

A Knowledge Based Approach to Mesh Optimization in CFD

Domain :

1D Euler Code Example

Tharini Santhanam, J.C. Browne, J. Kallinderis and D. Miranker

Department of Computer Science

The University of Texas at Austin

Austin, TX 78712

Abstract

This paper gives the first step in a systematic program of application of knowledge-based system methods for optimization of the computation process for adaptive fluid dynamics computations. The results presented are a systematic exploration of the parameter space for optimization of the mesh for solution of an explicit time-marching finite volume 1D Euler code. The substantive benefit possible from intelligent adaptation is determined. The results are put in the context of a more ambitious project.

lection of these parameters can in principle yield an exponential reduction in the computational cost to attain a given level of accuracy of the computational cost over use of a single fixed set of parameters. Selection of "optimal" parameter values is, however, still largely based on heuristic methods. Therefore parameter selection for the computational processes for fluid dynamics is a promising domain for application of knowledge-based systems (KBS). This paper gives the results from the first small step in an ambitious program for application of KBS methods to optimization of adaptive computational fluid dynamics (CFD) programs. Space precludes review of related research. A survey can be found in [4].

1 Introduction

Computation of a solution to a system of partial differential equations defining a fluid dynamics problem can be viewed as a process whose execution behavior can be optimized over a complex set of parameters. It has been shown [3] that dynamically optimal se-

2 Approach

Full scale multi-dimensional production codes contain a very large number of parameters. It seemed prudent to begin this study with a thorough and systematic study of a simple case in order to establish

an infrastructure for study of the more complex systems, to evaluate the practicality of the approach and to determine how much of the potential impact of optimization can be readily realized with a straightforward approach.

The simple case of an explicit time-marching finite volume 1D Euler code which restricts adaptation to only the mesh size was chosen as the first system for study.

The system of the one-dimensional Euler equations of inviscid fluid flow may be written in cartesian coordinates conservation form as :

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \quad (1)$$

where

$$U = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, F = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ (E + p)u \end{pmatrix}$$

are the state and flux vectors respectively. In the above relations ρ is density, u is velocity, E is total internal energy per unit volume, and p is pressure. A one-step Lax-Wendroff-type integration scheme [1, 2] has been employed.

A typical use of this code might be to study the propagation of perturbations in a uniform field.

Grid adaptation consists of adjusting the grid spacing so that the numerical error is relatively small and equally distributed throughout the solution domain. This is accomplished by increasing grid resolution locally in regions in which flow features exist (i.e. regions where changes in flow-field quantities are large,

for example shocks and stagnation regions). Initial coarse grid-cells are divided by inserting additional points inbetween the initial points, thus creating a local embedded grid. Conversely, excessive resolution is removed by deleting grid-points locally over regions in which the solution does not vary appreciably.

This code was found to have five parameters which may exercise significant impact on computational cost. Formulation of the knowledge base for the 1D code was done primarily by interviews with the authors of the code, Y. Kallinderis (one of the authors of this paper) and A. Vidwans. The five parameters identified were: (i) the solution property to be used to invoke mesh adaptation (the detection parameter), (ii) the threshold for the detection parameter for initiating cell division, (iii) the threshold for the detection parameter for initiating cell coalescence, (iv) the interval between evaluations of the detection parameter (frequency of adaptation) and (v) the initial mesh. Another possible parameter is the factor by which mesh size is divided or multiplied when a cell division or coalescence is carried out. This was set to two for these studies. Yet another possible parameter is the size of the time step. The time step was not varied but instead was set to 1 for these experiments.

Parameter selection and binding can be done at several times. Certain parameters can be determined at the time of design of the code from the characteristics of the physical system being modeled. This is the case for the detection parameter. (The detection parameter will normally be an estimator for the error in the computed solution.) The flow is supersonic, inviscid and compressible. This set of conditions leads to the selection of the velocity difference between mesh points as the detection parameter. Certain parameters should be reasonably constant across the problem

set to which the code applies. These can be selected by executing the code for a span of parameter values and choosing a single "near optimal" value. The initial mesh and the threshold coefficients of the detection values for cell division and coalescence are examples. There are two modes of selecting a near optimal mesh. The code can be executed with an initial mesh with the detection parameter evaluated at each mesh point. The resulting values for the detection parameter can then be used to construct a near optimal mesh for a final evaluation. The other approach is to dynamically adapt the mesh as the code is executing using the values of the frequency and thresholds to minimize the total execution cost of the computation. The latter approach is used in this study.

The knowledge-based system is formulated in terms of production rules and encoded in the 'C' Language Integrated Production System (CLIPS) [5]. CLIPS was chosen as the programming vehicle because it compiles to C and thus both gives reasonable performance and is readily interfaced to Fortran which is the dominant language for CFD computations.

3 Results

The direct results reported are the optimal values for the four parameters, an assessment of the infrastructure for more extensive studies such as the feasibility of rule embedding in Fortran programs.

Figures 1,2,3 and 4 are examples of parameter selection experiments. The metric of goodness for selection is the number of iterations required to converge the solution to a target accuracy. Figure 1 shows the

effect of the frequency of adaptation on the number of iterations to convergence. There is a minimum at an interval of seventy steps. (It is interesting to note that the initial selection by the "experts" for the code was 50 steps.) Figures 2 and 3 show the effect of the thresholds for cell division and cell coalescence on the number of iterations to convergence. There are clear minima at 1.4 and 0.9. (The actual thresholds are the products of these constants with the average value of the velocity difference between mesh points.) Figure 4 is a plot of number of iterations before convergence as a function of the number on nodes in the initial mesh. The interpretation of this graph requires the additional information of what is the coarsest mesh which does not result in the loss of flow features in the resulting solution. An initial mesh of at least 31 points is required to insure capture of flow features. Figures 5,6,7 and 8 show the solution and the mesh (plotted as a series of straight lines at the bottom of the figures) at several successive points in time for the propagation of an initial triangle-shaped perturbation in a uniform one-dimensional field. Note how the mesh is successively refined and coarsened as the solution features evolves across the spacial coordinate.

A significant portion of the research so far has been to set up the framework for study of more complex systems. The actual programming of the knowledge base in the work to date could easily have been done in Fortran or C. The exercise of working in a formal rule-based language system has demonstrated the feasibility of working on more complex systems.

4 Comment and Future Work

Another goal is to determine the transferability of knowledge across problems. There will be a further series of experiments on the 1D code to define the robustness of the knowledge base across the problem set reachable with this code. We will report at the the Workshop on the span of applicability of the knowledge gained in these studies.

The next step in this project after completing the analyses for the 1D Euler code is to formulate and apply a knowledge-base for a more complex problem structure as implemented in a 2D finite element Navier-Stokes code which includes both mesh optimization and approximation function optimization.

The long term goals include development of a programming environment for CFD codes which will facilitate formulation and implementation of intelligent adaptive CFD codes.

5 Acknowledgements

This research has been supported by the IBM Corporation through grant number 61653 and by the National Science Foundation through grant number ASC-9111540.

References

- [1] Y. Kallinderis, and J. R. Baron "Adaptation Methods for a New Navier-Stokes Algorithm," Journal of the American Institute of Aeronautics and Astronautics, vol. 27 , no. 1 , pp 37-43 ,1989.
- [2] Y. Kallinderis, "Adaptation Methods for Viscous Flows," Ph.D. Thesis, MIT, Dept. of Aeronautics and Astronautics, CFDL-TR- 89-5, May 1989.
- [3] J.T. Oden - Private Communication.
- [4] Tharini Santhanam, J.C. Browne "A Knowledge Based System for CFD" Technical Report, Department of Computer Sciences, UT Austin, 1992.
- [5] CLIPS Version 5.1 Reference Manual, Software Technology Branch, NASA Johnson Space Center, October 1991.
- [1] Y. Kallinderis, and J. R. Baron "Adaptation Methods for a New Navier-Stokes Algorithm," Journal of the American Institute of Aeronautics and Astronautics, vol. 27 , no. 1 , pp 37-43 ,1989.

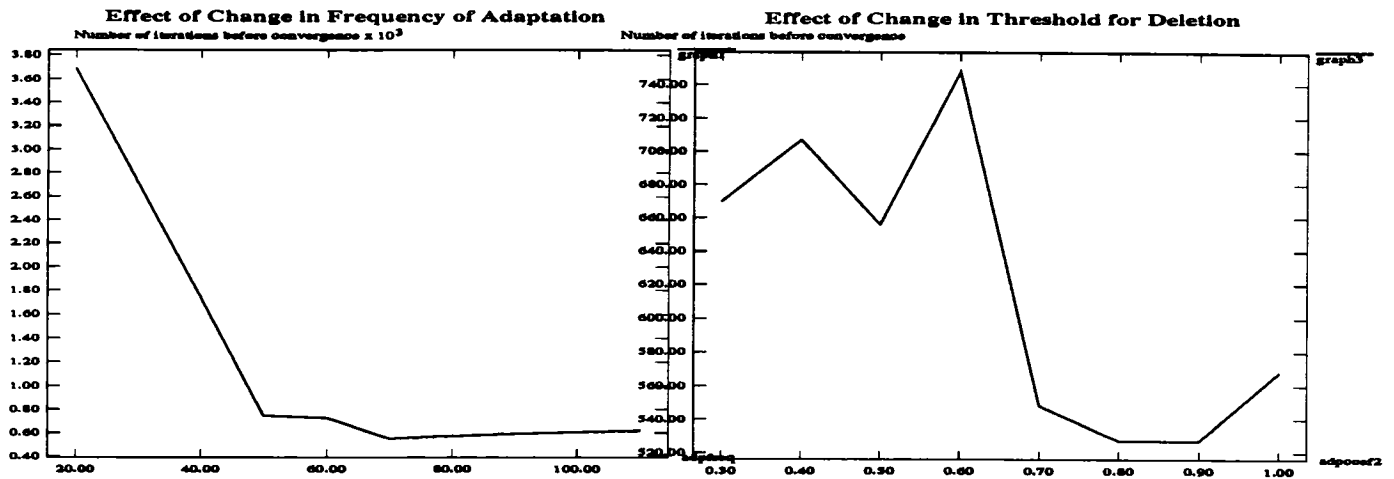


Figure 1: Number of iterations before convergence versus Frequency of Adaptation

Figure 3: Number of iterations before convergence versus Threshold for Cell Deletion

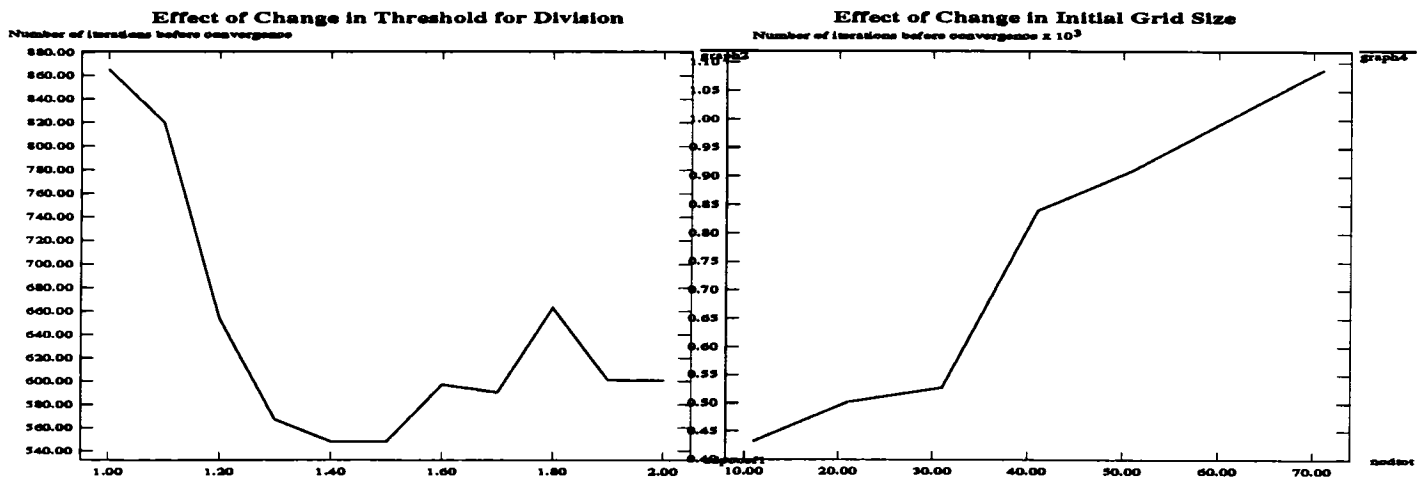


Figure 2: Number of iterations before convergence versus Threshold for Cell Division

Figure 4: Number of iterations before convergence versus Number of Initial nodes

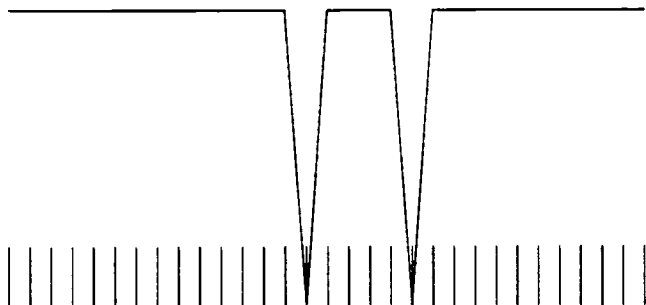


Figure 5: Propagation of solution perturbation and corresponding adapted grid



Figure 7: Propagation of solution perturbation and corresponding adapted grid

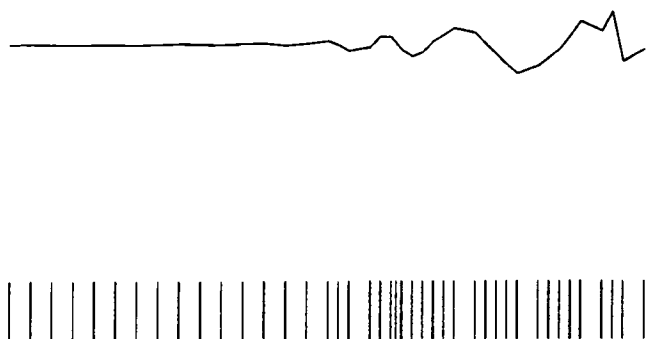


Figure 6: Propagation of solution perturbation and corresponding adapted grid

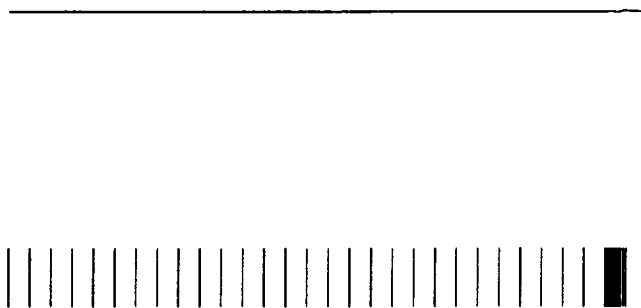


Figure 8: Propagation of solution perturbation and corresponding adapted grid