

SMARTFIRE: an Integrated Computational Fluid Dynamics Code and Expert System for Fire Field Modelling

S. TAYLOR, M. PETRIDIS, B. KNIGHT and J. EWER

Artificial Intelligence Group

University of Greenwich

London SE18 6PF, UK

E.R. GALEA and M. PATEL

Fire Safety Engineering Group

University of Greenwich

London SE18 6PF, UK

ABSTRACT

SMARTFIRE, an open architecture integrated CFD code and knowledge based system attempts to make fire field modelling accessible to non-experts in Computational Fluid Dynamics (CFD) such as fire fighters, architects and fire safety engineers. This is achieved by embedding expert knowledge into CFD software. This enables the 'black-art' associated with the CFD analysis such as selection of solvers, relaxation parameters, convergence criteria, time steps, grid and boundary condition specification to be guided by expert advice from the software. The user is however given the option of overriding these decisions, thus retaining ultimate control. SMARTFIRE also makes use of recent developments in CFD technology such as unstructured meshes and group solvers in order to make the CFD analysis more efficient. This paper describes the incorporation within SMARTFIRE of the expert fire modelling knowledge required for automatic problem setup and mesh generation as well as the concept and use of group solvers for automatic and manual dynamic control of the CFD code.

KEYWORDS: fire modelling, field modelling, CFD, expert system, artificial intelligence

INTRODUCTION

Over the past few years, the practise of fire field modelling [1] has begun the transition from the confines of the research laboratory to the desk of the fire safety engineer. To a certain extent, this move has been driven by the demands of performance based building codes. Unfortunately this transition has not been matched by significant improvements in the design of fire field modelling software or in the nature of its operation.

For the most part, software developers have relied on the computer hardware revolution - i.e. the development of inexpensive powerful desktop computers - to encourage this migration from research laboratory to engineer's desk top. Fire field modelling software still maintains the 'batch mode' operation philosophy associated with mainframe computers - i.e. long complicated manual problem set-up, long processing period, stop, analyse/correct, restart, long processing period, etc. Furthermore, in the transition from research laboratory to engineer's desk the nature of the end-user has changed dramatically. Whereas previously, researchers developing and using fire field models were experts in theoretical fire science and CFD, for the most part, todays users - fire safety engineers, architects, fire inspectorate, etc. - are possibly neither and certainly not experts in CFD. These issues have not been addressed by improvements in the software user interface or its mode of operation. While education of end users is a partial answer to this issue, it is not the complete answer.

These issues are addressed in this paper through the description of a new approach to fire field modelling software design which has been applied in the development of the SMARTFIRE [2,3] software. The paper concentrates on the use of artificial intelligence techniques to assist in the set-up of fire situations for CFD simulation and in the dynamic control of the CFD computations. Furthermore, the paper briefly describes a new concept in the solution of the algebraic equations associated with fire field modelling known as a group solver. Through the use of group solvers the computational efficiency of the fire field model is increased.

SMARTFIRE

The SMARTFIRE project has two primary aims, the first is to improve CFD computational efficiency thereby making SMARTFIRE an attractive tool for the fire modelling community. The second aim is to simplify the set-up and control of fire field models for non-expert CFD users. An associated sub aim of the project is to produce a teaching tool which the fire community can use to learn good CFD (and hence fire field modelling) practise.

In practice, SMARTFIRE will have two modes of operation: the novice and expert modes. These are intended for use by those with limited knowledge of CFD practice (i.e. novice mode); and experienced users who wish to utilise all features of the CFD code (i.e. expert mode). In expert mode, all CFD control is available to the experienced user, whereas in the novice mode, most of the control is handled by an expert system component.

SMARTFIRE has been developed by the University of Greenwich using a combination of in-house and proprietary software building blocks. SMARTFIRE will run on PC's under NT or Win32 and on UNIX based workstations running MOTIF. The minimum specification PC required to run SMARTFIRE is a 33 MHz 486 PC with 16 Mbytes of memory.

There are three major components to the SMARTFIRE program: interactive CFD code, User Interfaces, and Expert System. The SMARTFIRE CFD code has been described in a previous paper [2], and so only a brief outline will be presented here.

The CFD code, called CWNN++ is written in C++ and has been developed in-house from an existing FORTRAN code [4]. CWNN++ uses the validated numerical methods of the legacy FORTRAN code enhanced by object oriented developments in C++. CWNN++ uses three-dimensional unstructured meshes, enabling complex irregular geometries to be meshed without the need for body fitted co-ordinate grids. The code can solve coupled turbulent or laminar

flow problems under transient or steady state conditions. Unlike conventional CFD technology such as PHOENICS [5], FLOW3D [6], JASMINE [7] and SOPHIE [8], this allows extremely complex geometries to be efficiently meshed. The code also allows localised mesh refinement and the ability to remove cells from blocked regions - thus potentially saving tremendous amounts of otherwise wasted computer time.

The CFD engine of SMARTFIRE is continually being evaluated and tested. Predictions made using CWNN++ are compared with other commercial CFD codes and against data generated through physical experimentation. Test cases include standard CFD benchmark cases such as the backward facing step and moving lid problems and the Steckler room fire experiments [9]. Results produced by CWNN++ for the Steckler experiments [2, 10] compare favourably with those produced using commercial CFD codes such as CFDS-FLOW3D and PHOENICS [11].

The current version of SMARTFIRE is restricted to fires in single compartments. This is not a reflection of the limitations of CWNN++ but of the expert system knowledge base currently implemented. Furthermore, at this stage of the project development, fires are represented by a transient volumetric heat and mass source. Planned extensions of the software will include standard combustion models. The remainder of this paper concentrates on the expert system used to assist in problem set-up, the dynamic control of the CFD process and the use of the group solver. Information concerning SMARTFIRE including a demonstration may be found on the World Wide Web [12].

EXPERT SYSTEM

The expert system's prime tasks are to assist the novice user in problem specification/set-up and in dynamic control of the CFD engine. During problem specification the scenario is set up and meshed ready for input to the CFD engine. Dynamic control involves the evaluation of the solution process. Here the solution is monitored by the expert system to determine if any dynamic improvements can be made. The expertise incorporated within SMARTFIRE is elicited from several fire modelling and CFD experts from the University of Greenwich. The process by which this knowledge is acquired and the means by which it is incorporated within SMARTFIRE are described next.

Knowledge Acquisition.

The methods for knowledge acquisition have included interviews with an expert and example problems for the expert to work on. The interviews are concerned with the 'why' of the expertise in order to elicit expert 'rules'. A rule is an *if-then* condition-action clause which can activate at any time if the conditions are met. Rules attempt to model the expert's heuristic rules of thumb, commonly called 'shallow knowledge'. An example of such a rule could be : *if* there is a free vent on a particular wall - such as a door or window - *then* that wall needs an extended region to cater for the bi-directional flow which usually develops. Other examples of rules concern maintaining cell aspect ratios to within a predetermined limit and refining the computational mesh in the vicinity of walls.

An example problem consists of a scenario which the expert sets-up on paper, specifying the complete case, including boundary conditions, mesh, etc. Many of these were given to the experts and the results were coded and stored in a *case library*. This forms the basis of a

reasoning paradigm called *Case Based Reasoning* (CBR), which is a subset of expert systems technology. CBR is based on the premise that human experts rely on experience for their expertise. This experience is usually case based, relying on problems they have previously seen and solved. When faced with a new problem, they will recall a past case which is similar to the current problem. They will then adapt the past problem to take account of the new problem's differences, thus constructing a new case, which is then stored. CBR models this approach by having a *library* of past cases, a *retrieval* algorithm to find the closest case to the current problem, *adaption* rules to adapt the library case to the nuances of the problem, and a *storage* mechanism to store the new case. Thus CBR is a learning system, as it has the ability to store new cases.

SMARTFIRE's case library is based on room fire problems and is organised as a series of *room cases* and *slices* within each room case. A room case contains qualitative information describing the complete case, such as: the fire's location relative to walls, obstacles, vents etc., the vent's location on a particular wall, and the fire's physical size compared to, for example, the vent's size or the floor area. A slice is a cut across the geometry in a particular direction with information about its location, size and contents. It is over slices that the meshing occurs, and the slices contain the meshing instructions for each room. Each slice in a case is identified by its contents, which may be singular (i.e. fire) or multiple (i.e. fire and vent), and what other slices it is between (i.e. this slice is between a fire and a vent, or a fire is between this slice and a vent). The concept of a slice being between other slices is important, as many slices in the same geometry are likely to have the same contents.

EXPERT ASSISTED PROBLEM SET-UP.

The objectives for assisted problem set-up are primarily related to producing reasonable CFD code input from simple geometric and scenario data. The overall aim is that the process is automated to such a degree that a novice user can simply specify the geometry and scenario using interactive drawing tools leaving the remainder of the set-up procedure to the expert system. Furthermore, by interrogating the decisions of the expert system, the novice user may learn good CFD techniques. As part of the geometry/scenario specification, the user also specifies the nature of the boundary conditions.

For example the wall material type can be selected from a library which contains material properties for common building materials such as brick, wood, steel, etc. The user can also select special conditions such as insulating (adiabatic) or isothermal (constant temperature) properties. The user also specifies the heat output history of the fire.

The assisted set-up procedure addresses two areas: grid generation and initial condition specification. Grid generation is the process of determining how many control volumes are required for the problem and how they are to be distributed across the geometry. Initial condition specification is concerned with setting up suitable starting conditions for elements such as solver type per variable, initial solver iterations, and initial variable values. Due to space limitations we will restrict our discussion to grid generation.

Grid Generation.

The automated grid generation component uses expert system technology to produce a reasonable grid based on the geometrical and scenario data input by the user. The user

specifies room dimensions, vents on walls, fire properties such as location, size, and heat output. The user may also specify the cell budget to be expended in meshing the problem. This can be specified either in terms of actual number of cells or in the amount of time the user is prepared to wait for the solution. If the later is specified, the required number of cells is approximated from a determination of the time required to solve for one cell on the user's computer. Once the geometry, scenario and cell budget has been specified by the user, the expert system begins operation.

First, the cell budget is checked. Expert rules are used to suggest an approximate minimum realistic cell budget from the user defined input. These expert rules are based on experience and take into consideration overall geometry dimensions, aspect ratios, wall refinement etc. If the user has specified a cell budget which is less than the system recommendation a warning is flagged to the user. The overall cell budget is then split up into the three orthogonal direction components X, Y, and Z, guided by the estimator.

Next, the geometry is checked. If two objects are too close to each other in a direction, this will be reflected in the mesh as a slice of very thin cells in that particular direction. Expert rules concerning aspect ratios across neighbouring cells dictate that the differential should not be excessive. If there are no vents in the geometry (for example, the scenario being that of a fire in a room with all its doors and windows closed), then a small pressure relief vent is put into the geometry to simulate the effects of leakage from gaps under doors, imperfectly sealed windows, etc. This ensures that pressures in the room does not build up to excessive proportions. Again, this models human expertise, as human experts would do this to simulate the reality of an imperfectly sealed room.

Qualitative data describing the geometry is then inferred from the geometry. This is done to enable the geometry to be matched with a meshed case in the case library. As described above, the case library is the result of case based knowledge acquisition, where the expert is given a set of example problems which he solves. The cases in the case library determine the number of cells per slice in the geometry.

The case library is searched and the closest match found. At the current time, the case library consists of cases which are based on a regular room with a single vertical vent and fire. The expert system can deal with arbitrarily sized rooms, multiple vents and fires through the use of adaptation rules which are capable of combining more than one stored case from the library.

Through case retrieval, the number of cells for each slice in each direction is determined while observing the cell budget in that direction. At this point all cells in all slices are distributed evenly. Rules are then activated which apply single power laws to near wall and extended regions with the aim of packing cells closer to the wall. The packing for the single power law function obeys the y^+ boundary layer distance. For slices not at the edges of the domain, other rules operate. These attempt to resolve extreme aspect ratios of cells across slice boundaries. The desired cell sizes are passed into a function that fits a spline polynomial to cells in a slice to distribute them across that slice. At the same time, the function attempts to keep the aspect ratios of cells inside the slice at a reasonable level. If a failure is reported (i.e. keeping the slice boundary aspect ratios inside a tolerance forces the inner slice aspects to be suboptimal), a cell is added or subtracted from the slice and the process is repeated. Currently the "reasonable" tolerance for slices across boundaries and inside slices is 3:1, though aspects of 1:1 and 2:1 are attempted first. Once the cells have been distributed and pass the tolerance for the aspect

ratios, the grid is complete, and instructions are passed on to the mesh generator to generate the physical mesh file.

Example of Expert Generated Mesh

As an example of the mesh generation capabilities of SMARTFIRE, we will consider a case which does not correspond to any cases in the knowledge library. This consists of a rectangular room (2:1 aspect ratio) with three doors situated on three walls (see figure 1).

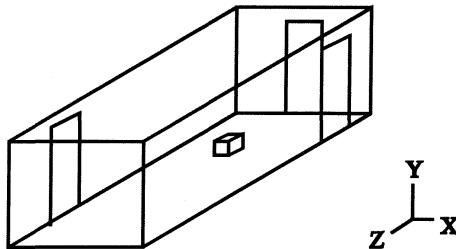


Figure 1: Room geometry consisting of three doors.

Two of the doors are not centrally located and the fire, which is situated on the floor, is not located centrally. The room dimensions are, 5.6 m in length, 2.8 m in width and 2.18 m in height. Each door has dimensions of 0.74 m in width and 1.83 m in height. The expert system suggests that for this case a cell budget of approximately 15,000 cells be used.

This geometry is significantly different from any cases in the library. For problems such as this, each geometry feature such as vents are considered in turn and matched to a library case individually. This provides a suitable starting point for the mesh generation process. The cases are then adapted to account for the differences between the current problem and the retrieved library cases. This is achieved through the use of the expert rules in the knowledge base. The grid is then distributed using spline power laws to resolve aspect ratios in adjacent cells.

The expert generated mesh is depicted in figure 2. The cells near the walls have been refined observing the y^+ boundary layer distance, and the cells in the vicinity of the fire are finer. Cells have also been graduated smoothly across the grid, refining towards the walls and the fire. The expert system has placed an extended region by each of the three vents. It is interesting to note that the extended regions associated with the vents on the long walls are differently meshed. This is due to one of the vents being significantly closer to the fire. As a result, the flow through that vent is expected to be significantly influenced by the fire hence requiring a more refined mesh.

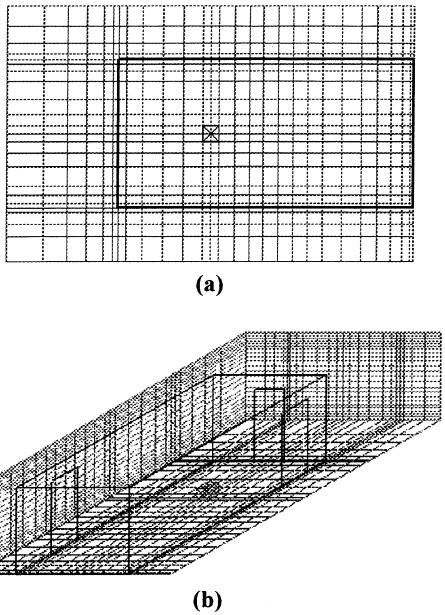


Figure 2: Floor plan (a) and perspective view (b) of expert generated grid.

GROUP SOLVERS AND DYNAMIC CONTROL.

Group solvers are a novel feature of the CFD component of SMARTFIRE. In traditional CFD codes, solver type and control apply over all the cells in the solution domain. Using group solvers, the solution domain is split into a collection of groups. A group is defined as a collection of cells that has its own control parameters independent of other cells in the solution domain. A group solver is used for a particular variable on a particular sub-region of the domain. The group solver makes use of standard numerical solution methods such as JOR or SOR. Still undergoing development and testing, the group solver is intended to speed up the solution process.

There are several different criteria to determine cell groupings. For example, cells may initially be grouped by location i.e. a near wall group or a fire group. Cells may be dynamically reassigned to other groups during the solution process. This process is triggered by selection criteria which are dependent on the magnitude of selected variables. For each group, there is a lower and upper value of the variable in question which defines an acceptance band for membership of that group. When the values in a cell comply with a particular group's entry criteria, the cell will be transferred to the appropriate group.

In the current version of the software four base groups are implemented: Active, Moderate, Inactive, and Void. Active has the upper band, moderate has the medium band and inactive has the lowest band. Void is used for areas in the geometry that are not part of the flow domain and there is a zero value of the variable (e.g. regions that have been meshed for convenience but are not part of the flow domain).

The main purpose of the group solvers is to reduce the overall computation time. This is achieved by directing computational effort to where it is most needed. One way in which this is achieved is by controlling the number of iterations the solver implements in the various groups. For instance, the maximum number of iterations performed in the Inactive group will be considerably smaller than the number performed in the Active group. As the solution develops, cells migrate to and from groups, thus receiving more or less computational attention. As the overall convergence criteria are set as for conventional problems, there should be no difference in the quality of the converged solution obtained using this technique.

As an example of the technique consider the extended region associated with a free vent such as a door. At the beginning of the simulation, the extended region consists of largely stagnant flow, and its cells will therefore be in the Inactive group. As the solution develops and entrainment begins to take effect the values in the extended region begin to increase. This causes more and more cells in the extended region to migrate from the Inactive group to one of higher priority and thus claim a larger share of computer time. This example also serves to demonstrate one way in which the expert system can exercise dynamic control of the solution process.

Still under development is another means by which the expert system will exercise dynamic control. This involves the optimisation of CFD control parameters while the CFD engine is running. The method of control is via the group solvers. The expert system receives information from a group and issues instructions back to the group, with all cells in the group responding to the instructions. Control of the groups concern relaxation parameters, solver iterations, time stepping and even solver type. It is likely that patterns such as "this group is converging slower than the others" will be passed onto the expert system to act upon. This type of automatic control is hoped to save considerable amounts of computer time while achieving converged solutions.

Dynamic access of this type is also available to the human expert user. Complete graphical access to the entire solution field is also available through the graphical user interface. Thus the expert human user may manually perform these types of interventions without the need to periodically stop, analyse and restart the solution process. This approach therefore allows the user to save considerable amounts of time associated with stopping and restarting the solution process. Furthermore, expert human users will potentially be able to catch convergence problems very early in the solution process - thus saving time - and immediately see the outcome of their interventions.

Examples of the Group Solver in Operation.

Here we will consider two two-dimensional fire simulations which utilise the group solver. The simulations were performed using a 90 MHz pentium PC. The first example involves a single compartment with two doors. Both doors open to the outside and hence involve two small extended flow regions. The second example involves a similar compartment in which one door opens to the outside while the other door opens to a second compartment. For simplicity, all confining boundaries are assumed to be adiabatic. In both cases a small volumetric fire of 50 kW is situated in the centre of the fire compartment, a situation similar to one of the Steckler room scenarios [9,11]. The fire compartment measures 3.0m in length by 2.18m in height. The door has a height of 1.83m.

In the first example, one of the doors is open throughout the simulation while the second door is opened 40 seconds into the fire simulation. The solution domain is thus made up of three distinct regions, the first external region by the open door, the fire compartment and the second external region by the closed door. The length and height of each subdomain is 3m x 2.18m. The computational mesh in each region comprises of 8 x 21 cells, 22 x 21 cells and 8 x 21 cells respectively i.e. a total of 798 cells. Using standard CFD solution techniques the equation solver operates equally in all the cells throughout the solution domain, even the cells in the dormant external region by the closed door. This is clearly a waste of CPU time as nothing of significance occurs in the region closed off by the door.

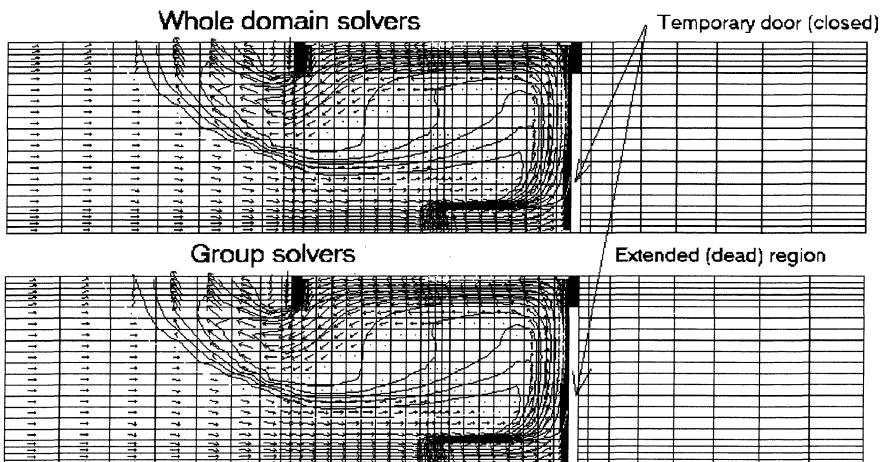


Figure 3: Solution prior to opening of second door obtained using conventional and group solvers.

Using the group solver, this region is marked as Inactive resulting in the solver spending a minimum amount of effort in this region. When the second door opens after 40 seconds, the Inactive region changes to Active and the solution domain extends to cover the second extended region. As demonstrated in figure 3, the solution just prior to the second door opening when the group solver is used is identical to the solution when the conventional solver is used, i.e. there is no loss of accuracy.

When both doors are opened, both methods converge to the steady state solution depicted in figure 4. However, using the conventional solver, the run time up to the point where the second door opens was approximately 3.02 hours while using the group solver this was reduced to 2.72 hours, a saving of 10%. While only a modest saving, this was achieved by saving the computational effort over only a small proportion of the solution domain. When Inactive regions occupy a greater proportion of the mesh savings in computational time can be improved significantly.

Central fire with both doors open

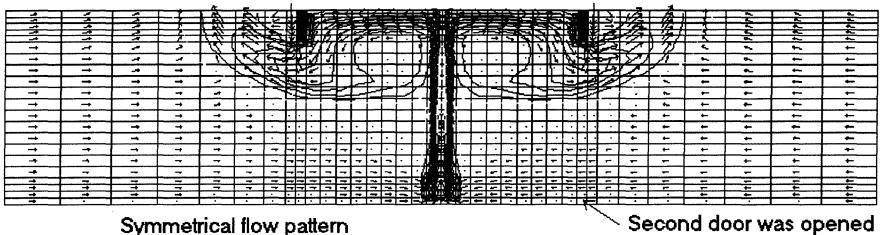


Figure 4: Steady-state solution obtained after both doors are opened (example 1).

In the second example, the second door is opened after 40 seconds but rather than opening to the outside it opens into an otherwise sealed compartment. The solution domain again consists of three distinct regions, the external region by the open door, the fire compartment and the second sealed compartment. The length and height of each subdomain is 3m x 2.18m, 3m x 2.18m and 4m x 2.18m. The computational mesh in each region comprises of 8 x 22 cells, 22 x 22 cells and 30 x 22 cells respectively i.e. a total of 1320 cells. Using standard CFD solution techniques the solver operates equally in all the cells throughout the solution domain, even the cells in the second sealed and isolated compartment.

Fire in a partitioned room

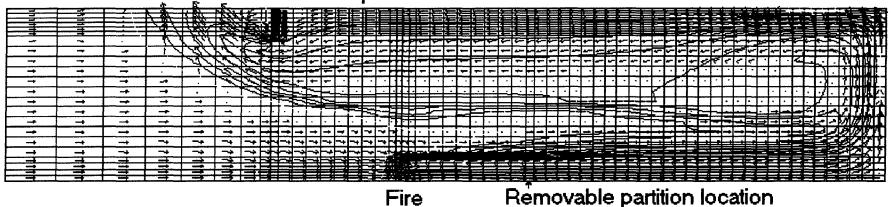


Figure 5: Steady-state solution obtained after both doors are opened (example 2).

Once again, the dormant region is marked as Inactive resulting in the numerical solver spending the minimum amount of effort in this region. When the second door opens after 40 seconds, the Inactive region changes to Active and the solution domain extends to cover the second compartment. As in the previous case both solution techniques result in identical solutions prior to the opening of the second door. When both doors are opened, both methods converge to the steady state solution depicted in figure 5. However, using the conventional solver, the run time up to the point where the second door opens was approximately 4.40 hours while using the group solver this was reduced to 3.06 hours, a saving of 31%. Thus, by effectively reducing the computational domain by 50%, a saving in computational time of 31% is achieved. While this represents a considerable saving in resources, further gains are possible through planned enhancements to the group solver. Furthermore, while these examples have

been restricted to two-dimensions, the group solver is able to operate over three-dimensional flow domains.

Finally, SMARTFIRE has been introduced to several potential users. While these people have experience of fire safety and have superficial knowledge of fire field modelling they have never used a fire field model. After a 30 minute introduction to SMARTFIRE, they were able to construct a room fire scenario - including mesh - within 15 minutes and subsequently perform a successful fire simulation.

CONCLUSIONS.

SMARTFIRE is an open architecture integrated CFD code and knowledge based system which attempts to make fire field modelling accessible to those who are not experts in CFD such as fire fighters, architects and fire safety engineers. This is achieved by embedding expert knowledge into CFD software. SMARTFIRE also makes use of recent developments in CFD technology such as unstructured meshes and group solvers in order to make the CFD analysis more efficient.

The current user interface and knowledge base extends to scenarios involving a single room configuration, including multiple vents and fire sources. Within these present constraints, the interface and knowledge base allows non experts to specify and run a problem within minutes. This includes setting the geometry, boundary conditions, computational mesh, initial conditions and solver parameters. Using conventional CFD codes, non-expert users may require many hours to set up a similar problem. They also suffer the risk of generating a poor mesh or incorrectly setting the other parameters. The SMARTFIRE set up procedure is such that the generated mesh is of a good quality, obeying the mesh generation rules that expert CFD practitioners generally follow.

Once the scenario has correctly been set up, the CFD code can achieve a converged solution within hours. Run times are dependent on the size of the computational problem and the computational power of the computer. As SMARTFIRE runs on PC and Workstation platforms, run times may vary considerably. However, through the use of group solvers, considerable savings in runtimes can also be achieved.

Further developments in the SMARTFIRE software will include extending the capabilities of the knowledge base and user-interface to multiple room geometries, extending the knowledge base to include dynamic control of the solution process and implementation of standard combustion models. Associated with these developments are the on-going requirements of field testing and validation. SMARTFIRE is currently supported by the University of Greenwich and the UK Home Office. Those interested in acquiring a copy should contact the authors.

SMARTFIRE attempts to bring the complex world of fire field modelling a little closer to the desk of those in the fire community who are not experts in CFD. Due to the growing acceptance of performance based building codes by legislators, code enforcement agencies and fire safety engineers, developments such as SMARTFIRE are desirable, essential and indeed inevitable.

ACKNOWLEDGEMENTS

Professor E.R.Galea is indebted to the UK CAA for their financial support of his personal chair in Mathematical Modelling at the University of Greenwich. The authors gratefully acknowledge the financial support of the Home Office Fire Research and Development Group and the University of Greenwich.

REFERENCES

1. Galea E.R, On the field modelling approach to the simulation of enclosure fires, Journal of Fire Protection Engineering, 1 (1), 1989, pp 11-22.
2. Taylor S, Galea E.R, Patel M, Petridis M, Knight B and Ewer, J, SMARTFIRE: An Intelligent Fire Field Model, Proc Interflam 96, Cambridge, UK, March 1996, pp 671-680.
3. Taylor S, Galea E.R, Patel M, Petridis M, Knight B and Ewer, J, SMARTFIRE: An Integrated CFD environment using knowledge based reasoning for fire simulation modelling. Conference Abstracts, First European Symp on Fire Safety Science, Zurich August 1995.
4. Ewer J, Knight B and Cowell D, Case Study: An Incremental Approach to Re-engineering a Legacy FORTRAN Computational Fluid Dynamics Code in C++, Advances in Engineering Software, vol 22, 1995, pp 153-168.
5. Spalding D.B, A General Purpose computer Program For Multi-Dimensional One- and Two- Phase Flow, Mathematics and Computers in Simulations, North Holland (IMACS), Vol XXIII, 1981, 267.
6. FLOW3D Release 2.3.3 Reference Guide, CFD Dept AEA Harwell UK, Feb 1991.
7. Kumar S, Gupta, A.K and Cox G, Effects of Thermal Radiation on the Fluid Dynamics of Compartment Fires, Fire Safety Science - Proc of the Third Intl Symp, 1991, pp 345-354.
8. Bjorkman J, Keski-Rahkonen and Lewis M.J, First simulations of the Steckler Room Fire Experiment by Using SOPHIE, Conference Abstracts, First European Symp on Fire Safety Science, Zurich August 1995.
9. Steckler K.D, Quintiere J.G and Rinkinen W.J, Flow Induced By Fire in a Compartment, NBSIR 82-2520, National Bureau of Standards, Washington, September 1982.
10. Petridis M, Galea E.R and Knight B, Report to the Home Office: SMARTFIRE Project: Report to Accompany the First Set of Deliverables, University of Greenwich, Feb 1995.
11. Kerrison L, Mawhinney N, Galea E.R, Hoffmann N and Patel M, A Comparison of Two Fire Field Models With Experimental Room Fire Data, Fire Safety Science - Proc of the Fourth Intl Symp, Ottawa, Canada, 13-17 July 1994, pp 161-172.
12. www: <http://www.gre.ac.uk/research/cms/fire>