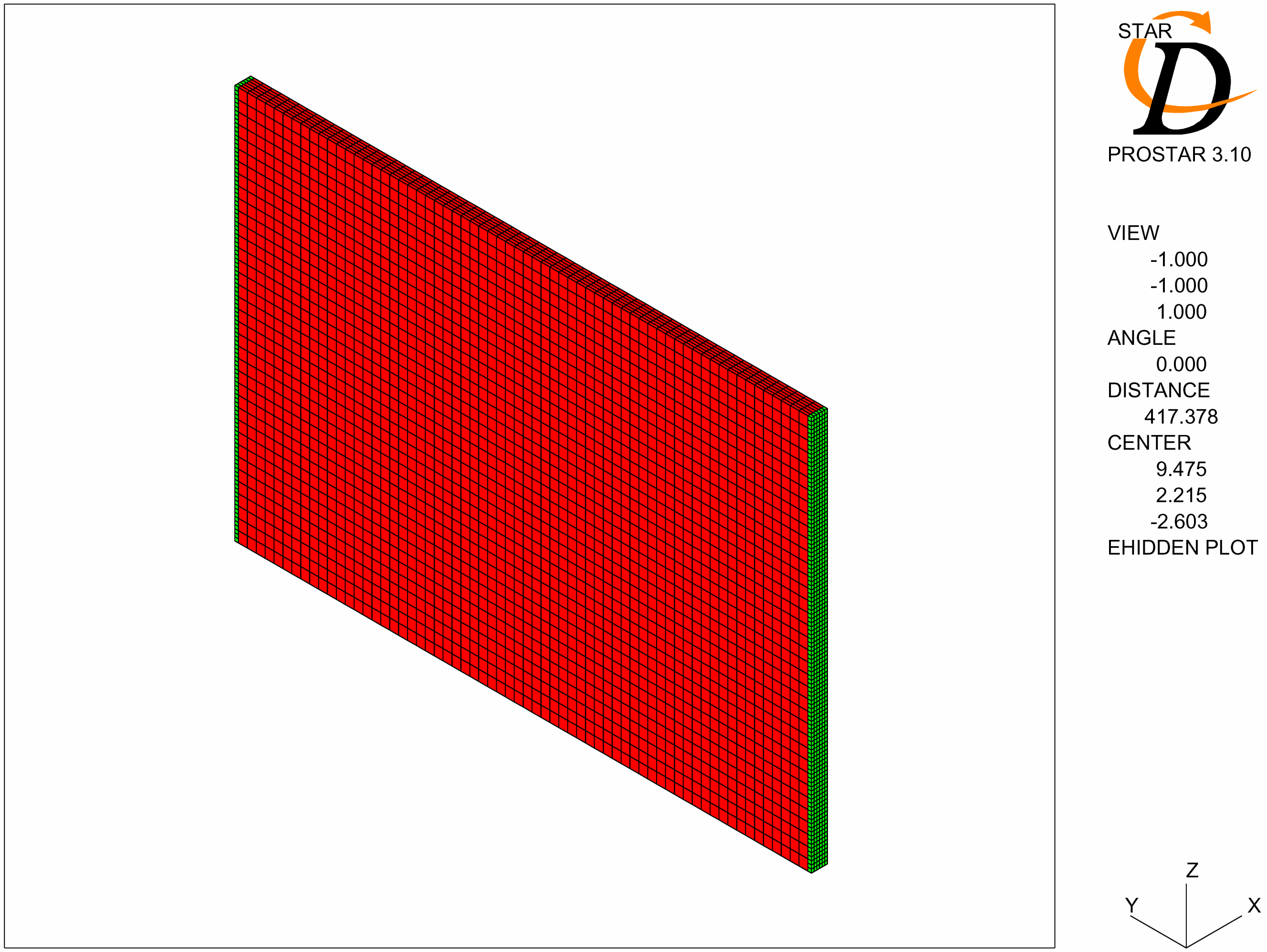
Analysis of Radiator Performance (m1rtestg)

Fig. 4.1 : STAR-CD Mesh Configuration



Analysis of Radiator Performance (m1rtestg)

Fig. 4.2 : Mesh Configuration Near Radiator



Analysis of Radiator Performance (m1rtestg)

Fig. 4. 3 : Model of Radiator

5. Initial Calculations for Performance Evaluation

Presently, the RAE LINUX Cluster consists of one master node (RE1) and 39 slave nodes (RE2 - RE40). The master node has two 800 MH Pentium 3 processors, and 1 GB memory. Each slave node, between RE2 through RE17, has one 800 MH Pentium 3 processor and 512 MB memory. Other slave nodes, RE18 through RE40, have an 856 MH processor and 512 MB memory.

Before developing an efficient method, several test calculations were made with the STAR-CD code to study the CPU time required for various runs. The CPU time is important in developing the methodology and in determination of the solution strategy. Both steady-state and transient calculations were made. For the illustration purpose, the CPU times on different calculations using the radiator model provided by DaimlerChrysler are given here.

5.1 Steady State Calculations

In the steady state calculation, the STAR-CD analysis was performed initially in serial mode. The solution converged at the 580th iteration and the CPU time is about 11.58 hours. Since the calculation that utilized the serial mode is quite time consuming, several steady state calculations were performed with the STAR-CD code using the RAE LINUX cluster and the results were compared with the steady-state experiments. In these calculations, three different coolant mass flow rates, 0.5, 2.5, and 3.5 kg/s are used at the coolant inlet , while two different air mass flow rates, 1.91, and 3.32 kg/s are utilized at the air inlet.

Table 5.1 provides some results of selected steady state calculations. In the first three runs, the UADAT option of hxinput.dat is used. In the fourth run the QDATA option of hxinput.dat is utilized to test the accuracy of this option which would be needed for the transient problems where both air and coolant mass flow rates may be varied. Results indicated that in general the STAR-CD predicted outlet temperatures of both air and coolant agree well with the measured data. Figures 5.1 and 5.2 show the residuals of various field variables for the air (Fluid 1) and coolant (Fluid 2), respectively.

However, under the same input conditions, the agreement between the calculated and measured result of the Run 4 is not as good as that of the Run 2. This implies that the curve fitting method used in the QDATA option for calculating the heat transfer coefficients may not be a good approach. A new option UCDAT allowing the use of linear interpretation when multiple performance curves for multiple values of the coolant flow rate are specified has been developed during this work and is described in Section 6.

The CPU times for various steady state runs are also given in Table 5.1. Compared the runs with the same air and coolant mass flow rates shown in Runs 2 and 4 revealed that the CPU times is reduced more than 4 times if 16 processors is used rather than the 4 processors. For a typical run using 4 processors, such as Run 4, the CPU time is about 4.5 hours.

5.2 Transient Calculations

In the transient analysis, the inlet air and coolant flow rates are kept constant. However, the inlet coolant temperature of the radiator is varied and is described in a user developed subroutine. Two sets of short calculations were made. The first set of calculations involves 20 time steps (cycles). Because of considerations of numerical stability and accuracy, the step size used in the numerical integration is quite small, about 0.0008 sec. The CPU time for this short, coupled air flow and radiator transient analysis using a single processor (the serial mode) is about 9,782 sec, or 2.72 hours. The same transient analysis was also performed on the RAE LINUX cluster parallel computing system using 4 nodes (processors). The CPU time for this parallel calculation is about 1,280 sec (0.36 hour). The ratio of the CPU times for the serial and parallel runs is about 7.6 to 1.

As a sensitivity study, a second set of transient calculation involving 40 time steps (cycles) were also made. Again, STAR-CD calculations were performed with both serial and parallel modes. The resulting CPU times for the serial and parallel calculations are 23,577 and 3,234 sec, respectively (or 6.55 and 0.90 hours, respectively). The ratio of the CPU times for the serial and parallel runs is about 7.30 to 1. For the ease of comparison, results of the computational times for all the runs are given in Table 5.2.

To study the effect of number of parallel processor on the computational time, two more calculations (Cases 5 and 6) were performed using 8 and 30 nodes (processors). The CPU time for the 30-node calculation using the same step size is 235 sec, or 0.065 hour. The ratio of the CPU times for the 4- and 30- node runs is 13.76 to 1. Comparison of the computational times is given in Table 5.3.

From the results presented in Tables 5.2 and 5.3, it is shown that the relationship of the computing time involved with the number of processors used is nonlinear. The more processors used, less computing time is needed. Note from Table B that the difference of CPU times for the 4- and 8- node calculations is quite large. However, for the 8-node and 30-node calculations the difference is relatively small.

In Tables 5.2 and 5.3, the elapsed time for each STAR-CD calculation is included. This time represents the clock time needed for the numerical simulation, including times for computation, data exchanges, and waiting time when the system was used by other users.

Computing performance, defined by ratio of real time and the computer time, for transient calculations using 1, 4, 8, and 30 nodes (processors) are given in Table 5.4. The performance curve based on the CPU time is shown in Fig. 5.3.

5.3 Remarks on Initial Performance Evaluations

It is noted that in the second set of the transient calculation, the numerical solution covers only 0.032 sec real time. For a typical radiator transient of 15 minutes, such as climbing a long steep slope, the CPU time for a full transient calculation would be extremely large even using the parallel machine. This points to the need for the use of the quasi transient methods which have been developed during this work and are described in Section 6.

Table 5.1. STAR-CD Steady State Analysis of Radiator Experiments

Run	**Mass Flow Coolant (kg/s)**	**Mass Flow Air (kg/s)**	**Inlet Temp Air (°C)**	**Outlet Temp Air (°C)**	**Inlet Temp Coolant (°C)**	**Outlet Temp Coolant (°C)**	**STAR-CD Air T_{out} (°C)**	**STAR-CD Coolant T_{out} (°C)**	**No. of Processors**	**CPU Time (sec)**	**File Name**
1	0.5	3.31	25.43	32.69	80.49	67.32	31.82	68.32	2	23,193	mlrb1
2	2.5	3.32	26.58	45.59	80.13	73.08	45.48	73.15	16	3,255	mlrtestg
3	3.5	3.32	26.37	47.43	80.18	74.60	47.71	74.51	4	3,855	mlrc1
4	2.5	3.32	26.58	45.59	80.13	73.08	43.76	73.79	4	16,282	mlrtesth

Notes: (1) Runs 1, 2, 3 use UADAT in the hxinput.dat file.

(2) Run 4 uses QDATA in the hxinput.dat file.

(3) T_{out} is the outlet temperature calculated with the STAR-CD code.

(4) For Runs 1 and 3, solutions converged at 495th and 354th iteration, respectively.
For both Runs 2 and 4, solutions converged at 582nd iteration.

Table 5.2. Comparison of Computing Times for Various STAR-CD Transient Calculations.

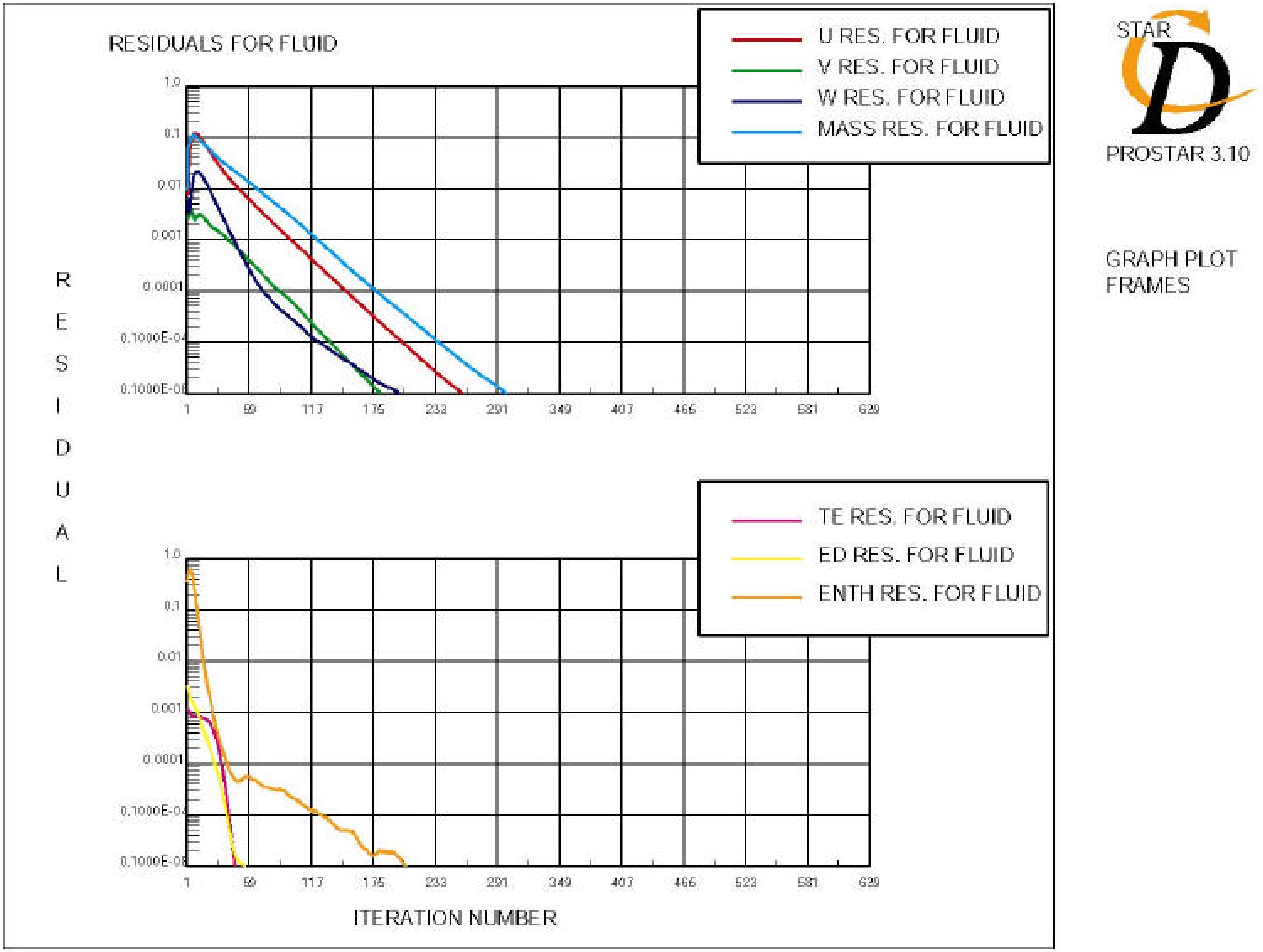
Case No.	Computing Mode		No. of Time Steps	Size of Time Step (sec)	CPU Times (sec)	Ratio S/P	Elapsed Time (sec)	Ratio (S/P)	File Name
	Serial	Parallel (# Nodes)							
1	S		20	0.0008	9,782	7.6	10,334	7.6	m1rt
2		P (4)	20	0.0008	1,280	1	1,353	1	m1rp
3	S		40	0.0008	23,577	7.30	24,108	7.05	m1rt2
4		P (4)	40	0.0008	3,234	1	3,419	1	m1rp2a

Table 5.3. Comparison of Computing Times for Calculations with Different Number of Processors.

Case No.	**No. of Processors**	**No. of Time Steps**	**Size of Time Step (sec)**	**CPU Time**			**Elapsed Time**			**File Name**
				(sec)	**(hr)**	**Ratio**	**(sec)**	**(hr)**	**Ratio**	
4	4	40	0.0008	3,234	0.90	13.76	3,419	0.95	4.31	m1rp2a
5	8	40	0.0008	584	0.16	2.48	793	0.22	1	m1rp3
6	30	40	0.0008	235	0.065	1	794	0.22	1	m1rp4

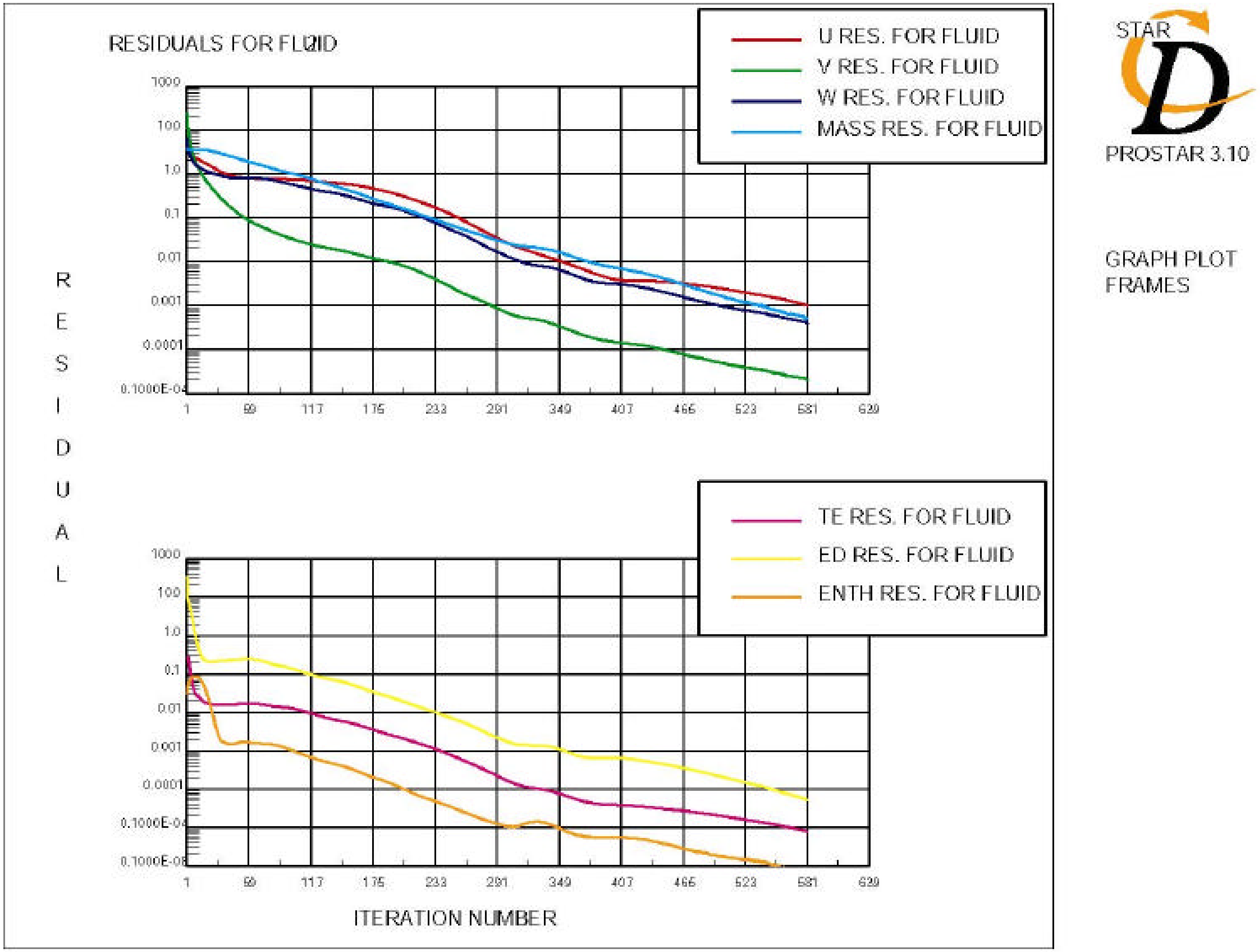
Table 5.4. Comparison of Computing Performance for Various Transient Calculations

(1) **Case No.**	**(2)** **No. of Processors**	**(3)** **No. of Time Steps**	**(4)** **Size of Time Steps (sec)**	**(5)** **Real Time (sec)**	**(6)** **CPU Time (sec)**	**(7)** **Performance (5)/(6) x 10^{-5}**	**(8)** **Elapsed Time (sec)**	**(9)** **Performance (5)/(8) x 10^{-5}**
3	1	40	0.0008	0.032	23,577	0.14	24,108	0.13
4	4	40	0.0008	0.032	3,234	0.99	3,419	0.94
5	8	40	0.0008	0.032	584	5.48	793	4.03
6	30	40	0.0008	0.032	235	13.62	794	4.03



Analysis of Radiator Performance (m1rtestg)

Fig. 5.1 : Residuals for Fluid 1 (Air)



Analysis of Radiator Performance (m1rtestg)

Fig. 5.2 : Residuals for Fluid 2 (Coolant)

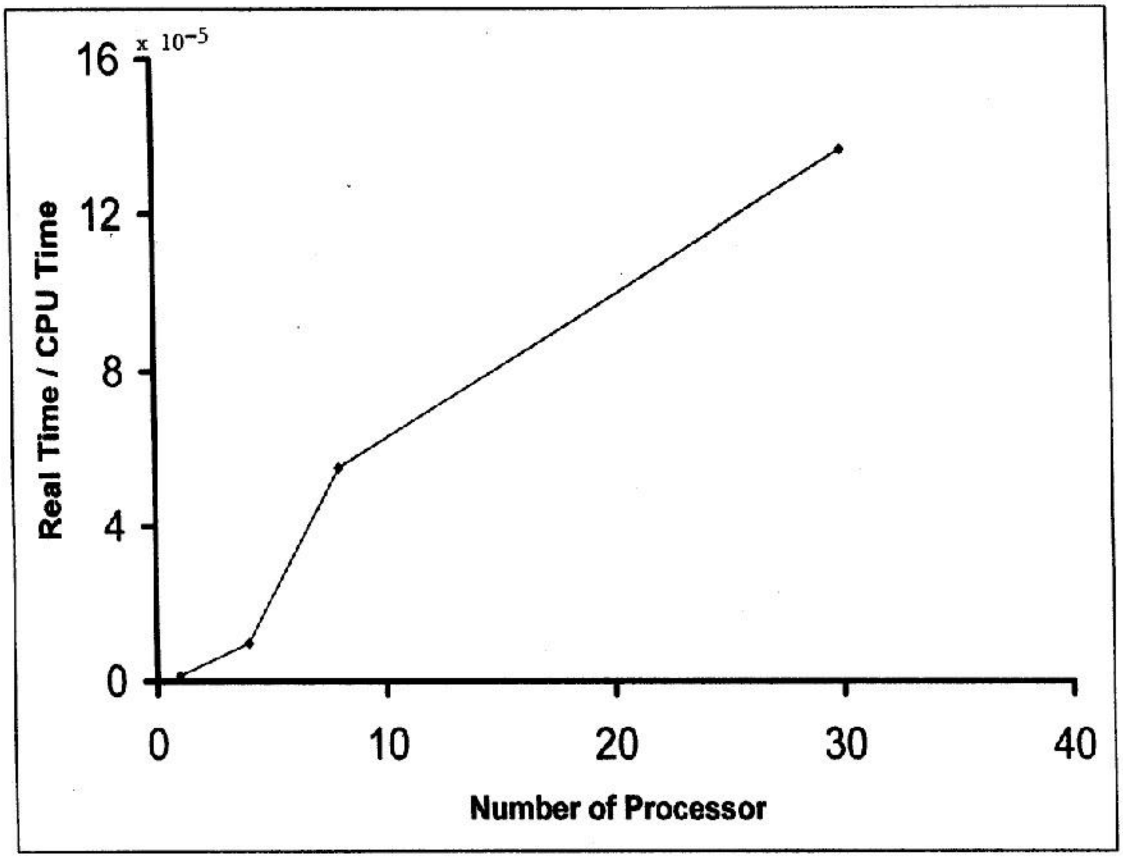


Fig. 5.3. Computing Performance for Transient Calculations.

6. Description of new quasi-transient methodologies and models

This section describes the new quasi-transient method. which have been implemented in STAR-CD during this work. A number of new models and software features necessary for the correct calculation of transient events have also been developed and are described in this section.

6.1 Quasi-transient methodologies for the study of extended thermal-hydraulic transients.

It was shown in Section 5 that the use of conventional full transient calculations with STAR-CD becomes quickly prohibitive for periods of time of interest, even if powerful parallel computer are used. In this work we have developed two new quasi-transient methodologies which allow STAR-CD to calculate such extended transients using reasonable amount of CPU time with only small losses of accuracy. These two methods are described below. Although they have been studied independently in this preliminary work, they are complementary and it is likely that a combination of the two approaches will be able to provide further improvements in computational efficiency and accuracy.

6.1.1 Quasi-transient methodology using a combination of thermal transients and multiple steady state STAR-CD calculations.

This methodology is based on the fact that the time constants that characterize the air and coolant flow field changes are much shorter than the heat transfer time constants. When modeling a thermal-hydrodynamic transient with STAR-CD the time steps are limited to very small values, in the range of 0.001 s, due to numerical stability requirements, as shown in Section 5. If, however, the velocity fields are kept constant and only the temperature fields are updated, we can advance the solution using much longer time steps, in the order of 1 s, because of much less stringent stability limitations imposed by the energy conservation equations. The velocity fields can then be updated periodically, by performing a steady state calculation at intervals dt_ss which depend on the rate of change of the coolant and air flow rates and the desired accuracy of a selected calculated quantity. The procedure for the selection of the next dt_ss time interval is outlined below.

In this work we focused our attention on the accuracy of the coolant outlet temperature Tcool, requiring that over the selected time interval:

$$\text{Tcool(calc)-Tcool(actual)} < \text{dTcool\_error\_max} \qquad (6.1)$$

Where:

Tcool(calc) - Coolant outlet temperature calculated using the flow rate which remains constant over the time step, in K

Tcool(actual) - Coolant outlet temperature calculated using the actual flow rate, which varies over the time interval, in K

dTcool_error_max - specified maximum error in the coolant outlet temperature, in K

For an experiment where the inlet air flow Qair changes:

$$\text{Tcool} = \text{Tcool(Qair)} \tag{6.2}$$

By further assuming a linear behavior over the time interval, i.e. neglecting the higher order derivatives, we obtain:

$$\text{dTcool(dt\_ss)} = \text{dTcool/dQair} * \text{dQair/dt} * \text{dt\_ss} \tag{6.3}$$

Where:

dTcool(dt_ss) = coolant outlet temperature change during the time step due to changes in the air flow rate, absolute value in K

dTcool/dQair = rate of change of coolant outlet temperature with air flow rate, absolute value obtained from experimental data, in K/(Kg/s)

dQ(air)/dt_ss = rate of change of air flow rate with time, assumed constant during the time step, absolute value obtained from experimental data, in (Kg/s)/s

dt_ss = time step to the next steady state, in s

The value dTcool(dt_ss) obtained from equation 6.3 represents the expected maximum error we would be making in the steady state calculation of Tcool using the value of Qair at the end of the time step.

Comparing equations 6.1 and 6.3 we obtain the maximum value of dt_ss that ensures the required accuracy in the calculation to Tcool:

$$\text{dt\_ss\_max} = \text{dTcool\_error\_max} / [\text{dTcool/dQair} * \text{dQair/dt}] \tag{6.4}$$

The use of equation 6.4 automatically increases the time intervals between steady state calculations when the air flow rate change. The time step dt_ss_max is larger when the time changes in the air flow rate, measured by dQair/dt are small, and decreases when the changes in the air flow rate increase. Similar considerations can be used to determine dt_ss_max when Qcoolant changes occur during the transient.

We can increase the time step by using an average air flow rate Qair_average(dt_ss) over the time step. With the assumption of linear changes over the time step, the maximum time step is given by::

$$\text{dt\_ss\_max} = \text{dTcool\_error\_max} / [0.5 * \text{dTcool/dQair} * \text{dQair/dt}] \tag{6.5}$$

The method used for quasi-transient calculations involves the following steps:

a. An initial steady state calculation that determines the initial velocity and temperature fields for both air and coolant;
b. Determine the time of the next steady state calculation using the approach

outlined above;

c. Determine the average coolant and air flow rates for the selected time interval dt_ss;

d. If either the coolant or air average flow rate determined in c above is different from the flow rate used in the previous steady state calculations, perform a new steady state calculation adjusting the flow rates to the average values over the time interval dt_ss, but keeping the temperature fields unchanged;

e. Perform a transient thermal calculation using time steps in the order of 1 s, that updates the temperature fields only, but keeps the velocity fields unchanged;

f. Check if the end of the transient has been reached. If not, repeat steps b through d.

Results obtained using this method are presented in Section 7. It is shown that the desired accuracy can be obtained within a reasonable computational time. The use of the new heat exchanger input option UCDAT, described in Section 8, is necessary when analyzing transients that involve changes in the coolant flow rate. The use of this new option allows the user to specify the radiator performance curves for various values of the coolant flow rate, and then call the appropriate new heat transfer routines hxmthd32 and hxmthd33 described in Section 8 to determine the heat transfer coefficients. No attempt has been made in this pilot project to automate the sequence of thermal transients and steady state calculations required by the method. It is recommended that such an automatic procedure, including the determination of the time steps between consecutive steady state calculations, be developed and implemented in STAR-CD as part of a future effort.

6.1.2 Quasi-transient methodology using a velocity field similarity assumption.

A second quasi-transient methodology developed during this work is based on the assumption that changes in inlet coolant and/or air flow rates lead to proportional changes in the corresponding velocity fields. This assumption allows a more accurate modeling of the transient phenomena than does the first method described in Section 6.1.1. Two effects related to changes in the air or coolant flow rate are modeled in the new heat transfer routines hxmthd32 and hxmthd33: a) the changes in the heat transfer coefficients due to changing flow rates and b) the effect of changing flow rates on fluid temperature changes.

To use this method the user must specify the option UCDAT, described in Section 8, as well as the key words TRANC (for coolant flow rate transients) and TRANA (for air flow rate transients) in the heat exchanger input file. The new heat transfer routines hxmthd32 or hxmthd33 determine each time step the ratio of the current air and coolant flow rates to the initial respective values. This factor is then used to determine the local current velocities, by multiplying the initial values determined during the steady state calculation. The local current velocities are used to determine the correct heat transfer coefficients. These coefficients are used to determine the heat flux exchanged between the coolant and air during each time step and is passed on to the STAR-CD solver for use in finding the hydrodynamic solution during the next time step. Because we are not changing the actual velocity fields during the quasi-transient calculation, the heat transfer routines must adjust the heat flux to account for the air and coolant flow rate changes. To achieve this, the heat flux transferred to the coolant is obtained by dividing the calculated heat flux by the current coolant flow rate factor. Similarly, the heat flux transferred to the air is obtained by dividing the calculated heat flux by the current air flow rate factor.

Results obtained using this approach are illustrated in Section 7 and appear to capture very well the transient effects related to coolant and air flow rate changes, while avoiding prohibitively lengthy full hydrodynamic transients. Future work should explore the flow rate changes and flow configuration over which the velocity similarity assumption remains valid. A more complex strategy will be needed for flow configurations which contain multiple independent inlets for the same fluid. It is likely that a combination of the similarity quasi-transient approach with the first method described in Section 6.1.1 will be necessary for extended transients involving large flow rate changes. Such a combined method, to be explored in future work, could use the similarity quasi-transient approach combined with periodic steady state calculations which would update the velocity fields when the velocity similarity assumption might become questionable.

6.2 Effect of Radiator Mass on Transient Effects

The STAR-CD heat exchanger models do not explicitly account for the effect of the radiator mass during thermal transients. During this work we developed a preliminary approach which allows the modeling of these effects. This was achieved by introducing a new array - tmetal - which contains the current temperature of the radiator metal in each radiator cell. The user must define the metal properties - specific heat, conductivity, smeared density - in the heat exchanger input file, as described in Section 8. The radiator metal effects are considered in the new heat transfer routine hxmthd33 which is called if the input smeared density RHOM is greater than 0. This routine is called first during the steady state calculations to determine the initial metal temperature in each radiator cell. This is achieved by calculating the components of the total heat transfer coefficient in the new routine calual4:

$$1/Htot = 1/Hc + 1/Hm + 1/Ha \qquad (6.6)$$

Where:

Htot = total heat transfer coefficient, in J/(m2*K)

Hc = coolant side heat transfer coefficient, in J/(m2*K)

Hm = metal heat transfer coefficient, in J/(m2*K)

Ha = air side heat transfer coefficient, in J/(m2*K)

The heat transfer coefficient between coolant and metal is then calculated from:

$$1/Hcm\_ss = 1/C * (1/Hc + 2/Hm) \qquad (6.7)$$

Where:

C = Htot_ss / Htot is a scaling factor, that accounts for the fact that Htot obtained from equation 6.6 may be different from the Htot_ss, the actual total heat transfer coefficient determined from the performance curves and used in the heat exchanger model

Htot_ss = total heat transfer heat transfer coefficient, obtained from the radiator performance curves

The steady state metal temperature is then calculated as:

$$Tmetal(i) = Tcool(i) - [Tcool(i)-Tair(i)] * Htot\_ss/Hcm\_ss \qquad (6.8)$$

Where:

Tmetal(i) = metal temperature in radiator cell i

Tcool(i) = coolant temperature in radiator cell i, calculated by STAR-CD

Tair(i) = air temperature in radiator cell i, calculated by STAR-CD

The Tmetal values are saved in a file at the end of the steady state calculations and will thus be available for the following transient calculations. At the beginning of the transient calculations, the key word TRANM must be specified in the heat exchange input file, in addition to having a positive value for the key word RHOM. This will cause the Tmetal values to be read from file in the routine hxinit1. The changes in the metal temperature during transients are calculated in the routine hxmthd33 using a quasi transient approach, which assumes that the heat transfer between the coolant and metal has reached a steady state consistent with the heat flux obtained from the performance curves. This approach was used in this preliminary implementation to avoid potential temperature instabilities due to the low thermal inertia of the metal, high heat transfer coefficients between the coolant and the metal and relatively long time steps. An initial attempt to perform a full transient metal calculation using an explicit method has led to numerical instabilities, indicating the need for a more implicit approach that should be explored in future work. During transients, the new metal temperature is calculated using equation 6.8. The volumetric heat flux associated with the change in radiator metal temperature is calculated from:

$$Hmetal = [Tmetal(i) - Tmetal\_old(i)] * cp\_metal * rho\_metal / dt \qquad (6.9)$$

Where:

Tmetal(i) = New metal temperature in radiator cell i, in K

Tmetal_old(i) = Previous time step metal temperature in cell i, in K

cp_metal =metal specific heat in J/(Kg*K)

rho_metal = metal density, smeared over the cell i volume, in Kg/m3

dt = thermal transient time step, in sec

The value Hmetal calculated with equation 6.9 is then used to modify the volumetric heat flux transferred to the air side thermal calculation:

$$Hair\_metal = Hair – Hmetal \qquad (6.10)$$

Where:

Hair_metal = Vulumetric heat flux transferred to the air, adjusted for the metal thermal effect, in J/(s*m3)

Hair = Volumetric heat flux transferred to the air using the standard steady state equilibrium assumption used by the STAR-CD heat exchanger model, in J/(s*m3)

The volumetric heat flux used in the coolant energy conservation equation is not modified in the current implementation. This approach provides a reasonable approximation for the two cases studied, i.e. increasing coolant temperature and increasing air flow rate and was used to evaluate the impact of the radiator thermal inertia during thermal transients as shown in Section 7. In future work this approximate procedure should be replaced by a full transient calculation of the metal temperature. Furthermore, the Tmetal array should be made part of the STAR-CD data structures which are saved for restarts and decomposed for parallel computing platforms. The current implementation has been developed for and tested on two processors only.

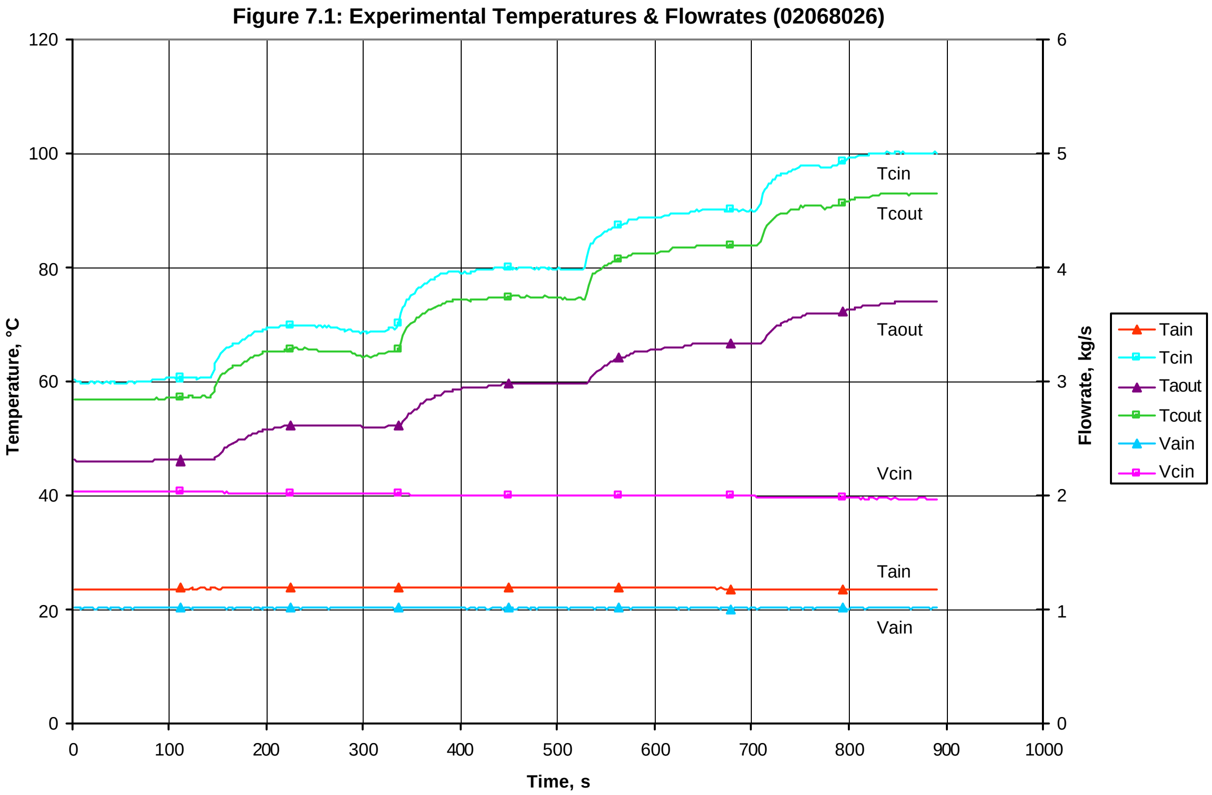
7. Experiment Analysis Results

This section describes the results of STAR-CD analyses of the radiator transient experiments described in Section 3. The experiments were performed by DaimlerChrysler Corporation and the data was provided by Adapco and DaimlerChrysler in Excel format. We selected three experiments that were used to validate the new methods and models. All the calculations are done with STAR-CD using the radiator model described in Section 4.

7.1 Experiment Changing the Inlet Coolant Temperature (02068026)

Figure 7.1 shows the experimental data that our calculation is trying to match. The air inlet flow rate (Vain), coolant inlet flow rate (Vcin), and air inlet Temperature (Tain) were held constant. The coolant inlet temperature (Tcin) was stepped up every 3 minutes. The average air outlet temperature (Taout) and average coolant outlet temperature were measured. Our analysis using the new methods and models are comparing the outlet coolant and air temperatures, calculated using the prescribed inlet temperatures and flow rates, with the corresponding experimental data.

We did three different calculations using the data from this experiment. In all cases we performed a single initial steady state calculation, solving both the flow and temperature fields. Because the flow rates do not change in this experiment, a single transient run from 0 to 891 seconds, solving only the temperature fields followed the initial steady state.



For the steady state run the following parameters were used:

	Coolant	Air
Temperature, °C	60.	23.6
Flow rate, kg/s	2.03	1.0
Specific Heat, J/kg-°C	3518.92	1007.0
Density, kg/m**3	1048.38	1.164

During the transients the inlet coolant temperature was changed as a function of time. The results of the calculations are described below and illustrated Figure 7.2. A more detailed view of the same results during the first 200 seconds of the experiment is shown in Figure 7.3.

7.1.1 Run using UADAT Input Option

This run used the original heat exchanger model that was developed at Adapco. None of the new models developed during this work was used in this analysis, and the goal was to evaluate how well the base version of the STAR-CD performs in a case where transient coolant inlet temperatures are present. The calculated coolant outlet temperatures are in good agreement with the experimental measurements. The calculated coolant temperature increase is slightly delayed compared to the measured data, as shown is Figure 7.3. The calculated air outlet temperatures are in reasonable agreement with the experimental data. The initial steady state calculated value is 1.85 C higher than the measured corresponding value. As shown in Fig. 7.2, the calculated air outlet temperature remains higher than the measured values throughout the experiment, but it gradually approaches the measured value toward the end of the experiment. The calculated air outlet temperature increase occurs about 10 s earlier than the measured increase, as illustrated in Figure 7.3.

The following input file for the radiator model (hxinput.dat) was used for this run.

```
PARALLEL
NAME    (of the heat exchanger)
RADIATOR
METHOD   (1,2,3)
3
ATYPE
20
0
CTYPE
10
0
AUTYPE
21
ADTYPE
22
CUTYPE
11
CDTYPE
12
NACS
2368
NHXC
64
NHXA
6
TINC
333.15
TINA
296.75
CPC
3518.92
CPA
1007
RHOA
0.3277824
XMC
2.
URFQ
1.0
URFT
1.0
UADAT
0.10,100.0
0.50,430.0
0.90,630.0
1.31,750.0
1.71,850.0
2.11,920.0
2.51,990.0
2.91,1040.0
3.32,1080.0
0.0,0.0
ISTART
1
```

7.1.2 Run using UCDAT Input Option

This run was identical to the UADAT described above, except that the new UCDAT option for the heat exchanger model was used. This new capability, described in Section 8, allows the use of radiator performance curves for multiple coolant flow rates and uses linear interpolation between the various performance curves. While this option is important for transients where the coolant flow rate changes, it was not required in the analysis of this test. The goal of this run was to verify that the new option UCDAT gives similar results to the UADAT option when the coolant flow rate remains constant. As illustrated in Figures 7.2 and 7.3 the results obtained using the UCDAT option are almost identical for both the coolant and air outlet temperatures. The following input file for the radiator model (hxinput.dat) was used for this run.

```
PARALLEL
NAME    (of the heat exchanger)
RADIATOR
METHOD    (1,2,3)
3
ATYPE
20
0
CTYPE
10
0
AUTYPE
21
ADTYPE
22
CUTYPE
11
CDTYPE
12
NACS
2368
NHXC
64
NHXA
6
TINC
333.15
TINA
296.75
CPC
3518.917
CPA
1007
URFQ
1.0
URFT
1.0
AREAA
0.2816
AREAC
0.00792
```

```
UCDAT
0.50
0.10,292.92,353.43,90.0
0.50,291.86,353.54,270.0
0.90,292.63,353.43,330.0
1.31,294.40,353.38,370.0
1.71,295.25,353.47,390.0
2.11,296.27,353.63,400.0
2.51,296.96,353.59,410.0
2.92,297.75,353.66,420.0
3.31,298.58,353.64,430.0
0.0,0.0,0.0,0.0
1.50
0.10,293.80,353.16,100.0
0.50,292.27,353.38,410.0
0.90,293.31,353.39,580.0
1.31,294.72,353.34,690.0
1.71,296.07,353.33,770.0
2.11,297.05,353.32,830.0
2.51,298.15,353.31,880.0
2.91,298.91,353.29,920.0
3.31,299.73,353.28,960.0
0.0,0.0,0.0,0.0
2.50
0.10,293.80,353.16,90.0
0.51,292.27,353.38,450.0
0.91,293.31,353.39,660.0
1.31,294.72,353.34,800.0
1.71,296.07,353.33,900.0
2.11,297.05,353.32,990.0
2.51,298.15,353.31,1060.0
2.91,298.91,353.29,1130.0
3.32,299.73,353.28,1180.0
0.0,0.0,0.0,0.0
3.50
0.10,294.59,353.32,100.0
0.51,292.54,353.42,460.0
0.90,293.40,353.43,690.0
1.31,294.42,353.38,850.0
1.71,295.65,353.37,970.0
2.11,296.98,353.37,1070.0
2.51,297.94,353.37,1160.0
2.91,298.73,353.32,1230.0
3.32,299.52,353.33,1300.0
0.0,0.0,0.0,0.0
0.0
ISTART
1
```

7.1.3 Run using Metal Model

This run takes into account the thermal inertia of the metal in the radiator. The results illustrated in Figure 7.2 show that accounting for the radiator thermal inertia has delayed the air outlet temperature increase, bringing the results closer to the measured values. However, the coolant outlet temperature increase has also been delayed, degrading the agreement with the experimental data. As mentioned in Section 6.2, the preliminary metal thermal inertia model implemented in this pilot project is approximate, and both a more exact implementation and further study of the transient effects of this model will be needed in the future. This run was terminated at t=393 s, due to a computational problem encountered by the STAR-CD solver. It was not clear if the problem was related to the new metal model, as the coolant and air temperatures remained stable throughout the calculation as illustrated in Figure 7.2. The cause of the problem will have to be investigated after the full implementation of the metal models in future work, if it persists. The modified radiator model including the metal models described in Section 6.2 was used, which needed additional input variables. The following input file for the radiator model (hxinput.dat) was used for this run.

```
PARALLEL
NAME(of the heat exchanger)
RADIATOR
METHOD   (1,2,3)
3
ATYPE
20
0
CTYPE
10
0
AUTYPE
21
ADTYPE
22
CUTYPE
11
CDTYPE
12
NACS
2368
NHXC
64
NHXA
6
TINC
333.15
TINA
296.75
CPC
3518.917
CPA
1007
URFQ
1.0
URFT
1.0
AREAA
0.2816
AREAC
0.00792
UCDAT
0.50
0.10,292.92,353.43,90.0
0.50,291.86,353.54,270.0
0.90,292.63,353.43,330.0
1.31,294.40,353.38,370.0
1.71,295.25,353.47,390.0
2.11,296.27,353.63,400.0
2.51,296.96,353.59,410.0
2.92,297.75,353.66,420.0
3.31,298.58,353.64,430.0
0.0,0.0,0.0,0.0
1.50
0.10,293.80,353.16,100.0
0.50,292.27,353.38,410.0
0.90,293.31,353.39,580.0
1.31,294.72,353.34,690.0
1.71,296.07,353.33,770.0
2.11,297.05,353.32,830.0
2.51,298.15,353.31,880.0
2.91,298.91,353.29,920.0
3.31,299.73,353.28,960.0
0.0,0.0,0.0,0.0
2.50
0.10,293.80,353.16,90.0
0.51,292.27,353.38,450.0
0.91,293.31,353.39,660.0
1.31,294.72,353.34,800.0
1.71,296.07,353.33,900.0
2.11,297.05,353.32,990.0
2.51,298.15,353.31,1060.0
2.91,298.91,353.29,1130.0
3.32,299.73,353.28,1180.0
0.0,0.0,0.0,0.0
3.50
0.10,294.59,353.32,100.0
0.51,292.54,353.42,460.0
0.90,293.40,353.43,690.0
1.31,294.42,353.38,850.0
1.71,295.65,353.37,970.0
2.11,296.98,353.37,1070.0
2.51,297.94,353.37,1160.0
2.91,298.73,353.32,1230.0
3.32,299.52,353.33,1300.0
0.0,0.0,0.0,0.0
0.0
ATUBE
0.000036
PTUBE
0.072
TTUBE
0.00022
CPM
961.4
XKM
225.0
RHOM
1000.0
AMATER
1
CMATER
2
ISTART
1
TRANM
```

Figure 7.2: Outlet Temperatures (02068026)

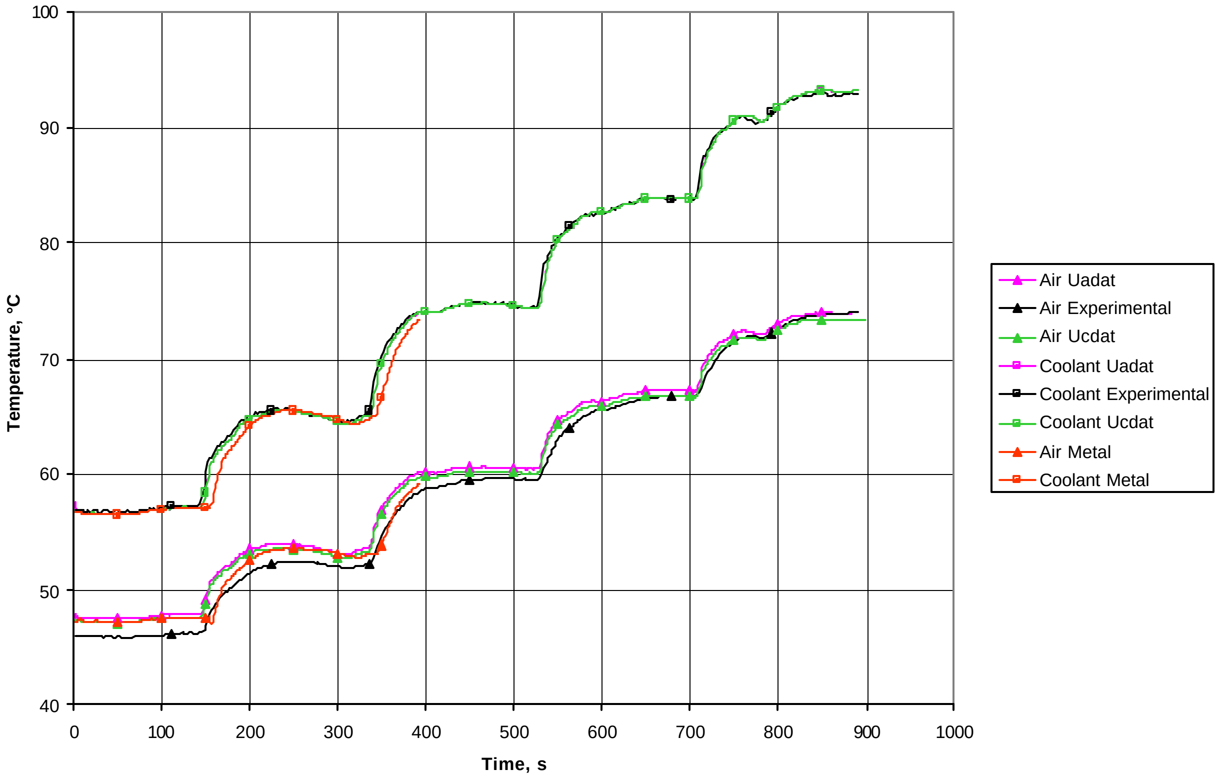
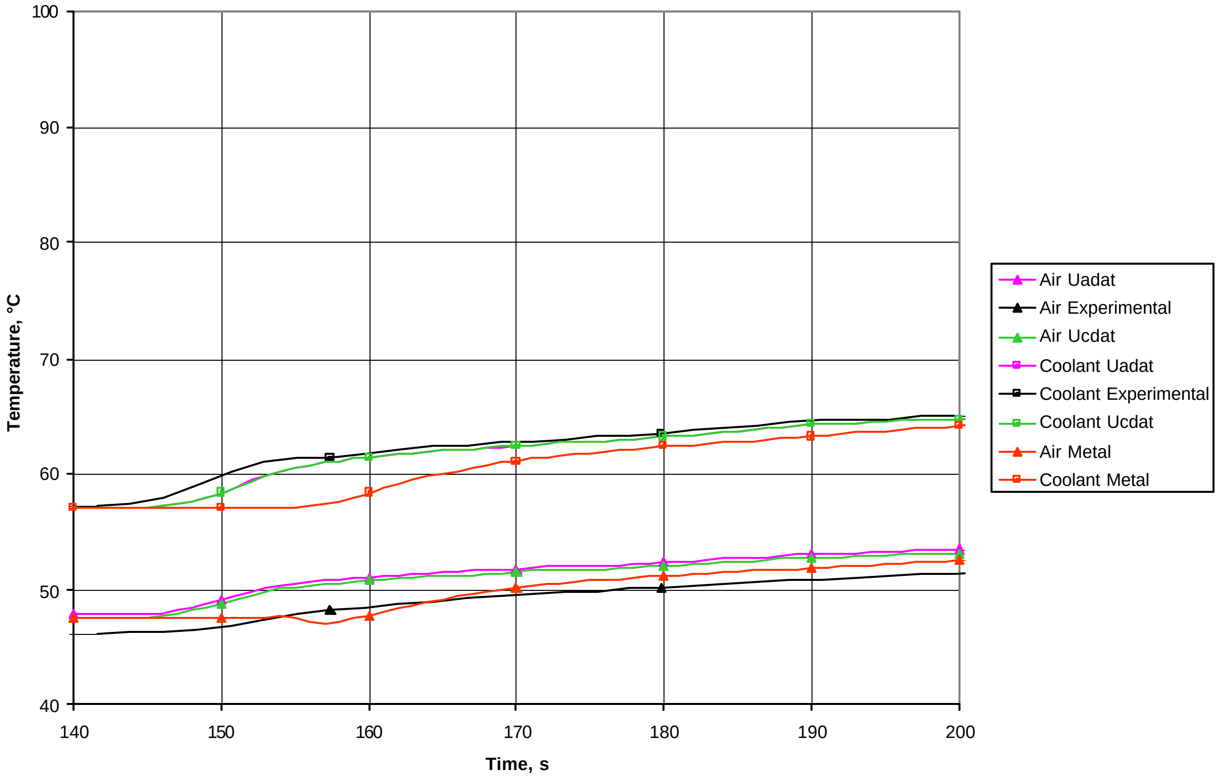


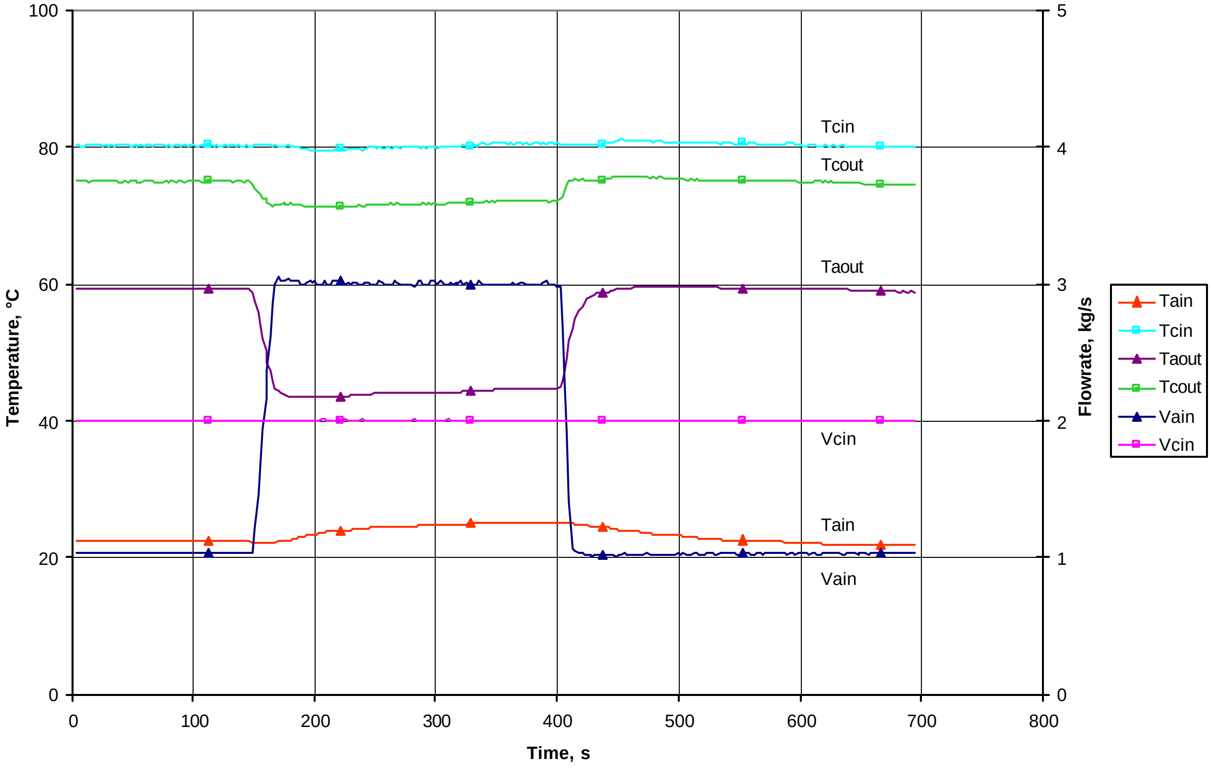
Figure 7.3: Outlet Temperatures (02068026)



7.2 Experiment that changed only the air inlet flow rate (02068022)

During this experiment the coolant inlet flow rate (Vcin) was held constant while the air inlet flow rate was changed as a function of time. Figure 7.4 shows the experimental data recorded during the experiment, which lasted approximately 700 s. The data indicates that the inlet temperatures (Tain and Tcin) also changed slightly during the experiment. Our analysis compared the calculated the average air outlet temperature (Taout) and average coolant outlet temperature (Tcout) with the corresponding measured values. We performed three separate analyses of this experiment, described below in Sections 7.2.1-7.2.3. The results of these analyses are shown in Figure 7.5. A more detailed view of the results during the first 200 s of the experiment are shown in Figure 7.6.

Figure 7.4: Experimental Temperatures & Flowrates (02068022)



For the steady first state run the following parameters were used:

	Coolant	Air
Temperature, °C	80.3	22.6
Flow rate, kg/s	2.002	1.030585
Specific Heat, J/kg-°C	3595.1	1007.0
Density, kg/m**3	1034.42	1.164

The hxinput.dat file shown in Section 7.1.2 was modified by changing these lines:

```
TINC
353.45
CPC
3595.1
```

7.2.1 Run using Quasi Transient Multiple Steady State Method

In this analysis we used the quasi transient multiple steady state method described in Section 6.1.1. The problem time was divided into segments, which are calculated by individual Star-CD runs, as summarized in Table 7.1. The problem was executed on the RAE Linux cluster computer, using 8 processors.

Table 7.1

time s	CPU time s	Iter.	Time/ iter	Inlet Air Flow Rate kg/s	Description
0	6604	558	11.8	1.03	Steady state that calculated both flow & temperature
1-148	420	148	2.8	1.03	Transient that calculated only temperature
148	1126	163	6.9	2.114	Steady state that calculated flows only
149-159	33.6	11	3.05	2.114	Transient that calculated only temperature
159	1159	155	7.5	3.005	Steady state that calculated flows only
160-402	699	243	2.9	3.005	Transient that calculated only temperature

All the transient runs used time steps of 1 second and did not calculate flow rate changes. The initial steady state run calculated both the flow and temperature fields, while subsequent steady state runs were used to update the flow fields, leaving the temperature fields unchanged.

In this run we focused on modeling the first air flow rate change, which was divided into 2 equal time steps. A steady state was run at the beginning of each time interval, changing the inlet air flow rate to the value corresponding to the end time of the step. As shown in Table 7.1, the first isothermal steady state was run at 149 s, changing the air flow rate from 1.03 Kg/s to 2.114 Kg/s. The second isothermal steady state was run at 159 s, changing the inlet air flow rate from 2.114 Kg/s to 3.005 Kg/s, the value at the end of the air flow rate change.

As shown in Figure 7.5, the calculated coolant outlet temperatures are in good agreement with the experimental values during the step change in the air flow rate. The effect of the two steady state changes can be clearly seen in the step structure of the coolant outlet temperature change. The agreement between the calculated air outlet temperature and the measured data is also good, but some oscillations can be seen in the calculated results at the time of restarts after the isothermal steady state runs. Whenever we restarted STAR-CD, it took several iterations to stabilize. The cause for these oscillations will have to be investigated in future work. They may be caused by the restart procedure and may be eliminated if STAR-CD were modified to automatically switch from steady state to transient mode without the user having to stop and restart the calculation. The differences between the computed and measured coolant and air temperatures during the transient can be further reduced by increasing the number of steady state runs during the rapid air flow rate change period, using the procedure outlined in Section 6.1.1. E.g., by performing 3 steady state calculations instead of two during the air flow rate change (at times 148 s, 159 s, and 169 s) which adjust the air flow rate to the average value during the step (1.57 Kg/s, 2.56 Kg/s, and 3.005 Kg/s respectively) the differences shown in Fig. 7.5 could be reduced by a factor of 2.

The total number of STAR-CD iterations (or steps) needed to calculate 402 s of experiment time was 1278, requiring a total of 10,041 s of CPU time when using 8 processors. A substantial amount of effort and time was required to manually schedule the sequence of steady state and transient runs outlined in Table 7.1. Future work should focus on automating this procedure and allowing the user to control it through the STAR-CD user interface.

7.2.2 Run using Quasi Transient Flow Similarity Method

In this analysis we used the quasi transient flow similarity method described in Section 6.1.2. This method allow the use of extended STAR-CD isothermal transients which can use relatively long time steps and accounts for changes in the coolant and air flow rates through changes implemented in the heat exchanger model during this work. This analysis used the same steady state run from the Section 7.2.1, followed by a single isothermal transient run from 0 to 694 seconds.

The hxinput.dat file was modified in the transient run to append the following lines, which activate the new models relevant for the quasi transient velocity similarity method:

```
TRANA
TRANC
```

The results obtained using this method agree well with the experimental data, as illustrated in Figure 7.5. The computed coolant outlet temperature changes are in close agreement with the measured corresponding values. The small delay of the calculated coolant temperature decrease (approximately 2 s) appears to be due to a delay in the measured air flow rate change. A close examination of the measured data found that the air inlet flow changes start approximately 2 seconds later than the outlet temperatures. After consultations with the DaimlerChrysler experimentalists, it was concluded that the air flow rate data is probably delayed by approximately 2 seconds due to the air flow meter response characteristics. This points to the need for a closer examination of the instrumentation response characteristics during transients in future work. The agreement between the calculated and measured air outlet temperatures is also quite good when using this method, although some discrepancies are observed at the end of both the air temperature decrease and increase periods. These discrepancies are similar to those observed when using the multiple steady state method described in Section 7.2.1 and thus are not likely to be due to the use of the velocity similarity method. The causes of these discrepancies will have to be examined in future work.

The CPU time needed for the transient part of this run using 8 processors was 2003sec for 694 time steps (694 s experiment time), for an average 2.89 seconds (CPU)/ seconds (experiment). This number can be compared to the multiple steady state run performance, which required a total of 3437 s to cover the first 402 s of the transient, for an average 8.55 seconds (CPU) /

seconds (experiment). The velocity similarity method provides both better agreement with the experimental data and better computational performance than the multiple steady state approach. Furthermore, this method can be used in STAR-CD without additional development effort, although some additional research to determine its characteristic response and limitations is highly recommended.

7.2.3 Run using Metal Model

In this analysis the new metal thermal inertia model was included while using the Quasi Transient Velocity similarity method described in Section 7.2.2. Due to the limitations of the preliminary metal model implementation described in Section 6.2, this run was executed on 2 processors. It required both new steady state and transient calculations. The goal was to evaluate the effect of the metal thermal inertia on the computed results. The results illustrated in Figures 7.5 and 7.6 show little difference between the outlet air and coolant temperatures calculated with the metal model and without it. These results are consistent with the nature of the transient experiment and the current implementation of the metal thermal inertia model. It is recommended that they should be reviewed in future work, after the implementation of a fully transient metal temperature calculation.

Figure 7.5: Output Temperatures (02068022)

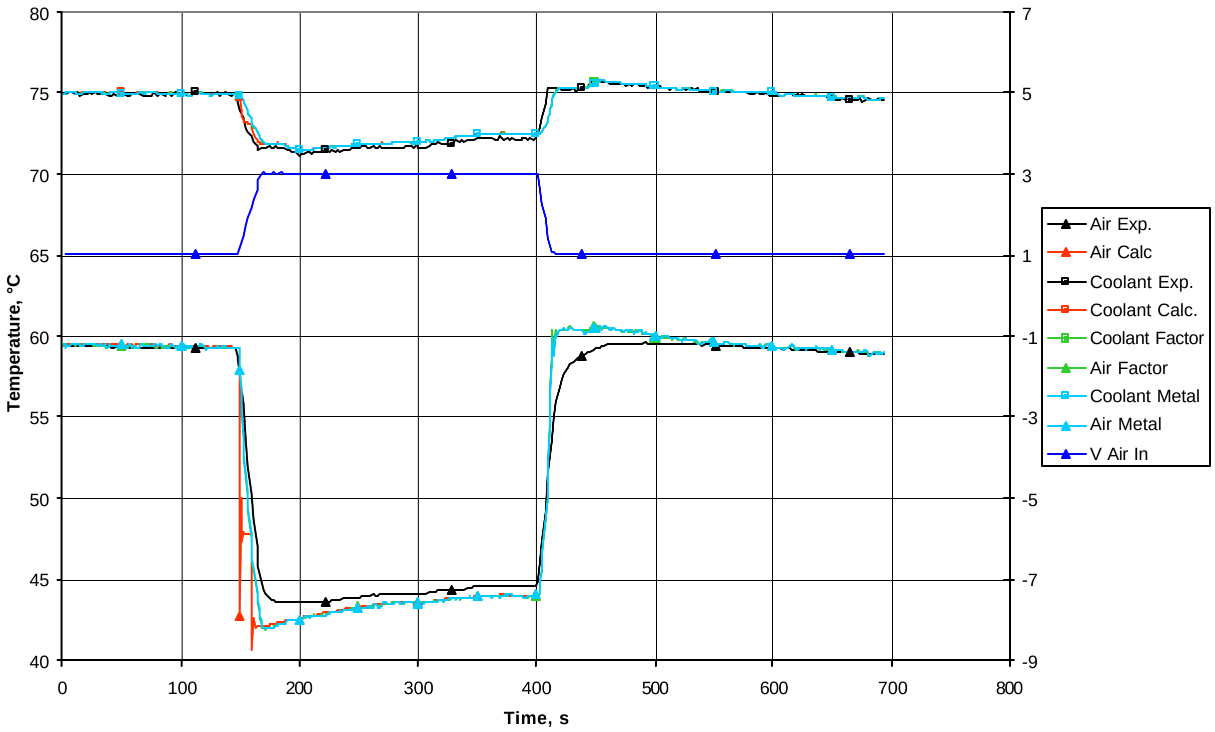
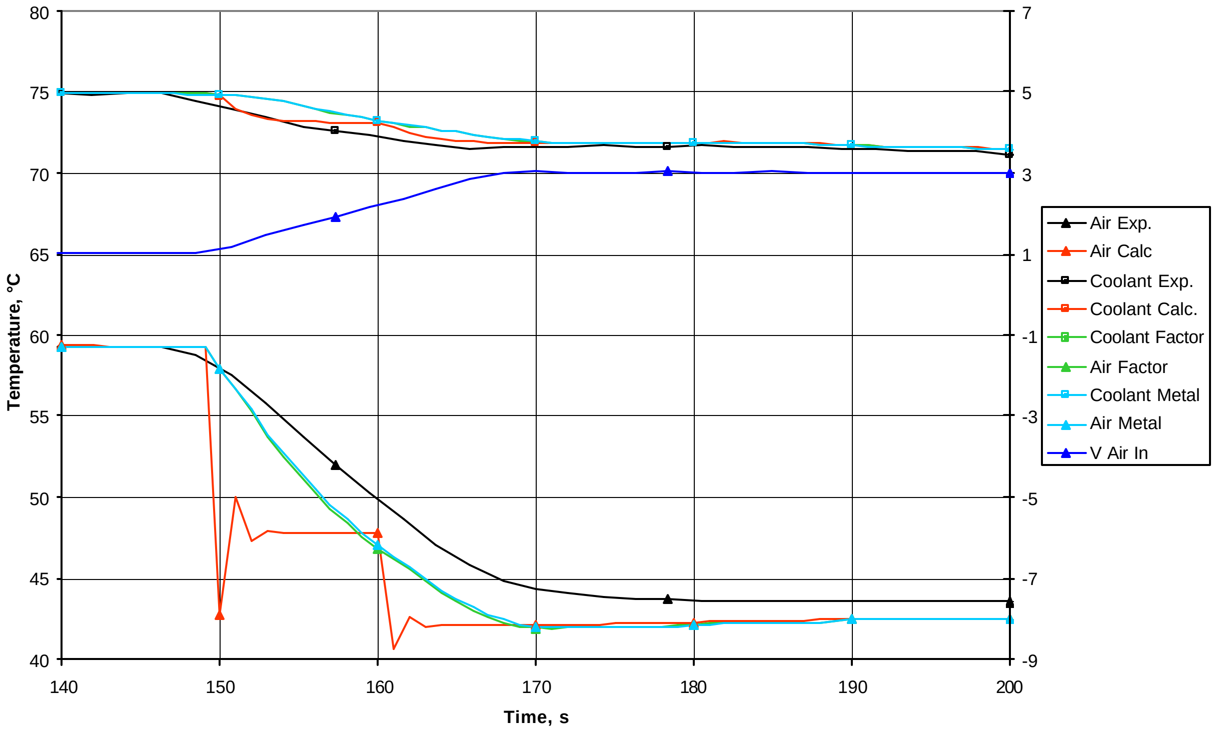


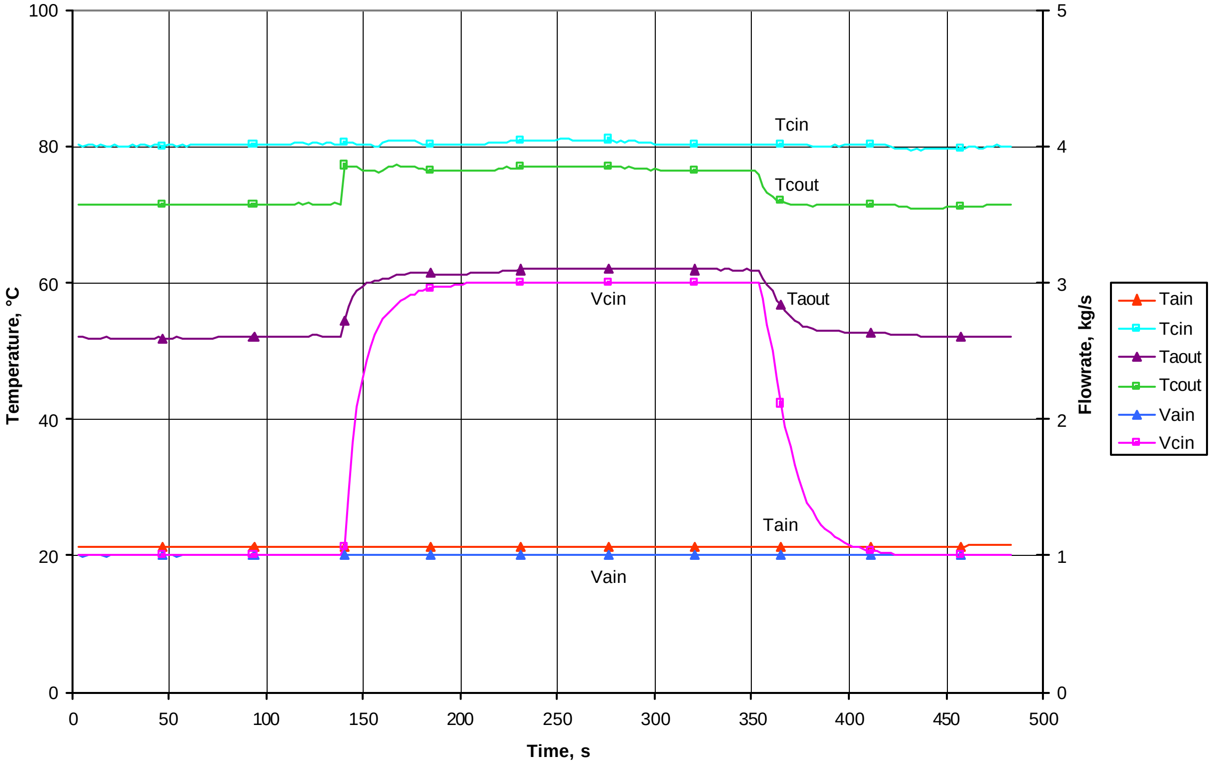
Figure 7.6: Output Temperatures (02068022)



7.3 An experiment that changed only the coolant flow rate (02068024)

During this experiment the air inlet flow rate (Vain) was held constant while the coolant inlet flow rate (Vcin) was changed as a function of time. Figure 7.7 shows the experimental data recorded during the experiment, which lasted approximately 500 s. The data indicates that the inlet temperatures (Tain and Tcin) changed little during the experiment. It is noted that the first rise of the measured coolant outlet temperature exhibits some unusual features, reaching the maximum value well before the coolant flow rate and then oscillating before settling down. This may be indicative of a problem related to the temperature probe response. Our analysis compared the calculated the average air outlet temperature (Taout) and average coolant outlet temperature (Tcout) with the corresponding measured values. We performed two separate analyses of this experiment, described below in Sections 7.3.1-7.3.2. The results of these analyses are shown in Figure 7.8. A more detailed view of the results during the first 200 s of the experiment are shown in Figure 7.9.

Figure 7.7: Experimental Temperatures & Flowrates (02068024)



For the steady first state run the following parameters were used.

	Coolant	Air
Temperature, °C	80.05	21.34
Flow rate, kg/s	1.097	1.006
Specific Heat, J/kg-°C	3594.3	1007.0
Density, kg/m**3	1034.6	1.164

The hxinput.dat file was modified from the previous experiment by changing these lines:

```
TINC
353.2
TINA
294.49
CPC
3594.3
```

7.3.1 Run using Quasi Transient Multiple Steady State Model

In this analysis we used the quasi transient multiple steady state method described in Section 6.1.1. The problem was divided into segments, which are calculated by individual Star-CD runs, as summarized in Table 7.2. The problem was executed on the RAE Linux cluster computer, using 8 processors.

Table 7.2

Time s	CPU time s	Iter.	Time/ iter	Inlet Coolant Flow Rate kg/s	Description
0	7569	629	12.0	1.001	Steady state that calculated both flow & temperature
1-138	526	138	3.8	1.001	Transient that calculated only temperature
138	5518	499	11.06	1.699	Steady state that calculated flows only
139-144	17.6	6	2.93	1.699	Transient that calculated only temperature
144	6003	540	11.12	2.397	Steady state that calculated flows only
145-151	22.7	7	3.24	2.397	Transient that calculated only temperature
151	4449	406	10.96	3.0	Steady state that calculated flows only
152-353	664	202	3.3	3.0	Transient that calculated only temperature

All the transient runs used time steps of 1 second and did not calculate flow rate changes. The initial steady state run calculated both the flow and temperature fields, while subsequent steady state runs were used to update the flow fields, leaving the temperature fields unchanged.

In this run we focused on modeling the first coolant flow rate change, which was divided into 3 time steps, with approximately equal changes in the coolant flow rate of 0.7,0.7,0.6 Kg/s respectively. A steady state was run at the beginning of each time interval, changing the inlet coolant flow rate to the value corresponding to the end time of the step. As shown in Table 7.2, the first isothermal steady state was run at 138 s, changing the air flow rate from 1.001 Kg/s to 1.699 Kg/s. The second isothermal steady state was run at 144 s, changing the inlet air flow rate from 1.699 Kg/s to 2.397 Kg/s. The third isothermal steady state was run at 151 s, changing the inlet air flow rate from 2.397 Kg/s to 3.0 Kg/s, the value at the end of the coolant flow rate ramp.

As shown in Figure 7.8, the calculated coolant outlet temperatures are in good agreement with the experimental values during the step change in the air flow rate, although the rate of increase of the measured data is higher. The calculated coolant temperature increase follows closely the coolant flow rate increase and, as mentioned above, it is possible that the more rapid increase of the measured coolant outlet temperature is due to the temperature probe response. The effect of the three steady state changes can be clearly seen in the step structure of the coolant outlet temperature change. The agreement between the calculated air outlet temperature and the measured data is reasonably good, but some oscillations can be seen in the calculated results at the time of restarts after the isothermal steady state runs. Whenever we restarted STAR-CD, it took several iterations to stabilize. The cause for these oscillations will have to be investigated in future work. They may be caused by the restart procedure and may be eliminated if STAR-CD were modified to automatically switch from steady state to transient mode without the user having to stop and restart the calculation. The air outlet temperature calculated by the initial steady state calculation is seen in Figures 7.8 and 7.9 to be about 2 C below the corresponding experiment measurement, which suggests. At the end of the coolant flow increase period, the calculated air temperature is higher than the experimental corresponding temperature by about 3.5 C. This temperature difference decreases in time, but the calculated air outlet temperature remains higher than the measured value by about 2.5 C. A possible cause for these discrepancies may be heat transfer coefficients which are not accurate enough for the steady state and transient experiment conditions. Future work should investigate the reason for the discrepancies between the calculated air outlet temperatures and measured data.

The total number of STAR-CD iterations (or steps) needed to calculate 353 s of experiment time was 2427, requiring a total of 24,770 s of CPU time when using 8 processors. A substantial amount of effort and time was required to manually schedule the sequence of steady state and transient runs outlined in Table 7.2. Future work should focus on automating this procedure and allowing the user to control it through the STAR-CD user interface.

7.3.2 Run using Quasi Transient Velocity Similarity Model

This analysis used the velocity similarity method to analyze the transient experiment with variable coolant inlet flow rate. This run used the same steady state run as the analysis described in Section 7.3.1 followed by a single transient run from 0 to 483 seconds.

The hxinput.dat file was modified in the transient run to append the following lines:

```
TRANA
TRANC
```

The results obtained using this method agree well with the experimental data, as illustrated in Figure 7.8. The computed coolant outlet temperature changes are in close agreement with the measured corresponding values, except for early rapid rise in the measured coolant temperature which, as pointed out above, may be due to the temperature probe response. The agreement between the calculated and measured air outlet temperatures is also quite good when using this method. The difference between the initial steady state calculate air temperature and the measured value remains the same as described above in Section 7.3.1, as the two analyses shared the same steady state. The air temperature increase follows the measured data closer that the results presented in Section 7.3.1, but at the end of the coolant flow rate increase period the calculated air temperature is higher than the measured value by about 1.5 C. This run was terminated at step 416 of the transient with an error message from the STAR-CD solver. It is not clear if this error is related to the new methods and models implemented in the heat exchanger routines, as all heat exchanger results remain stable until the end of the calculation. The source of this problem will have to be examined in future work, if it persists after the integration of the new models in STAR-CD by Adapco staff.

The CPU time needed for the transient part of this run using 8 processors was 769 seconds for 416 time steps (416 s experiment time), for an average 1.85 seconds (CPU)/ seconds (experiment). This number can be compared to the multiple steady state run performance, which required a total of 17201 s to cover the first 353 s of the transient, for an average 48.7 seconds (CPU) / seconds (experiment). The velocity similarity method provides both better agreement with the experimental data and better computational performance than the multiple steady state approach, confirming the results presented in Section 7.2.2.

Figure 7.8: Output Temperatures (02068024)

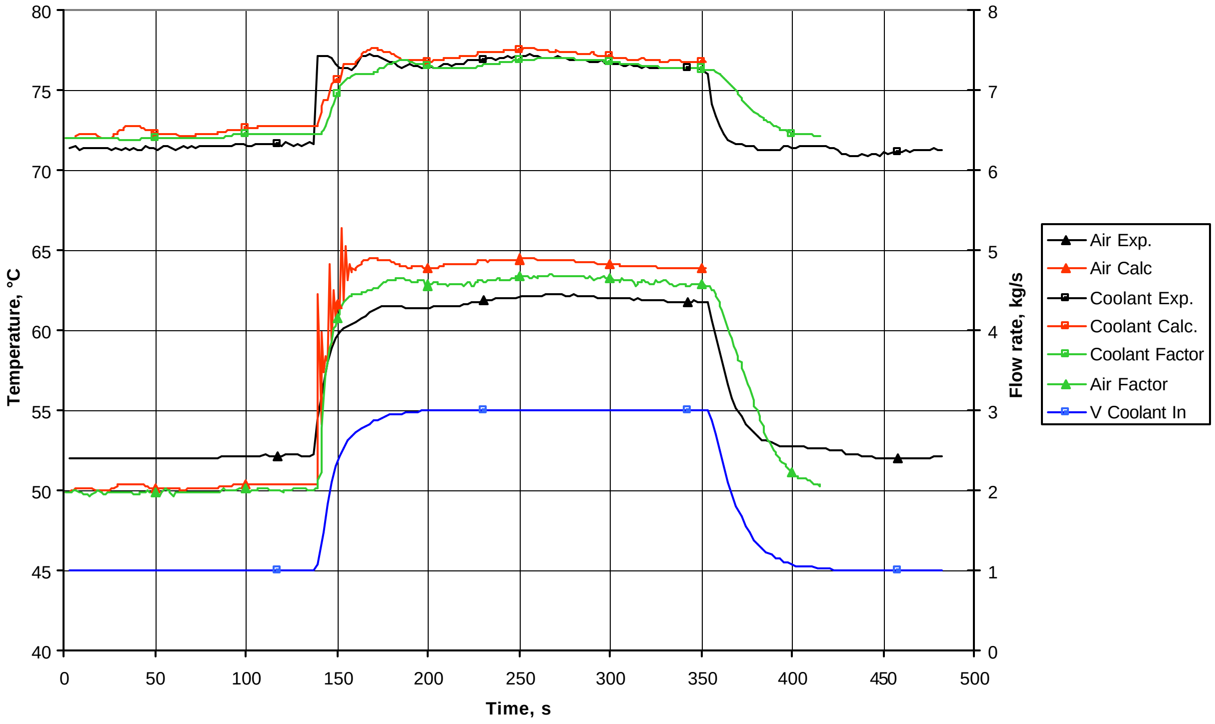
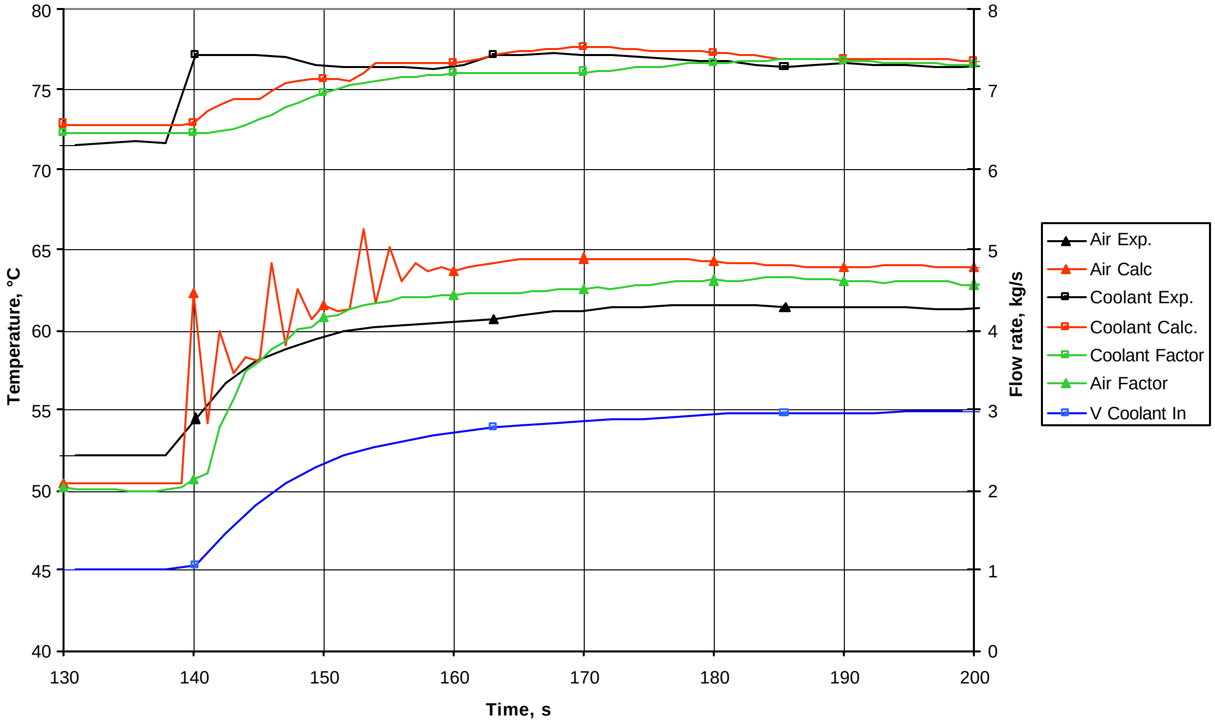


Figure 7.9: Output Temperatures (02068024)



8. Software implementation

This section provides on overview of the software changes made in order to implement the methods and models described in Section 6. During this work we obtained from Adapco a copy the heat exchanger model source and all changes were made exclusively to the heat exchanger model routines and common blocks. To activate the new features, the user must use the key word MTHD with value 3 and the key word UCDAT, described below, in the heat exchanger input file. The new methods and models affect the rest of the STAR-CD calculations mainly by modifying the values of the volumetric heat sources **qsrchx** passed from the heat exchanger model to the coolant and air energy equations, which are solved by the STAR-CD solver.

8.1 New UCDAT option

A new option UCDAT has been implemented in the modified routine **hxinit1** that allows the input of multiple coolant flow rates and corresponding radiator heat exchange performance curves. This option is necessary for transient calculations where the coolant flow can change. A previous option, QDATA, allows the input of heat transfer coefficients for multiple pairs of coolant and air flow, and calculates the heat transfer coefficients using a surface fitting method. As mentioned in Section 5, this approach does not ensure that the correct heat transfer values are retrieved for all values on the performance curves. The new option is requested with the key word UCDAT, followed by a coolant flow rate value and a set of air flow values and corresponding heat transfer coefficients. A line beginning with a 0.0 for the air flow rate indicates the end of the current coolant flow rate data, and should be followed by the next coolant flow rate value. A line beginning with 0.0 for the coolant flow rate indicates the end of data for the UCDAT option. The UCDAT option also sets the new flag **mcflag=1**. This flag is used later in the modified **heatex** routine to determine which heat transfer routines should be called, as explained in the next section. The UCDAT option also sets the option **mthopt=3**. This value is used in the new heat transfer routines to calculate the heat transfer coefficient using the correct values of the coolant and air flow, by calling the new routine **ualocal5**. The routine **ualocal5** calculates the local heat transfer coefficient by linear interpolation between the performance curves for various coolant flow rates, ensuring that the calculated heat transfer coefficients are in agreement with the input data.

8.2 New heat transfer routines

Two new heat transfer routines have been developed, using the routine **hxmthd3** as starting point. The new routine **hxmthd32** is called for transient calculations where the effect of the radiator thermal inertia is neglected, while the new routine **hxmthd33** is called when the the user wants to study the effect of the radiator metal mass on transient effects. The decision to call the new routines is made in the modified **heatex** routine. On the branch mthd=3, if the flag mcflag=1 then one of the new heat transfer routines is called. The decision to call hxmthd32 or hxmthd33 is based on the value of the flag **imetal**, discussed in section 8.4. Both hxmthd32 and hxmthd33 will calculate the current local coolant and air flow rates using the corresponding velocities received from STAR-CD and then call the new routine ualocal5 to determine the correct local heat transfer coefficient. This heat transfer coefficient is then used to calculate the volumetric heat sources qsrchx(air) and qsrchx(coolant) which are sent to the STAR-CD solver.

8.3 Implementation of quasi-transient velocity similarity option

The use of the quasi-transient velocity similarity methodology described in Section 6.1.2 is requested using the key word TRANC for the variable coolant flow rate and TRANA for the variable air flow rate. The use of the key word TRANC causes the modified routine **hxinit1** to set the flag **mtrcflag=1**. Similarly, the use of the key word TRANA causes the modified routine **hxinit1** to set the flag **mtraflag=1**. These flags are checked in the new heat transfer routines hxmthd32 and hxmthd33, both of which implement the quasi-transient velocity similarity option. Each time step, a coolant flow rate multiplier **cmult_tr** and an air flow rate multiplier **amult_tr** are defined. The multipliers are set to 1.0 if the corresponding flag is not 1. Otherwise, the multiplier is calculated using the new function **radinp** which receives the current time as an argument and a second literal argument which indicates if the coolant or air flow rate multiplier is needed. To use the function **radinp** the user must supply the input file **radinp.csv**, that contains on each line, the time, coolant inlet temperature, air inlet temperature, coolant inlet velocity, and air inlet velocity. The coolant multiplier is then applied to all coolant velocities to determine local coolant flow rate, and a similar approach is used to determine the local air flow rate. These adjusted flow rates are used to determine the local heat transfer coefficient, by calling the new routine **ualocal5** described in Section 8.1. It is noted that the actual local velocities calculated by STAR-CD are not changed during this procedure. The heat transfer coefficient adjusted for changes in the air and coolant flow rates is then used to determine the volumetric heat fluxes **qsrchx(air)** and **qsrchx(cool)** used in the air and coolant energy conservation equation solved by the STAR-CD solver. However, before sending these heat sources to the STAR-CD solver, another important adjustment is made. Because the actual velocities used by the STAR-CD solver have not been changed to reflect the transient flow rate changes, we adjust the volumetric heat fluxes so that the STAR-CD solver will calculate the correct temperature change for the transient flow rate. The volumetric heat flux **qsrchx(air)** is divided by the factor **amult_tr** and **qsrchx(cool)** is divided by **cmult_tr** before sending these values to the STAR-CD solver.

8.4 Implementation of radiator thermal inertia models.

Several new key words have been added to hxinit1 routine which reads the heat exchanger input file. These key words and the corresponding values must be added to the heat exchanger input file:

CPM = metal specific heat, J/(Kg*K)

XKM = metal conductivity, W/(m*K)

RHOM = metal smeared density, Kg/m3 (physical density smeared over the radiator cell volume)

ATUBE = tube cross-sectional area, m2

PTUBE = tube perimeter, m

TTUBE = tube thickness, m

While all these inputs are necessary, the metal models are activated when the value RHOM is greater than 0. When this is the case, the modified routine **hxinit1** sets the flag **imetal=1**. This flag is used in the modified **heatex** routine to determine which of the new heat transfer routines is called. If **imetal=1** the routine **hxmthd33** which includes the radiator metal models is called, otherwise **heatex** calls the routine **hxmthd32**. To indicate that a transient calculation is performed using the metal option, the user must add the key word TRANM to the heat exchanger input file. This will cause **hxinit1** to set the flag **lhtrans=1**. The new heat transfer routine **hxmthd33** check the flag **lhtrans** to determine if a transient or steady state calculation is needed. Before leaving the routine hxmthd33, the metal temperatures are saved in a file. This is a temporary solution, needed because the **tmetal** array is not yet part of the STAR-CD data structures and thus is not automatically saved for restarts. The user must select the appropriate name and path for this file before running the steady state or transient calculations. For restarts that include the key word TRANM, indicating a transient, the hxinit1 routine will try to read the metal temperatures from a specified file. The user must make sure that the correct name, pointing to the file created during the steady state or previous transient is specified in the **hxinit1** routine. One other limitation caused by the fact that tmetal is not yet part of the STAR-CD data structures is related to multiple processor calculations. The current implementation of the radiator thermal inertia models has been developed and tested on two processors only. This limitation will be removed in future work by including the array tmetal in the STAR-CD data structures which are subject to the data decomposition and file management needed for parallel computing systems.

9. Conclusions

This work has explored new strategies and developed models which extend the capabilities of an existing established CFD code, STAR-CD, allowing the car manufacturers to analyze the impact of transient operational events on the underhood thermal management by exploiting the computational efficiency of modern high performance computing systems.

In particular, the project has focused on the CFD modeling of the radiator behavior during a specified transient. The 3-D radiator calculations were performed using STAR-CD, which can perform both steady-state and transient calculations, on the parallel computing system available at ANL in the NE Division. The results of initial calculations, presented in Section 5, show that performing conventional STAR-CD transient calculations over time intervals of hundreds of seconds would require prohibitively large computing times, even on powerful parallel computing platforms. Two quasi-transient computational strategies which require considerably shorter computing times were developed to allow the modeling of transient phenomena over extended time periods using STAR-CD. Several experiment analyses using radiator transient experimental data provided by Adapco and DaimlerChrysler were performed in order to test and validate the new quasi-transient methods and models. The results of these calculations have been compared with the experimental data and the results presented in Section 7 indicate that both quasi-transient strategies developed during this work provide results that are in reasonably good agreement with the experimental data.

The quasi-transient strategy based on velocity similarity assumption provides a good solution to the modeling of transients involving both flow and temperature changes. Its implementation is close to the final form, but future work should explore the validity limits for the velocity similarity assumption when large flow changes occur in complex geometries or when multiple independent inlet orifices are present for the same fluid. The quasi-transient strategy based on multiple steady state calculations also gave promising results, but its full implementation in STAR-CD will require the automation of the procedure. Future work should develop an automated procedure that will allow the user to request a quasi-transient calculation composed of a sequence of steady state and transient calculations as described in Section 6. The user should be able to specify the times of the steady state calculations, or allow the code to automatically select these times. The possibility of combining elements of this strategy with the velocity similarity strategy should also be explored. I.e., several steady state calculations could be performed at selected time intervals when using the velocity similarity strategy, to adjust the flow fields to current values if necessary.

Several new capabilities and models have been developed which directly support the modeling of transient thermal-hydraulic phenomena with STAR-CD. They include the ability to model the thermal inertia effects of the heat exchanger mass and the ability to analyze transients that involve coolant flow rate changes. The models involving the heat exchanger thermal inertia will require future work, both in terms of implementation and model development. The data structures necessary for this model should be included in the STAR-CD data structures and the models should be refined and further validated as indicated in Section 6.

The results of this work open the way for the development of a STAR-CD based CFD tool for the transient analysis of underhood thermo-hydrodynamic events, which will allow the integrated transient thermal analysis of the entire cooling system, including both the engine block and the radiator, on high performance computing systems.

10. Acknowledgments

The authors would like ot thank Mr. Bob Brewster from Adapco for helpful discussions on the STAR-CD heat exchanger model and Mr. Walter Bauer of DaimlerChrysler for helpful discussion on the radiator experiments and experimental data.