A ZONAL APPROACH FOR THE SOLUTION OF COUPLED EULER AND POTENTIAL SOLUTION. (U) INDIANA UNIV-PURDUE UNIV AT INDIANAPOLIS SCHOOL OF ENGINEERING. A CER 01 FEB 88 UNCLASSIFIED AFOSR-87-0015
Final Report
submitted to
Air Force Office
of Scientific Research
Bolling Air Force Base
Washington DC 20332

for
Instrumentation Request for
A Zonal Approach for the Solution of
Coupled Euler and Potential Solutions of Flows with Complex Geometries

for the period of
July 1, 1986 - June 1, 1987

by
Akin Ecer, Principal Investigator
Division of Engineering P.O. Box 647
Purdue University at Indianapolis
1201 E. 38th Street
Indianapolis, In 46223

DISTRIBUTION STATEMENT A
Approved for public release
Distribution Unlimited
Table of Contents

Summary of the Progress..................................................1
  1. Objectives.........................................................1
  2. Summary of the Development Effort..............................1
  3. Publications.....................................................2

Details of the Performed Work........................................4
  1. Objectives.........................................................4
  2. Solution Scheme..................................................4
  3. Computer System..................................................5

References........................................................................7
Summary of the Progress

1. Objectives

The main objective of the ongoing research program with AFOSR is to develop a block-structured solution scheme for Euler equations [1-5]. The methodology is aimed at providing computational techniques for solving three-dimensional, transonic flow problems around complex geometries. Also, we are interested in treating such large problems using large computers with vectorizing and parallel processing capabilities.

Our past work has been implemented on main frames such as IBM, Cray and CYBER computers. It was based on storing a large problem in terms of many smaller regions (blocks) on available disc storage. It was assumed that the main memory is not large enough to store the entire problem. In these applications, portions of this data was brought into a main memory which was shared by one or a few processors. Each processor operated on its share of data (a single block). The instrumentation was requested to implement the same scheme on a computer with distributed memory and many processors. In this application the problem is again divided into many blocks. However, each block is stored in the distributed memory of an individual processor and computed seperately. Only the boundary conditions between the neighboring blocks are communicated between the blocks.

2. Summary of the Development Effort

The hardware (Intel IPSC hypercube) was purchased and installed. It was linked to the other computers for data transfer. A version of our Euler solver was developed to run on a distributed memory machine. Up to date,
two basic test cases were computed [5], although the activity is continuing at a part of our AFOSR research program.

The flow field around a NACA0012 profile at $M_\infty = 0.80$ was solved by dividing the flow field into: 4-blocks with 4000 grid points, 8-blocks with 8000 grid points and 16-blocks with 16000 grid points. It was demonstrated that by using 4, 8 and 16 processors for each of the above problems one can obtain the same computation time.

The second example was to model a 16-block representation of the flow around a wing-body such that 16 processors are combined to each other similar to the way blocks are connected. This set-up provides an efficient way of representing a problem with a complex block structure with minimum data transfer. In this case the data transfer again occurs between the neighboring blocks.

The work is ongoing to demonstrate:

1. the need for increasing the number of processors for solving large problems as well as their speeds.
2. the need for using many loosely coupled computers for solving complex geometric problems so that all computer resources (memory, CPU, vectorizing capabilities disk I/O etc.) can be simultaneously utilized.

3. Publications

Two papers were presented describing this activity:


Also two presentations were made at two conferences. The written versions of these presentations are being prepared.


Details of the Performed Work

1. Objectives

The objectives of the ongoing research were:

i) To develop a finite element mesh generation scheme that will allow optimum grid configurations to be established for different flow zones [1],

ii) To develop a "zonal" solution scheme for three-dimensional transonic flows that will allow independent grids to be generated in each flow zone and produce an iterative solution, coupling the local solution of each block, [2, 3].

iii) To implement the "zonal" solution scheme on computers with parallel processing capabilities [4, 5].

This last task was completed first on IBM 3090, Cray XMP and CYBER 201 computers [4]. The present attempt was aimed at working with a computer with many processors with distributed memory and evaluate the requirements for computing large problems [5].

2. Solution Scheme

The solution procedure involves a block-structures relaxation scheme where two basic computational tasks exist:

- the formulation of residual vectors and
- the solution of a set of implicit equations.

In the developed scheme, the problem is to define the complex flow field in terms of a series of simpler blocks. The grids are generated for each of the blocks and stored separately on the memory of individual processors. The residuals are calculated for each block. If each processor
has sufficient memory much of the information for computing residuals and which is related to the geometry does not change over the iterations. Such information is stored rather than recalculated. The equations are then solved implicitly for each block separately. The data transfer involves only the exchange of boundary conditions between the neighboring blocks [5].

3. Computer System

IPSC (Intel Personal Super Computer) system involves hardware and software components necessary to work in a parallel processing environment. Our system is built on a cube where a basic cube consists of 16 processors (nodes). These nodes are connected to the neighboring nodes by 32 I/O channels as shown in Figure 1. The system can be expanded to 4 cubes or 128 processors. Each processor includes 4.5 m - bytes of memory, where the data transfer is at a rate of 10Mbit/second on the I/O channels. The cube manager itself is a 9Mbytes of memory.

Figure 1. Cube topology for the 16 node system.

Our solution scheme is very suitable to be executed using the above
system. The flow field can be divided up to 16 blocks and each block would be analyzed on a single processor. The data transfer between the neighboring blocks is in terms of boundary conditions (mass and entropy fluxes calculated through the iterations). The objective is to ensure that the conservation of mass and entropy is satisfied between the blocks. Since there is an individual data channel between the neighboring blocks there to provide such information, considerable improvement is obtained. In solving large problems, data transfer plays an important role in terms of efficiency. Compared to a single processor with a single I/O channel, the proposed system may provide 16 times speed improvement in terms of computational speed as well as 32 times speed improvement in terms of data flow. Also because of our computational scheme the data transfer between the neighboring blocks solve only a small data set (boundary conditions).

The availability of large real memory for each processor enables elimination of many computations which are represented through the iterations. The performance figures on this machine as related to main frame computers will be presented in coming papers.
References


A block structured solution scheme which was developed for the solution of three dimensional euler equations was implemented on the Intel Hypercube computer. By using this scheme, a complex flow problem is divided into a series of blocks. Each block is stored on an individual processor of the system and solved. The boundary conditions between the neighboring blocks are iteratively calculated. The communication of the boundary conditions between neighboring blocks occurs on individual I/O channels.
END DATE FILMED APRIL 1988 DTIC