

A Transputer based Parallel Algorithm for Surface Panel Analysis

Dr. S.R. Turnock
Department. of Ship Science
University of Southampton
Southampton. SO9 5NH U.K.

SUMMARY

A surface panel method has been developed to run in parallel across variable sized square arrays of transputers. A geometric parallelism is used for both the data distribution and the algorithm. A flexible geometry definition allows complex three-dimensional surfaces and multiple body problems to be solved. Each body surface is sub-divided into quadrilateral panels. A fast parallel block-iterative solver was developed which allows rapid solution of the dense but diagonally dominant linear system of equations. The parallel performance of the surface panel code is described and the necessary scaling of number of transputers and distributed memory per transputer to obtain solutions of surface panel problems of order of 10,000 panels is given. A final section gives, as an example, the use of the code in predicting ship rudder-propeller interaction.

1. INTRODUCTION

The use of surface panel methods for modelling potential flow around marine vessels is widespread. For the hydrodynamicist they provide a valuable tool capable of reasonable prediction of body forces without extravagant use of computational time. However, for larger problems involving detailed three-dimensional surfaces and multiple body problems the required computational time still restrict their use.

In general surface panel techniques are solved using implicit techniques which require the calculation of coefficients for a dense matrix and then the solution of a large linear system of equations. Parallel algorithms are easily produced for explicit schemes however there is a need for research into the ways in which the benefits of parallel processing can be applied to implicit algorithms such as the surface panel method. The work reported is part of a research programme to investigate ship rudder-propeller interaction and further details can be found in Molland (1992 a,b).

The implementation of a lifting surface panel method to run on an array of transputers using the developed communications harness is described. A suite of procedures for carrying out the various stages of the analysis has been written and is referred to as the PALISUPAN (PARallel LIFTing SURface PANEL) code. A geometric parallelism was used for the data distribution and the numerical formulation of PALISUPAN. The parallelism is based on equally dividing the total number of lifting surface and wake panels amongst the numbers of transputers available on a given parallel computer.

An important parameter in parallel processing is the measurement of the performance of a particular parallel algorithm on a given parallel computer. How this is quantified and

how performance is compared to that of an equivalent serial algorithm are necessary questions in determining whether transputer based parallel computers provide a cost-effective method for carrying out a particular application.

All the software was written in Occam2. The overall software design philosophy was to minimise the development time and subsequent debugging by the use of simple geometric algorithms. A structured approach making full use of the procedures and channel communications of Occam2 allowed this to be successfully carried out.

A variety of methods can be used to produce parallel algorithms to solve a lifting surface panel problem with a total of N panels using T transputers. A parallel geometric algorithm where each transputer is assigned (N/T) panels is the simplest method and is one which naturally lends itself to the solution of computational fluid dynamic problems. Also, problems with different total number of panels can be easily scaled without the need to alter the software simply assigning a different number of panels to each transputer.

2. TRANSPUTERS

The transputer is a micro-processor based integrated circuit designed as a basic building block for the construction of both large and small scale parallel computers. Associated with the transputer is Occam2: a computer language specifically developed to make full use of the parallel processing capabilities of the transputer.

Transputers are a range of high-performance VLSI (Very Large Scale Integrated) technology devices, developed by Inmos Ltd, which consist of local memory, four high speed two-way links and a micro-processor unit all mounted on a single silicon chip. The provision of high speed communication links allows transputers to be connected together to produce a parallel processing computer. There are no limits to the number of transputers which can be connected together in a network. The only restriction is in the topology of the parallel machine. Each transputer can be connected to a maximum of four. Massively parallel machines can be built up from large numbers of transputers.

Parallel computers are classed according to the number of tasks (or instructions) and number of data streams they can process simultaneously. Transputer based machines belong to the most general class of Multiple-Instruction-Multiple-Data stream (M.I.M.D.) machines. The advantage of transputer based parallel processing systems is that the same basic processing unit can be used for both small-scale and large-scale computational applications. Code can be developed on inexpensive machines with a small number of transputers and then executed on a large array of transputers.

The performance of transputer based parallel computers can be scaled if all the component transputers have identical computational and communication loads. This facility allows parallel computers which use small numbers of transputers to be used to assess the performance of large scale computations. An important proviso is that an appropriate scale of problem size is used. As an example of such a study Robinson (1990) investigated the parallelism of a commercial fluid dynamics software package ASTEC, Lonsdale (1989), which uses an implicit finite volume solution method on a finite element mesh. They concluded that transputer based parallel systems can deliver greatly increased performance and also that parallel systems allow problems to be solved which could not be tackled on

