Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

An intelligent user interface for computational fluid dynamics software

J. Ewer, M. Petridis, D. Cowell & B. Knight
Department of Mathematics, Statistics and Computing, The University of Greenwich, London SE18 6PF, UK

ABSTRACT

The development of advanced and successful numerical modelling packages for Computational Fluid Dynamics (CFD) within recent years has led to their widespread use in engineering and industrial environments. Increasingly this has led to a situation where package users have little or no specialist knowledge of the underlying physical principles upon which CFD packages are based. Generally users also have limited knowledge of the numerical software itself, or how to obtain accurate results efficiently. However this knowledge can often be vital to the correct usage of the software for producing reliable simulation data.

This paper describes an on-going research project to incorporate such expert knowledge into CFD software. The special problems encountered are those of interfacing knowledge based components with numerical routines via a blackboard[1] architecture. The design of appropriate user interfaces and the provision of dynamic run-time graphical displays and solution monitoring are also problematic areas. The approach adopted here allows the interaction of knowledge sources, numerical solution routines, display tools, and pattern recognition tools. The architecture of the system is described, together with experience of its capabilities and benefits to CFD simulation.

INTRODUCTION TO CURRENT RESEARCH

The widespread use of Computational Fluid Dynamics by novice users and the extensive demands placed on CFD techniques has led to a situation where traditional batch mode packages are no longer satisfactory. This paper outlines the merits of using a Knowledge Based System (KBS) approach to CFD and proposes an alternative architecture for interactive simulations. The design and development of a prototype system is considered with emphasis placed on reliability and overall efficiency rather than speed performance.

Prior research into intelligent front ends to numerical simulation packages has indicated the feasibility of KBS techniques for CFD. Much of the existing research treats the numerical component as a batch process. The system proposed by this paper provides interactive control of the CFD numerical process. The current research is based on FLOWES[2], a prototype, inference controlled CFD code. The continued development of the FLOWES architecture has indicated problems associated with existing approaches for developing and using CFD codes. This paper details solutions and proposes a new approach to CFD simulations.

BACKGROUND TO RESEARCH

A KBS is a system that uses human expertise and knowledge, usually in the form of heuristic rules, to reason about specific application areas. The application areas have to be limited because computational demands for inferencing are far higher than those for numerical calculation. Knowledge is elicited from human Experts and then encapsulated in some suitable form within an application specific database, often called a Knowledge Source (KS). In the work described here the application area is that of fluid dynamics simulation. The encapsulated knowledge is used to make decisions about the solution strategy adopted for a particular simulation.

Much of the previous research has been devoted to KBS support for the specification of a simulation. The techniques utilised have the advantage of ensuring that the simulation is specified correctly, completely and consistently. Such support is limited because there is no dynamic control of the numerical simulation component. FLOWES makes use of a blackboard architecture which provides a flexible and extensible framework within which numerical and knowledge based components can interact cooperatively.

There has been limited research into KBS control of a numerical code[3]. This was restricted to the adjustment of a limited number of parameters by hard-coded conditional rules. This integration of rules into the source code of an application suffers from a lack of flexibility and has only very limited inference capability. There is little possibility of conflict resolution and the monitoring capabilities are limited. The approach adopted did not provide any interaction for the system user and the numerical component was still, essentially, a batch mode process.

PREVIOUS RESEARCH

FLOWES[2] is a two dimensional heat transfer code that provides KBS support for problem set-up, mesh generation and solver selection. The system also provides dynamic solution control by the integration of a KBS component[7] into the numerical solver. There have been a number of problems during the development of FLOWES due to the nature of the control structure required. A blackboard architecture[8] was chosen for truly interactive solution control. The severe restrictions of existing CFD codes to external interactive control have necessitated the re-implementation of the numerical component to provide KBS control capabilities.

The FLOWES system is currently under test for reliability of the inference techniques and for suitability of the design. The results, to date, are promising and suggest that the integrated blackboard approach is suitable for CFD. FLOWES currently has a number of limitations, namely:-

- Support for two-dimensional un-structured meshes only.
- No solution for fluid flow properties.
- No user interface for operations controlled by the KBS component.

These limitations have restricted the current research because the prototype system needs flow simulation capability and three-dimensional meshes.

Other researchers are investigating KBS techniques for CFD. Jambunathan, Lai, Hartle and Button are developing an Intelligent Front End (IFE) for PHOENICS[4], a well known commercially available CFD code. This IFE is designed to support problem set-up and specification during an interactive question and answer session at a computer terminal. Expertise is used to support the set-up and to pre-set the parameters and switches used by the PHOENICS system[5]. The output from the IFE is a simulation specification file that can then be used by the CFD code to run that problem. The CFD code is still treated as a numerical "black-box" since the user has little control of the computations once the processing has started.

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Finn, Hurley and Sagawa are developing AI-DEQSOL[6]. This is an Artificial Intelligence (AI) system for the numerical simulation of engineering problems that are defined by partial differential equations. The system comprehensively supports the specification of the mathematical model and the boundary conditions using an IFE. However the numerical solver is again used as a batch process. The output from the IFE is generated simulation code which is then processed by the DEQSOL simulation system.

AIMS OF CURRENT RESEARCH

The previous research has demonstrated the potential benefits of KBS techniques but little research has been conducted into comparison between KBS support for CFD and fully interactive CFD. There are currently no CFD packages that support dynamic solution control by either a KBS or the application user. The current project will have an intuitive Graphical User Interface (GUI). The KBS can then be switched off so that meaningful investigations into the relative benefits of a KBS, can be conducted.

Knowledge will be acquired for two example CFD application areas, namely external vehicle aerodynamics and fire simulation modelling. The CFD numerical component will have to support three-dimensional meshes, body fitted coordinates and solve turbulent and elliptic flows[9]. These requirements will affect the overall system design criteria.

Much of the existing CFD research is restricted to use within specific codes. It is intended that an application framework for KBS techniques will be developed during the current research. The data and control architectures will be sufficiently flexible to allow for the integration of other research work.

The limitations of many existing CFD codes is, in part, due to the techniques employed in their development. The current research is intended to highlight, to CFD developers, the benefits of using sound software engineering principles[10]. Future research and code maintenance would be greatly facilitated by the use of a software design methodology. CFD codes would be easier to re-use and modify if supported by comprehensive and accurate documentation. These techniques are rarely used by most CFD code developers.

THE PROBLEMS OF CFD RESEARCH

There are problems associated with the usage of many of the CFD codes currently available. Many of these difficulties stem from the complexity of the underlying code[11],[12] and the vast scope of fluid flow type simulations. The numerical approximation techniques used for CFD are vital to obtaining reliable solutions and users must currently be aware of the limitations and restrictions of the various techniques available.

CFD techniques have been used successfully for over 20 years but still lack the reliability and robustness of some other disciplines (e.g. CAD and structural Finite Element Analysis applications). The use of CFD codes is still only viable for CFD experts. These experts need to be well versed in both the physical principles upon which the codes are based and the quirks and implementation details of specific codes. These joint requirements mean that there are few users who can specify and run a simulation successfully in one attempt.

The complexity of the relationships between physical phenomena and the numerical models[9],[11] mean that there are many parameters to set for even relatively simple simulations. CFD simulations also have the problem that the configuration required for an accurate solution depends on the results of the simulation but these are obviously unknown at the start of the simulation. To overcome this problem CFD users frequently run quick coarse-grid simulations to get a "look and feel" of the solution. A new simulation can then be run with a more appropriate configuration. Unfortunately the sensitive nature of the CFD discretization techniques can mean that important features are missed in coarse-grid runs and hence the configuration could still be invalid.

The numerical processing of CFD is generally a batch mode process which follows the traditional cycle of specifying the simulation, running a batch mode solver and finally interpreting the results. The extended duration of many simulations (e.g. 64 hours for a typical three dimensional fire simulation on a workstation) is such that this batch mode of processing can be extremely inefficient because:-

- The results can be found to have diverged early in the simulation.
- There can be localised or wholly erroneous results.
- Important features can go undetected (e.g. Plumes above fires).
- The results can be unrealistic or physically impossible because of instabilities for a given configuration.

Many batch mode CFD codes provide only minimal information about the intermediate state of the solution and even where potential

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

problems are detected there is generally no way to correct the situation without restarting the simulation with a new configuration.

The poor monitoring of the intermediate solution status and the lack of run-time interaction means that problems can only be detected after the simulation. This is clearly unsatisfactory for novices who need help and guidance to get realistic results. CFD is also used widely in design and safety applications where solution reliability and the accuracy of results are of paramount importance. The fact that CFD techniques are now widely available and are used more extensively can only exacerbate this problem.

STRUCTURE OF A KBS FOR CFD

The proposed structure for a CFD application that supports dynamic interaction and knowledge based reasoning is simplified to the following five components:-

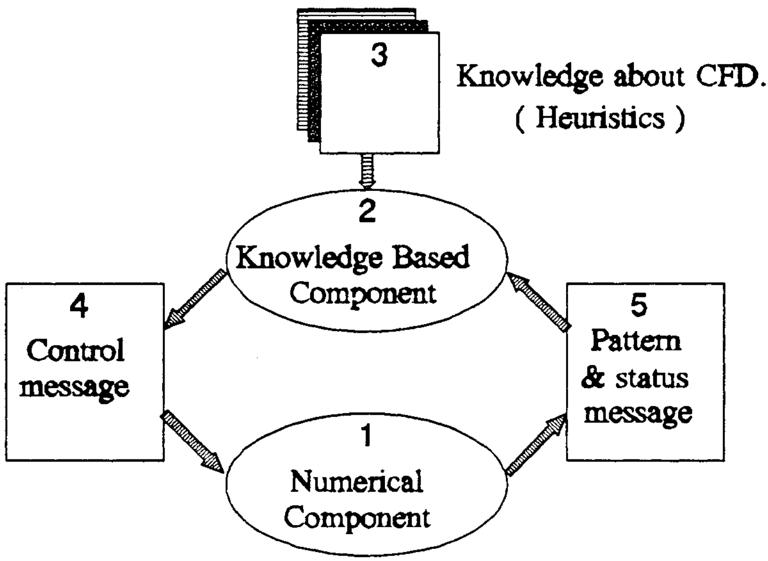

Figure 1

1) Numerical calculation component.
2) Knowledge Based reasoning component.
3) Knowledge Source is a database of heuristic rules.
4) Control message for the numerical component.
5) Pattern and solution status message for the KBS.

The computational and reasoning components have been deliberately separated because the implementation strategies for each component have very different requirements. These components are not suited to combined development within one code module. The KBS development and testing benefits from the use of a KBS shell tool or a language more suited to inference techniques (e.g. Prolog) whereas the numerical component is most efficiently implemented in a procedural language (e.g. FORTRAN or C++).

The numerical component can be viewed as a traditional CFD code but has enhanced control so that external agents can alter the numerical processing strategy. This control could be as simple as modifying one parameter or as complex as mesh refinement or a change of cell processing order. The requirement for such control flexibility means that the numerical component must have an open data architecture so that prescribed changes can be easily effected. There is also a need for the flexibility to add code modules for future development. The reasoning component must be aware of the solution status and features during the numerical processing. This necessitates the provision of monitoring and communication routines that will be embedded within the numerical component.

The knowledge based component is a pattern matching inference engine that searches through application specific heuristic rules to find appropriate actions. These are based on the intermediate solution status and current solver parameters. The KBS formulates a new control message that contains updated configuration settings and the new processing strategy. This control information is placed on the blackboard so that it is available for the numerical component to interpret. KBS components are notoriously slow at matching large amounts of data with patterns within a knowledge base. To overcome this limitation the status information sent to the KBS is in summary form only. This means that raw numerical data is never sent from the CFD component.

The knowledge for the inferencing is in the form of heuristic rules (rules of thumb) stored in a KS database. For ease of modification and maintenance, the database of rules is not hard-coded into any executable component. Inference driven searching of rules is made more efficient by defining rule-sets. Reasoning is then a layered process that efficiently locates relevant rules from the categorisation of the rules and rule-sets.

The pattern and status message contains the current status of the numerical solution. The detection of patterns is performed within the numerical component and the message then refers to a pattern by type, location and possibly extent (e.g. recirculation centred on cell number 48). The solution status is in summary form to keep communications overheads to a minimum and to prevent accidental modifications of intermediate solution data.

Finally the control message is the new solution strategy based on the KBS reasoning about the previous solution status. This is interpreted by the numerical component to set control parameters and to modify the solution strategy.

BENEFITS OF KBS APPROACH

The proposed system provides dynamic control based on intermediate solution status information. This is more efficient than running a batch mode solver because errors and potential problems can be detected and, where possible, corrective action taken. The overheads of inferencing and communications are mitigated by the enhanced solution robustness and overall system performance, particulary for large simulations.

The control and data architectures required for a KBS driven system will provide a flexible and extensible framework for the addition of future CFD research. If required, the KBS component can be switched-out so that reasoning and control is left to the system user. Even this is a considerable advance on batch mode operation because a knowledgeable expert can direct the numerical component based on the status messages (and data visualisation) during the simulation.

The monitoring routines necessary for summarising the intermediate solution results will help to ensure that the results of a simulation are reliable. The potential for the development of monitoring routines is vast. Initially the monitoring will detect divergence, unreasonable values and specified solution features that may need special numerical treatment (e.g. recirculating flows). The fact that these features could be detected and have prescribed numerical treatments could help to stabilise the solver under potentially error prone conditions.

One other major benefit to CFD is that the knowledge bases will form a comprehensive repository for expert rules about specific application areas. These repositories can easily be extended or modified as new techniques become available and thus provide future support for novice and expert users alike. The dynamic control will also need to be researched as it is a new technique that is not currently available in any other CFD system.

RESEARCH PROBLEMS AND SOLUTIONS ADOPTED

The need for pattern recognition and feature detection is problematic since patterns within raw numeric data sets are difficult and time consuming to search for. The techniques employed are also potentially unreliable. These problems can be overcome by allowing the user to visually detect patterns and features within the data set. The presentation of the data would give a number of pattern options for the user to match with the visual display. This means that the user need not know the consequences of the features found and can be unaware of the underlying physical principles of the phenomena. The user is merely applying visual pattern matching detection more efficiently than a data searching code module could.

Knowledge acquisition is a complex and error prone task because the system developer and the domain expert are often from different disciplines. This can lead to semantic errors in the developers conceptual model of the application. This leads to un-rewarding interviews and poor knowledge elicitation. There are also problems of contradictory knowledge and actions that require some form of conflict resolution. Knowledge may also be specific to one particular, but critical, area that never comes up in an interview.

The proposed solution is to use the prototype KBS system, without rules, to elicit knowledge from CFD experts. Experts would use the system to solve specific simulations and the rewarding actions taken can be used as heuristic rules for the developing system. There is a requirement for a good graphical user interface and for comprehensive run-time visualisation so that the expert user can detect potential problems, and moreover be able to apply corrective techniques. As previously mentioned the categorisation of rules into rule-sets, within a KS database, will facilitate efficient inferencing.

One major benefit of a database or rules is the ability to test and debug the KS without modification of the numerical module. The main reason for such an easily modified KS is that CFD codes are not currently used interactively. This means that the heuristic rules in the KS will need to be revised frequently as the dynamically controlled system is used for knowledge acquisition.

The following example rules indicate the potential for inference driven control. The first is a mesh cell checking rule. This rule detects

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

cells that would not allow continuity (Figure 2).

Categorisation:-

Rule set: Grid_rules
Application: Generic (for all simulations)
Related to: Cell_faces

Heuristic rule:-

Rule: If cell has less than 2 unblocked faces and solving for flow then conclude cell error.
Action: Re-mesh locally at the problematic cell.
Reason: Continuity requires that a cell has both an entrance and an exit face.

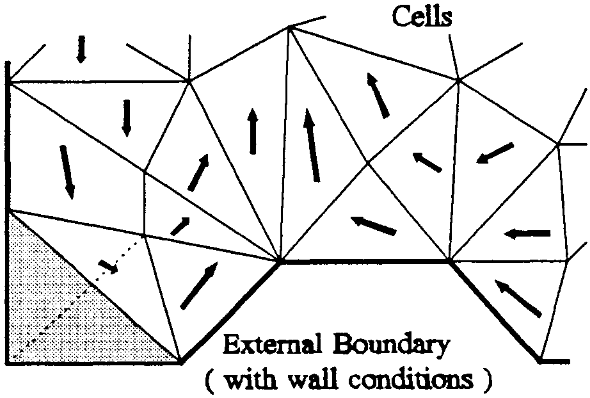

Figure 2

The shaded cell is in error because it only has one face open to flow and hence continuity cannot be satisfied. This is an example of a grid checking rule that would only be applied after grid generation or after grid refinement. The dotted line (bottom left corner) is one possible re-mesh solution.

The following rule provides dynamic control of the numerical processing based on the detection of divergence from the values of the residuals. Most CFD codes have no test for divergence (Figure 3 overleaf).

Categorisation:-

Rule set: Monitoring
Application: Generic
Related to: Residuals

Heuristic rule:-

Rule: If current iteration residual is greater than the last iteration residual and the current iteration is greater than a specified minimum iterations then conclude that the solution is diverging.

Action: Under-relax or refine the mesh or stop.

Reason: Divergence means that a reliable solution will not be obtained.

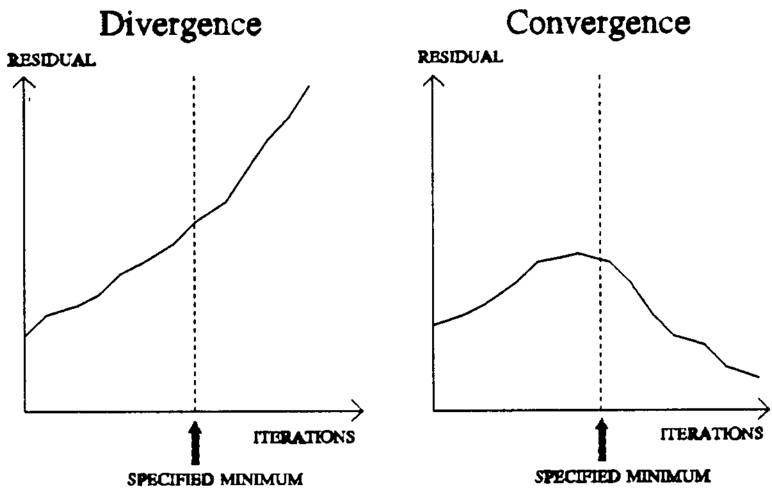

Figure 3

The specified minimum number of iterations before applying this rule allows the solution scheme to settle down. The detection of divergence means that processing should not be continued without taking some appropriate action as no valid solution can be obtained if the solution is diverging.

The following is an application specific rule would be for the detection of a plume in a fire simulation. These are common features of fire and smoke simulations that need specialised numerical treatment. There would be a similar rule for detecting horizontal jets caused by plume obstruction by blockages within the region (Figure 4 overleaf).

Categorisation:-

Rule set: Pattern_recognition

Application: Fire_simulations

Related to: Velocity values

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

Heuristic rule:-

Rule:	If many vertical velocities in adjacent cells then conclude there is a plume in that region.
Action:	Under-relax and refine mesh near the plume.
Reason:	Fires often have buoyancy driven plumes above the heat source that could cause divergence.

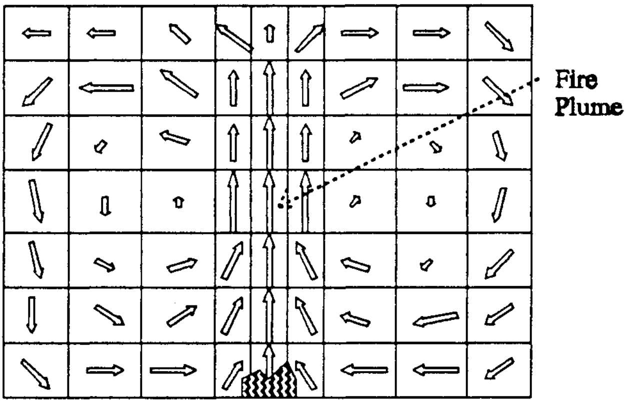

Figure 4

The solution stability near a fire plume (Figure 4) is doubtful if the mesh is too coarse or the relaxation parameter is too large. Under-relaxation and possibly grid refinement should be used near the plume. Fire plumes can often be predicted from the initial boundary conditions but run-time identification, from the velocity components, will be more accurate.

The numerical component has caused problems for the current research because of its inherent complexity. The use of an existing CFD code would seem appropriate, but most are difficult to use within a KBS system because of enclosed and inflexible data architectures and because most are batch mode codes with little or no external interactive control. There are also grave problems because most existing codes were developed in an evolutionary way with minimal documentation and limited use of software engineering techniques. Using a CFD code of this type is not feasible for the current research. The solution adopted is to re-engineer an "in-house" CFD code for use within the proposed KBS architecture developed for FLOWES.

There are still problems to be overcome in terms of re-implementation, integration, debugging and testing. However the system being developed is more likely to meet the aims and objectives of this research than a hybrid system that uses inflexible FORTRAN code directly.

The limited use of software engineering techniques within CFD research is a continued problem that is being addressed by the strict use of software design techniques[10],[13]. This helps to ensure that the design criteria are met and that the system provides sufficient flexibility for use in future research.

DEVELOPMENT ARCHITECTURE

The design of the prototype KBS is primarily for ease of implementation and flexibility rather than for speed performance. The system architecture currently under development is as follows.

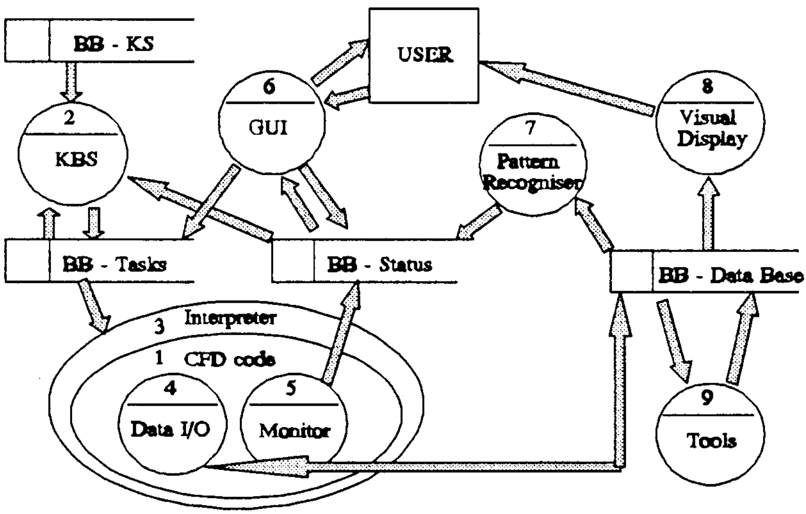

Figure 5

This design supports an external numerical database for the modular addition of processing and pattern recognition tools as and when they become available. The fact that many of the components are linked only by the blackboard database architecture allows the system to exploit cooperative network (or task) parallelism.

The blackboard architecture is designed to support cooperating tasks, so any component of the system can access the blackboard to determine if there are tasks for it to perform. This architecture gives an elegant solution to overall system control and is very flexible to alternate implementation strategies. The component modules are data linked only and do not have to be implemented in the same language or even run on the same processor. The external database causes significant overheads to update and maintain but, for the prototype system, gives the fastest development since existing tools can be used with minimal modifications (e.g. visualisation tools, grid generation tools and mesh refinement tools).

The Graphical User Interface will provide direct access to blackboard control and status information. This means the user can interactively control the CFD component with or without KBS support as required. This feature will be used for the knowledge acquisition.

The pattern recognition module(s) will initially be implemented as a visual pattern detection tool that presents selected data for the user to match with a visualisation of the intermediate results. This provides a simple method for pattern recognition without the overheads of research into reliable feature detection by computational search strategies.

PHASED DEVELOPMENT

The development of a prototype system to demonstrate the aims and objectives of this research is proceeding as follows:-

Stage (1) :-

- Implementation of a controllable CFD component.
- Development of an intuitive and flexible GUI.
- Definition of the control and pattern recognition message structures for use by the KBS and CFD components.
- Implementation of the external database and blackboard architectures to provide for extensible development.

Stage (2) :-

- Use of the prototype system from stage(1) to effect knowledge acquisition from application specific CFD experts.

Stage (3) :-

- Integration of the knowledge from (2) into a working system.
- Testing of the KBS system for consistency and efficiency.
- Extension of the system to other application areas.

Transactions on Information and Communications Technologies vol 1, © 1993 WIT Press, www.witpress.com, ISSN 1743-3517

CONCLUSIONS

The prior research and development of FLOWES indicates the potential benefits of interactive dynamic solution control by a KBS. Whilst FLOWES is currently limited in capability the extension of the research to complex flow simulations that are encountered in fire simulation and external vehicle aerodynamics will demonstrate the applicability of KBS techniques for CFD.

The code implementation techniques employed by researchers need to be reviewed in light of current trends in software engineering methodologies. That CFD codes are still being implemented with minimal design and with little or no quality assurance is appalling, particularly in light of the usage of CFD techniques for safety critical design applications (e.g. reactor design or jet-engine design).

The current research provides a new direction for CFD research based on reliability and overall efficiency rather than on merely speed performance of the numerical simulation phase. The flexibility of the proposed architecture is intended to allow easy maintenance and modification and not to preclude any future research.

The usage and extension of knowledge sources will eventually provide knowledge repositories that can support CFD research into the foreseeable future. The fact that knowledge will not be hard-coded within source code will be of particular benefit as new techniques and computational hardware are used.

This research is, to the best of our knowledge, unique in aiming to provide dynamic inference driven solution control of a CFD application using an easily modified database of heuristic rules.

REFERENCES

[1] D. Corkill "Blackboard systems" in 'AI Expert', Vol 6, No 9, pp 40-47, Sept 1991.

[2] B. Knight, M. Petridis "FLOWES: an intelligent computational fluid dynamics system" in 'Engineering Applications of Artificial Intelligence.' Ed. M.G.Rodd, Vol 5, No. 1, pp 51-58. Pergamon, 1992.

[3] D. B. Spalding "The Expert-System CFD code; problems and partial solutions" Conference Paper at International High-Tech Forum, Basel, May 1992.

[4] K. Jambunathan, E. Lai, S. Hartle and L. Button "Development of an Intelligent Front End for a CFD Package" in 'Artificial Intelligence in Engineering', Vol 6, No 1, pp 27-35, 1991.

[5] D. B. Spalding, H. I. Rosten "The PHOENICS Reference manual", CHAM Reference TR/200, Version 1.4, 1987.

[6] D. Finn, N. Hurley, N. Sagawa "AI-DEQSOL: A Knowledge Based environment for numerical simulation of engineering problems described by partial differential equations" in 'ESM: Modelling and Simulation', Copenhagen, Denmark, June 1991.

[7] B. Knight, M. Petridis, D. Cowell "Integrating an expert system with computational fluid dynamics software." in 'Expersys-92', Ed. A Niku-Lari, pp 153-159, 1992.

[8] M. Petridis, B. Knight, D. Edwards "A blackboard architecture for CFD software", Proceedings 13th IMACS World Congress on Computation and Applied Mathematics. Vol 2, pp 596-598, Dublin, July 1991.

[9] S. V. Patankar "Numerical Heat Transfer and Fluid Flow", Book, Publisher Hemisphere, British Library 536.2072, 1980.

[10] M. Petridis, B. Knight, D. Edwards "A Design for Reliable CFD Software" in 'Reliability and Robustness of Engineering Software II' Editors Brebbia, C.A., Ferrante, A.J., pp 3-18, Elsevier, 1991.

[11] C. Hirsch "Numerical Computation of Internal and External Flows", Book, Publisher Wiley, British Library 532.051'0724, 1989 (First 1988).

[12] C. Prakash "An improved control volume finite-element method for heat and mass transfer, and for fluid flow using equal-order velocity-pressure interpolation" in 'Numerical Heat Transfer', Vol 9, pp 253-276, 1986.

[13] R. S. Pressman "Software Engineering, A practitioners approach", Book, Publisher McGraw-Hill, British Library 005.1, 1987 (First 1982).