UNIVERSITY OF SOUTHAMPTON

Prediction of ship rudder-propeller interaction using parallel computations and wind tunnel measurements

S.R.Turnock, MA, SM.

A thesis submitted for the degree of

Doctor of Philosophy

Department of Ship Science

Faculty of Engineering and Applied Science

April, 1993

UNIVERSITY OF SOUTHAMPTON

Abstract

FACULTY OF ENGINEERING AND APPLIED SCIENCE

SHIP SCIENCE

Doctor of Philosophy

PREDICTION OF SHIP RUDDER-PROPELLER INTERACTION USING PARALLEL COMPUTATIONS AND WIND TUNNEL MEASUREMENTS

by Stephen Richard Turnock

A theoretical method has been developed to predict the forces developed due to the interaction between a ship rudder and propeller. A parallel lifting surface panel program (PALISUPAN) has been written in Occam2 which is designed to run across variable sized square arrays of transputers. This program forms the basis of the theoretical method. The rudder and propeller are modelled separately. Their interaction is accounted for through an iterative process whereby their respective inflow velocity fields are modified using a circumferential average of the disturbance velocity due to the other body. Prior to writing PALISUPAN, software techniques for the implementation of computational fluid dynamic algorithms across arrays of transputers were developed. The approach used is based on a geometric parallelism. At the outermost level on each transputer the particular CFD algorithm runs in parallel with a harness process. The harness controls the communication across the transputer array. To prove this concept an explicit finite volume solver for the two-dimensional Euler equations has been implemented. PALISUPAN itself uses a perturbation potential formulation and an explicit zero pressure loading condition is enforced at the trailing edge. Use of the communications harness greatly reduces code development time and although an implicit solver PALISUPAN gives good parallel performance. Wind tunnel tests were undertaken to derive experimental data for the validation of the prediction method. These used a 3.5m x 2.5m low speed wind tunnel and a range of flow and geometrical parameters were tested. Total rudder forces and moments, propeller thrust and torque, and quasi-steady rudder surface pressures were measured. Empirical relationships for the prediction of rudder lift, drag and stall for use in ship manoeuvring studies were also derived. The validated theoretical prediction for rudder-propeller interaction using PALISUPAN allows the detailed design of ship rudder-propeller systems to be enhanced. The parallel performance of the PALISUPAN code demonstrates the practicality of using transputer arrays to solve CFD problems.

Or think of a ship:
big as it is and driven by such strong winds
it can be steered by a very small rudder,
and it goes wherever the pilot wants it to go.

Good News Bible,
James chapter 3, verse 4

Acknowledgements

I would like to thank the many people who have helped in the progress of this work over the last four years. The skill of the technicians has ensured that the models and test rig are of the highest quality. The Department of Ship Science has provided an excellent environment for academic endeavour. In particular, I would like to thank Chris Satchwell, Malcolm Barnsley, and Shaun Hyde for their support and numerous discussions. Dr J.F. Wellicome for reading the thesis and supervising the work. I would especially like to thank Dr. A.F. Molland for employing me on the rudder and propeller contract, his instruction into the use of steel work in ship design and for all his help and advice. Finally as my long running excuse comes to an end, my thanks to Alison for her patience, support and love.

TABLE OF CONTENTS

LIST OF FIGURES

Appendices

LIST OF TABLES

NOMENCLATURE

a	Local speed of sound (m/s)
A	Rudder area (m^2)
A_i	Area of panel i, (m^2)
AR	Effective aspect ratio, 2S/c
AR_G	Geometric aspect ratio, S/c
A	Angular rotation vector (ordered x, y and then z)
b	Number of bytes
B	Ratio of the distance from panel centroid to that of the length of the largest panel diagonal

c	Chord (m)
C_D	Non-dimensional drag coefficient
C_{Do}	Non-dimensional form drag
C_{Di}	Non-dimensional induced drag
C_f	Coefficient of skin friction
C_l	Local sectional lift coefficient (m)
C_L	Non-dimensional lift coefficient
C_{Lmax}	Maximum developed lift coefficient
C_{Mn}	Normal moment coefficient about rudder stock
C_{Mx}	Moment coefficient about x-axis
C_{My}	Moment coefficient about y-axis
C_{Mz}	Moment coefficient about z-axis
Cp	Pressure coefficient
CP_c	Centre of pressure chordwise, %c, measured from leading edge
CP_s	Centre of pressure spanwise, %S, measured from rudder root
C_Q	Torque dynamometer calibration slope (μ V/N)
c_R	Rudder root chord (m)
C_T	Thrust dynamometer calibration slope (μ V/N)
c_T	Rudder tip chord (m)
Cv	Specific Heat capacity (m^3/sK)
d	Rudder drag force (N/m^2)
D	Propeller diameter (m)
$dC_L/d\alpha$	lift-curve slope (deg^{-1})
D_{ij}	Body dipole panel j's influence at point i
E	Specific energy (m^3/s)
e	Specific internal energy (m^3/s)
F	Body force vector
F,G	Flux vectors
h_d	Distance of leading edge from generator (m)
h_t	Distance of maximum thickness from leading edge (m)
[I]	Identity matrix
J	Advance ratio (V/nD)
k	Scalar constant
K_T	Propeller thrust coefficient
K_Q	Propeller torque coefficient
L	Rudder lift force, normal to free-stream air flow (N)
L	Number of transputer links

M	Mach number
[M]	General matrix with dimension N x N
M	Body moment vector
n	rate of revolution (sec^{-1})
N	Number of panels
N_f	Normal force, normal to centre-line of rudder (N)
Nr	Number of real numbers
Ns	Number of streamwise panels
Nt	Number of chordwise panels
NWs	Number of wake strips
n	Unit surface normal vector
O	Offset vector
P	Propeller pitch (m)
p	Pressure (N/m^2)
P	Pivot vector
Q	Propeller torque (Nm)
r	Radial distance between two points (m)
R	Propeller radius (m)
Rn	Reynolds Number (ρ Vc/μ)
R	General vector of dimension N
RHS	Right-hand side vector [D]ϕ
s	Semi-span (m)
S	Rudder span (m) S = 2s
S_B	Bounding body surface
S_{ij}	Body source panel j's influence at point i
S_u	Speed-up, Ts/Tp
S_W	Wake surface
S	Scale vector
t	Time to pass a communications message (Chapter 3)
T	Temperature (Chapter 3 deg. Kelvin)
T	Number of transputers
T_{calc}	Time to carry out numeric calculation (sec)
T_h	Propeller thrust (N)
T_{ovr}	Time to carry out numeric overheads (sec)
T_n	Time to carry out calculation on n transputers (sec)
Tp	Time to carry out process in parallel (sec)
Ts	Time to carry out process sequentially (sec)

T_1	Time to carry out calculation on 1 transputer (sec)
t/c	Rudder maximum thickness to chord ratio
U_∞	Free-stream speed (m)
U	State vector
U	Velocity vector, (u,v,w), m/s
$\mathbf{U_d}$	Disturbance velocity due to potential field (m/s)
$\mathbf{U_i}$	Total of body and inflow velocity (m/s)
$\mathbf{U_s}$	Undisturbed inflow velocity (m/s)
$\mathbf{U_T}$	Velocity at point (m/s)
v	Disturbance velocity vector (m/s)
V	Wind speed/vessel speed (m/s)
Vc	Calibration supply voltage (V)
V_D	Scaled and offset drag voltage (V)
V_{Dp}	Dynamic pressure voltage (V)
V_L	Scaled and offset lift voltage (V)
V_{Mx}	Scaled and offset Mx voltage (V)
V_{My}	Scaled and offset My voltage (V)
V_{Mz}	Scaled and offset Mz voltage (V)
Vo	No-load voltage (V)
V_P	Pressure transducer voltage (V)
V_{Po}	Pressure transducer zero-wind voltage (V)
V_Q	Torque dynamometer bridge output voltage (V)
V_{Qo}	Torque dynamic zero voltage (V)
V_{Qs}	Torque dynamometer supply voltage (V)
Vr	Radial velocity (m/s)
V_R	Effective axial velocity at the rudder (m/s)
Vs	Supply voltage (V)
V_T	Thrust dynamometer bridge output voltage (V)
V_{To}	Thrust dynamic zero voltage (V)
V_{Ts}	Thrust dynamometer supply voltage (V)
Vw	Circumferential or swirl velocity (m/s)
W_{ik}	Wake dipole panel k's influence at point i
$W_{\infty ik}$	Influence of semi-infinite wake strip at point i
X	Longitudinal distance, propeller plane to rudder leading edge (m)
X_1	Distance of rudder stock from leading edge (m)
Y	Lateral distance between propeller axis and rudder stock (m)
Z	Vertical distance between rudder root and propeller axis (m)

R,aβ	Cylindrical coordinates
t,s	Parametric arc-lengths along three-dimensional space curves
x,y,z	Cartesian coordinates
ε η	Computational space coordinates
α	Angle of incidence (deg)
α_o	Rudder incidence for zero lift (deg)
α_{stall}	Stall angle (deg.)
ρ	Density (kg/m^3)
μ	Dynamic viscosity (kg/m^3)
ξ	Coverage, proportion of rudder span covered by propeller race (λ D/S)
λ	Proportion of propeller race diameter impinging on rudder
γ	Yaw angle (deg.)
γ	Ratio of Specific Heats
ΔCp_{te}	Trailing edge pressure loading
Δt	Time step (s)
$\Delta\phi$	Correction vector/wake strength vector (m^2/s)
ϕ	Disturbance potential (m^2/s)
ϕ	Pitch angle (Chapter 6) (deg.)
Φ	Total potential (m^2/s)
σ	Source strength (m^2/s)
μ	Dipole strength (m^2/s)
ν	Kinematic viscosity (m^2/s)
ω	Angular velocity vector (rad/s)
η	Propeller efficiency
η_c	Code Efficiency
μ	Artificial viscosity
σ	Numerical smoothing factor

CHAPTER ONE
INTRODUCTION

1.1 Aims and Objectives

When a ship is moving ahead the flow passing through its propeller accelerates and rotates. The swirl and acceleration induced in the flow by the propeller alters the speed and incidence of the flow arriving at a rudder aft of the propeller. This controls the forces and moments developed by the rudder. These forces are important in determining the overall performance and manoeuvring characteristics of the vessel. The rudder itself influences the flow onto and through the propeller. This affects the thrust produced and torque required by the propeller.

The development of an accurate means of predicting the flow interaction between a ship rudder and propeller has been the primary aim of this research. The modelling of the flow to allow the prediction of rudder and propeller forces requires considerable computations. Therefore, a secondary aim of the work has been to demonstrate that a cost-effective method of modelling a complex fluid flow is through the use of an array of transputers to carry out the computation.

To achieve these aims the objectives of the work were to:

1. develop an effective method for the implementation of computational fluid dynamics (C.F.D.) flow codes on transputer based parallel computers;

2. demonstrate the ease of C.F.D. code development on such systems by implementing a numerical model of the flow around a ship rudder and propeller;

3. obtain accurate experimental data from wind tunnel tests of a representative ship rudder and propeller combination;

4. validate the numerical method against the wind tunnel test results;

5. develop parametric relationships for estimating the performance interaction between a ship rudder and propeller.

This introductory chapter details the background to the work. The subject of parallel processing using transputers is introduced and the overall layout of the thesis given.

1.2 Background

The coursekeeping and manoeuvring performance of a maritime vessel defines the ability of a ship to maintain or change course and alter speed at a given rate. At present the effectiveness of a vessel's steerage and propulsive systems, for example in hazard avoidance or in approach to port, are not governed by regulation. The need for such regulation becomes more urgent as shipping lanes become more crowded and the tonnage per vessel increases. The specification of vessel manoeuvring characteristics requires a knowledge of how the details of ship design affects those characteristics. Although research into ship manoeuvring has been on-going over many years there is still a fundamental lack of understanding about the forces and moments developed on a vessel while it manoeuvres.

A ship travelling in a seaway generates a complex fluid flow. This makes experimental and theoretical investigations of vessel manoeuvring characteristics difficult. If valid design decisions are to be made it is necessary to understand how the various components of a vessel interact to generate the overall force and moment acting on the vessel. The forward speed and direction of a powered vessel of given hull form are controlled by the arrangement of propulsor and steering systems on the vessel. As a first step in determining overall manoeuvring characteristics the work presented in this thesis investigates the interaction between a ship rudder and propeller. A thorough understanding of the physics of the flow interaction between these two isolated components will then allow the effect of ship hull design and its relationship to manouevring and coursekeeping to be understood.

A detailed experimental investigation of the interaction between a ship rudder and propeller coupled with a theoretical study allows the relative importance of the geometric and flow parameters of the problem to be assessed. To facilitate the

experimental work tests were conducted in the 3.5m x 2.5m low-speed closed return wind tunnel at the University of Southampton. The use of the wind tunnel removed many of the problems associated with conventional towing tank and cavitation tunnel tests. Measurement of rudder and propeller forces and moments for a wide range of geometries and flows allows parametric relationships to be constructed. Rudder surface pressure distributions can be integrated to give the distribution of loading across the rudder surface and allow validation of numerical models. In addition, the wind tunnel is equipped with laser doppler anenomometry (L.D.A.) which allows non-intrusive time accurate measurement of flow field velocities for comparison with theoretical models.

The theoretical study of a ship rudder and propeller combination requires a numerical model of sufficient complexity to account for their mutual influence. A complex numerical model requires considerable computational resources to achieve answers of desired accuracy within a reasonable time. The field of computational fluid dynamics has developed in step with advances in computational technology. Computers which can execute many commands in parallel offer the possibility of large reductions in computational costs but require a different approach to algorithm design to those algorithms used in conventional serial computers. It was decided to carry out the numerical modelling for this work using transputer based parallel computers. These machines represent, at present, the most cost-effective method of parallel processing. They are also the most general type of parallel machine and techniques used for algorithm development have general application. The implementation of the numerical model of rudder-propeller interaction provides a controlled study of the ease with which an actual problem can be developed on transputer based machines.

Therefore, the twin themes of this work have been the detailed investigation of the interaction between a ship rudder and propeller and the study of how this problem is numerically modelled using a transputer based parallel computer. The results of the work have application in the fields of ship manoeuvring, rudder-propeller design and that of computational fluid dynamics.

1.3 Interaction between a ship rudder and propeller

1.3.1 Introduction

The way in which a ship propeller mounted directly upstream of a rudder affects the manoeuvring performance (sideforce and drag developed) of the rudder can be qualitatively answered by any motor vessel helmsman. Similarly, it is well known that the presence of the rudder alters the thrust and torque of the propeller for a given speed of rotation. However, only a limited amount of research has been carried out to quantify these interaction effects. There is a need, as expressed in recommendation 4 of the Manoeuvring committee of the 18th International Towing Tank Conference of 1987 [1.1], for " Further investigation of hull-propeller-rudder interactions, including rudder stock torque, ...". This request for research was again repeated by the same committee of the I.T.T.C. at the 19th Conference in September 1990[1.2].

The understanding of the physics of the flow regime over propeller and rudder combinations is essential if efficient and effective commercial design decisions are to be made. For example, a recent article in Lloyds Register Magazine 100A1 [1.3] discussed the importance of the low-speed manoeuvring of the Bell Pioneer, a new design of cover-less container ship. As a short-sea-route ship is likely to visit several ports a week good manoeuvrability in confined water is required. The Bell Pioneer is fitted with two bow thrusters, a controllable pitch propeller and a Schilling Monovec rudder. This rudder is a proprietary design which enhances the generation of transverse thrust due to the propeller race flow over the rudder when hard-over. At present, such systems can be designed or retro-fitted to vessels. However, it is generally not possible to quantitatively predict whether the fitting of such systems is beneficial to the manoeuvring characteristics or to accurately estimate the economics of operation of a vessel using such a system.

The next section discusses the flow regime around a vessel stern. A review is made of current research and a method of modelling the interaction is proposed.

1.3.2 Description of rudder-propeller flow

Figure 1.1 shows views of the stern arrangement of a conventional ship propeller and rudder. A boundary layer flow develops along the surface of the hull, which, by the time it reaches the propeller and rudder is fully turbulent. This results in a non-uniform wakefield inflow into the propeller. In addition, if the vessel is yawed relative to its direction of motion the whole propeller inflow will be skewed with respect to the propeller's axis of rotation. The presence of a free surface and possible shallow water will also distort the propeller inflow from uniform flow. These alterations from the uniform flow used to determine the open-water performance characteristics of the propeller give rise to cyclical and unsteady loading on the propeller and affect its race.

The flow over the ship rudder is also influenced by the hull boundary layer, ship yaw angle and free surface, but of crucial importance is the relative magnitude of the velocity of fluid in the propeller race to that of the vessel's velocity. For a given propeller pitch ratio setting this is characterised by the non-dimensional advance ratio J, which is defined as:

$$J = \frac{V}{n\,D} \qquad [1.1]$$

where V is the ship's velocity; n the propellers rotational speed; and D the overall propeller diameter. For high values of J (low or zero values of propeller thrust: vessel coasting), the rudder's performance is determined mainly by the influence of the hull, free surface and the ship's angle of yaw. However, for the case of low values of J (high values of propeller thrust: zero or low speed manoeuvres), the performance of the rudder will be determined primarily by the amount of the propeller's race passing over the rudder. At service speed a ship has a moderate value of J and it is important to analyse the influence of the propeller on the rudder and the flow around the stern. In Appendix A the parameters which are expected to govern rudder-propeller interaction are discussed. For a given rudder-propeller arrangement the magnitude of the

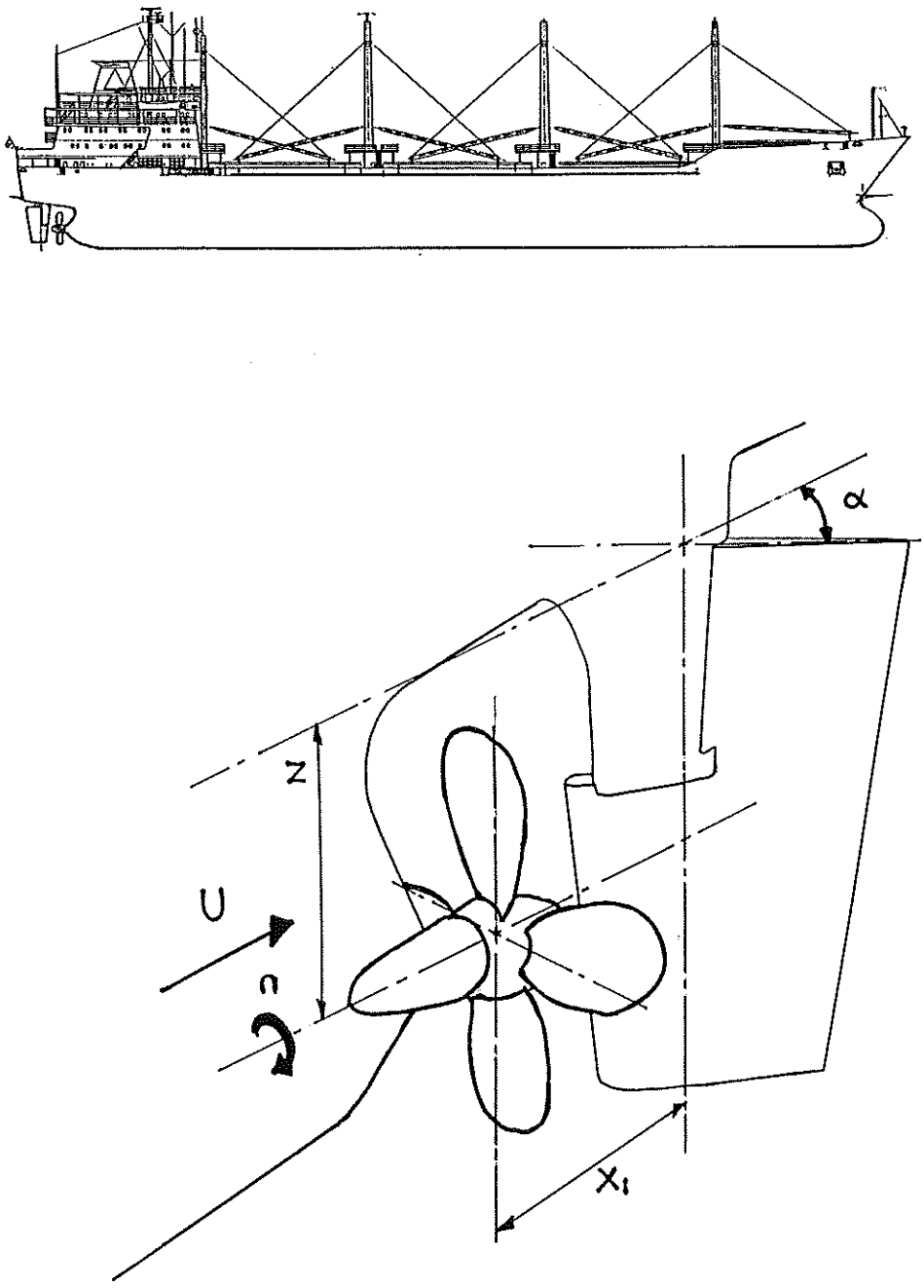


Figure 1.1 View of conventional ship rudder-propeller arrangement

developed rudder force is controlled mainly by the advance ratio J and rudder incidence α .

The standard geometrical arrangement of the rudder relative to the propeller, as shown in figure 1.1, is likely to be the controlling factor in how well the rudder operates for low J values. In general, the centre-plane of the rudder is coincident with the propeller axis of rotation and the lateral separation of the rudder stock and propeller axis(Y) is zero. Lateral separation of the rudder from the propeller axis most commonly occurs on twin-screw vessels to allow for the removal of the propeller shaft. In some cases the propeller wake may not impinge at all on the rudder and free-stream rudder performance information can be used for design purposes.

The relative vertical position of the propeller and rudder (Z) will determine the proportion of the flow developed by the propeller which arrives at the rudder. The longitudinal separation(X) controls the diameter of the propeller race arriving at the rudder and the velocity distribution within it.

The direction of rotation of the propeller introduces asymmetry into the flow. This may result in the rudder characteristics being dependent on whether the helm is to port or starboard. It is essential that when reporting research into the interaction of rudders and propellers that the propeller's direction of rotation is defined unambiguously. In this thesis the conventional Naval Architectural definition that a right-hand screw (clockwise rotation when viewed from aft) has a positive direction of rotation is used.

The speed of rotation and propeller blade load distribution determines the structure of the propeller's race. Also influenced is the local magnitude and direction of flow within the race. Theoretical models of varying levels of complexity can be used to describe the flow downstream of a propeller.

For the purposes of manoeuvring studies, where J is generally low, the race from the propeller is expected to control the flow regime over the rudder. On the basis of this assumption, the effect of the ship hull and the free surface have not been

considered. This removes the complication of the turbulent separated hull wake and the need to model its effect on the propeller inflow. It is hoped that the study of the interdependence of a rudder and propeller will then allow the combined hull-propeller-rudder problem to be considered in more detail.

In summary, when a ship moves ahead its propeller acts to integrate a non-uniform inflow. The resultant race flow is accelerated and rotated as it passes through the propeller. The acceleration and swirl induced in the flow by the propeller alters the local flow velocity and incidence angles arriving at the rudder and hence affects the performance characteristics of the rudder. The rudder itself influences the flow onto and through the propeller. The degree to which this affects the thrust produced and torque required by the propeller is of interest in the field of ship resistance and propulsion. This effect should be considered when modelling the mutual interaction of a ship rudder and propeller.

1.3.3 Current Research

Experimental work into the interaction between ship rudder and propeller combinations has been carried out, both in cavitation tunnels and in towing-tank tests. There are considerable difficulties involved in obtaining the large data set necessary for defining the influence of geometry on rudder-propeller interaction. Tests in towing-tanks suffer from the limited steady state running times whilst for most cavitation tunnels there is considerable time involved in altering test geometries and with the relatively high blockage effects. Surface pressure measurements carried out in water are significantly more difficult than those carried out in a closed-section wind tunnel. Measurement of surface pressure information is essential in determining the local load distribution on the rudder due to the propeller slipstream. Such information is necessary for the satisfactory validation of all but the simplest of theoretical models for rudder-propeller interaction.

Recent work by Stierman[1.4,1.5] concentrated on measuring the propulsive force (or drag) caused on both the rudder and propeller due to their interaction. The dominant parameters considered were the propeller pitch ratio, rudder thickness and

longitudinal propeller-rudder separation. All tests were carried out at a rudder angle of 0° only and sideforce was not measured. The rudder acts to recover the rotational energy of the flow and an important conclusion of this work was that the greater the pitch ratio (P/D) of the propeller the easier it was to recover the rotational energy as propulsive thrust. As rudder thickness increased (t/c), rudder thrust decreased and the thrust and torque of the propeller increased. Also, with increased separation rudder thrust increases and changes in propeller torque and thrust decrease. A rudder may experience a thrust force at low advance ratios and a resistance force at high advance ratios.

Suhrbier[1.6] investigated the influence of a ship rudder on propulsive performance for a series of 5 propellers, again with the rudder at zero angle of attack. It was found that the number of blades had no influence on rudder performance, however, the highest propulsive efficiency (greatest reduction in rudder drag or actual rudder propulsive thrust) came from highly loaded propellers with high pitch ratios(P/D).

English[1.7] investigated the performance of a jet-flap rudder for use in ship manoeuvring at zero and low-speed. In these cavitation tunnel tests a representative propeller and rudder were tested for a range of propeller advance ratios and rudder angles. The principal finding was that the effect of the propeller was to increase rudder stall angle dependent on advance ratio and propeller-rudder separation. Also investigated were the importance of scale effects. It was found that interaction experiments, for full-scale validity, must be carried out at a Reynolds number greater than 1.6×10^6 based on propeller diameter, rate of revolution and rudder chord.

Mathis[1.8] looked at the cavitation performance of high speed rudders for a wide range of rudder section shapes. For rudder side force, similar results to English were found, with the additional conclusion that drag and rudder stock torque are little affected by propeller loading at low angles of attack but are increased at high angles of attack as propeller loading increases.

Kerwin[1.9] carried out a detailed investigation of the performance of flapped-

rudders for a fixed set of longitudinal and lateral separations. The propeller upstream increased the lift curve slope ($dC_L/d\alpha$) of the rudder and that this was independent of the longitudinal separation. However, stall angle was delayed and hence C_{Lmax} increased for reduced propeller-rudder separation.

Using the facilities of the Berlin Model Basin Krackt[1.10] recently carried out a cavitation tunnel study of the effect of rudder section shape on the performance of a semi-balanced skeg rudder performance with and without a representative ship hull form.

Although the geometrical and experimental facilities used differ markedly a summary of the principal features of the rudder-propeller interaction is that for a rudder in a propeller slipstream:

1) sideforce (lift) developed increases with increasing propeller loading;
2) stall angle and as a result C_{Lmax} is increased with increased propeller loading;
3) at zero rudder incidence drag reduces with increased propeller loading and propulsive thrust can be developed at high propeller loading;
4) rudder thickness and longitudinal separation affect rudder drag whereas rudder lift is little changed;
5) increasing longitudinal separation reduces rudder stall angle for a given loading.

The rudder has an upstream influence on propeller performance which increases as the propeller-rudder separation is reduced. The basic effect of the rudder is to alter the operating point of the propeller from that of its open-water performance at a given advance ratio. No work has so far been carried out to quantify the effect for differing rudder incidence.

Attempts to theoretically model the performance of a ship rudder in a propeller race are more limited than the available experimental studies. One of the first attempts was by Lötveit[1.11] who combined momentum theory for the mean axial velocity of water in the propeller slipstream with low-aspect ratio wing theory to predict the rudder behaviour. In conjunction with his theoretical work Lottveit carried out towing tank tests of a model ship with propeller and pressure tapped rudder. The

spanwise and chordwise rudder surface pressure distributions obtained showed different loading distributions for the rudder to port and to starboard and confirms the asymmetric movement of spanwise centre of pressure.

Berlekom[1.12] compared momentum theory with experimental work using a cavitation tunnel for a rudder and propeller mounted on a model hull form and found:
1) a linear relation between propeller loading and rudder lift;
2) momentum theory over-estimates the effect of propeller loading;
3) the effect of propeller loading is dependent on the lift characteristics of the rudder section.

Okada[1.13] considered the performance of a rudder downstream of propeller again using axial momentum theory to estimate the velocity field over the rudder. An empirical relationship between the propeller slip ratio and increase in rudder normal force was developed.

Tsakonas et al.[1.14,1.15] used a sophisticated and computationally intensive un-steady lifting-surface method to directly model a propeller and rudder combination in a representative hull wake field. However, only a limited number of tests were carried out and no un-steady velocity field data was available to validate the method. The approach gave realistic results for the overall forces and moments on the propeller and rudder.

Most practical attempts to obtain a theoretical measure of a rudder's performance characteristics in way of a propeller race have only a limited application. The application of more sophisticated models require considerable computational resources to carry out complete parametric studies. They are also restricted in application because of the lack of detailed experimental studies for validation purposes. For the validation of more physically realistic approaches such as the lifting surface method of Tsakonas a minimum requirement is at least quasi-steady rudder surface pressure measurements. This then allows the local load distribution over the rudder to be compared with prediction. Surface pressure values are best obtained using air as a working medium as this minimises experimental difficulties. The lifting-line approach

used by Molland [1.16] in modelling free-stream skeg-rudder performance, with suitable empirical modifications for tip-vortex effects, gives very good agreement with wind tunnel experimental results. Such an approach with a modified velocity inflow to account for the propeller race ought to be an effective method for modelling the performance of the rudder. An advantage of the method is the low computational requirement which can allow the inclusion of the code directly in ship manoeuvring simulator programs. However, a prerequisite for such an approach is extensive experimental information to account for the strong cross-flow and tip vortex effects.

An alternative approach is to experimentally validate a more sophisticated and computationally intensive model, such as a lifting-surface representation of the rudder-propeller geometry, and then to use the lifting-surface model to modify the lifting-line results. This has the advantage of requiring less experimental tests to produce an effective method for modelling rudder-propeller interaction.

1.3.4 Proposed Rudder-Propeller Interaction Investigation

Investigation of the interaction between a ship rudder and propeller in the absence of hull form, free surface and yawed flow has application to the coursekeeping and manoeuvring characteristics of powered vessels. At present, only a limited understanding of the various parameters which govern the interaction has been gained. There is a requirement for a detailed experimental investigation, an essential feature of which would be quasi-steady pressure measurements on the rudder. The use of a large-scale wind tunnel would greatly facilitate experimental procedures and allow a detailed parametric study to be conducted.

In conjunction with the experimental study an improved numerical modelling of the flow for developing parametric performance relationships is required. It is proposed that the rudder and propeller be modelled separately. Their mutual influence will be taken into account through the modification of their respective inflow velocity field. A quasi-steady approach will be used. The velocity inflow to the rudder is the mean value obtained over a complete cycle of revolution of the propeller whereas the inflow to the propeller is the mean value for a particular radial position over one

revolution due to the upstream influence of the rudder. An iterative approach through the modification to the rudder and propeller's respective inflow velocity field allows the performance characteristics of both the rudder and propeller to be determined.

The quasi-steady approach matches the available experimental data for both rudder and propeller performance. The separate modelling of the two bodies removes the need for an explicit time-domain model of the physical flow between them which varies cyclically and thereby requires a more complex formulation of the fluid dynamics.

The method chosen for modelling the flow on the rudder and on the propeller will be discussed in detail in Chapter 4. For the rudder and propeller this will be a lifting-surface panel approach. This allows both spanwise and chordwise information of the flow over the rudder surface to be obtained. Although it is a detailed method for modelling the flow, a lifting-surface panel approach has a moderate computational requirement and is therefore suitable for parametric investigations of various geometries.

At a later stage this form of modelling approach can be readily adapted to take account of the velocity defect due to the ship hull and that from vessel yaw.

The lifting surface panel method for the parametric analysis of rudder-propeller interaction requires a substantial amount of computational resources (processing time and storage). Therefore it provides a useful test of the effectiveness of the recently developed transputer based parallel computers in solving a real problem.

1.4 Parallel Processing using Transputers

The techniques for the numerical analysis of fluid flows using computational methods have advanced at a great rate over the past three decades. The limiting factor for most computational techniques has been either the computational power available or the required amount of data storage. As computer technology has advanced so has the spatial and temporal resolution of the flows investigated. The method of solution of large-scale computational fluid dynamic (C.F.D.) problems on parallel computers is

an area of research which is rapidly developing as more advanced parallel computers are designed.

It should be noted that the proper use of computers in solving fluid flow problems does not just involve an accurate numerical algorithm but also the pre-processing and post-processing of the solution. In general, a viable method of numerically simulating a fluid flow can be classified into the four distinct stages of:
1) specifying initial conditions, both geometric and flow parameters (pre-processing);
2) numerical analysis, generally using a reduced form of the Navier-Stokes equations;
3) interpretation of the solution (post-processing);
4) validation of the obtained solution.

Recent developments in micro-processor technology, such as the transputer which can be connected together to form large-scale parallel computer, have resulted in parallel computers with sufficient computational power (speed of calculation) to allow complex fluid flows to be investigated. Parallel computers, as opposed to the hitherto conventional serial (or sequential) machines, can carry out many processing tasks simultaneously.

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 improvement in speed of execution when increasing the number of transputers in an array is a basic measure of the performance of a particular application.

The development of the transputer, methods for developing parallel algorithms, and techniques of programming transputer based parallel computers are discussed in Chapter 2.

The motives for investigating the use of parallel processors, specifically transputer based machines, are the increasing demand for large scale computations and the corresponding need to reduce computation time and cost. The ability to improve the performance of conventional sequential computers is limited by the technology for manufacturing their electronic components. The finite limit on sequential computer performance can be circumvented by sub-dividing a computational task into T parallel processes. For wholly independent processes, where the end result of each process does not affect or depend on the results of any other process, the parallel computational time Tp is equal to the time to carry out the calculation on one computer Ts divided by the number of computers over which the computation was divided (T).

$$Tp = \frac{Ts}{T} \qquad [1.2]$$

Most numeric calculations cannot be so easily sub-divided, and as a result this ideal situation will not be achieved. The granularity of a process task is a term used to describe the ease with which a computational process can be sub-divided into wholly independent processes. The more fine grained a problem, the smaller the overall process sub-divisions have to be. For example, matrix inversion using direct Gaussian elimination is a fine grained problem as each element in the inverted matrix is dependent on operations carried out on all of the other elements of the original matrix. The class of parallel machine and number of component parallel processors determines how effectively an algorithm of a given size can be solved.

An algorithm will run most effectively if the sub-division of processes matches that of the number of parallel processors used. Effective use is defined as minimising the amount of time spent by the component processors either communicating between processes or being idle whilst awaiting data to process.

The reason for using transputers to carry out large-scale computational fluid dynamics problems is that of cost. Jong[1.17] found that transputer based parallel computers gave more performance per unit cost than any currently available computer system. Although computer performance comparisons are by necessity crude and are valid only for a limited time span such work does indicate the trend for large reductions in computational cost through the use of parallel computers.

A considerable number of theoretical or paper studies have been carried out into the use of parallel computers. However, unless applications are actually tried and tested using transputer systems, the hidden costs of software development and verification cannot be properly assessed. As sequential computers and hence software have a finite performance limit, it is inevitable that larger-scale computations will increasingly be carried out on parallel computers. The practical implementation of CFD problems on parallel computers exposes the needs and requirements for the development of future generations of parallel computers. A reluctance to invest in parallel technology because of existing investment of resources in software development and computer expertise will be exposed as parallel computational costs reduce more rapidly than those of sequential machines.

Overall, in this work two standard CFD type applications have been investigated. These were the explicit 2-D compressible Euler equations (Chapter 3) and the implicit lifting-surface panel method (Chapters 4 and 5). The principal investigation carried out into the complex fluid flow associated with rudder-propeller interaction is considered to be a good test of the ease with which current parallel processing technology can be applied to CFD in general.

An engineering applications methodology has been used with regard to software development. This is opposed to the approach of optimising code to extract the maximum possible performance from a given system. The advance of technology will quite rapidly supersede the maximum parallel computational performance so obtained. However, it is thought that the general principles used in this work and results obtained will remain applicable to future parallel computers and software.

1.5 Thesis layout

The work developed and described in this thesis can be divided into three main threads. These are: the development of software tools to allow the implementation of general computational fluid dynamic codes on transputer based computers (Chapters 2 and 3); the implementation and development of a model of the interaction between a ship rudder and propeller on a transputer based parallel computer (Chapters 4,5 and

7); and wind tunnel measurements of rudder-propeller interaction to validate the numerical model and carry out a study of the parameters which control the interaction (Chapters 6 and 8).

Chapter 2 serves as an introduction to the use of transputers in producing powerful parallel computers and the software techniques used to solve numeric problems in parallel. The method of measurement of the performance of a parallel algorithm is detailed. Performance measurement of algorithms is coupled to the assessment of the relative performance of various serial and parallel computers. A primary consideration is made of the techniques used in developing parallel software on transputers.

In Chapter 3 the software design of a communications harness for controlling fluid dynamic algorithms running across an array of transputers is described. This harness allows the separation of the software which controls the communication of information around a network of transputers from the specific fluid dynamic analysis. As a result, the implementation of different types of computational fluid dynamic methods is greatly eased. As part of the development of the communications harness a two-dimensional finite volume scheme for solving the compressible Euler equations was implemented. This verified the communication harness software methods. The data transfer performance of the final harness design is given across various sized arrays of transputers.

The theoretical basis of a lifting-surface panel method is the subject of Chapter 4. The method has been developed to allow the modelling of both free-stream (open water) rudder and propeller flows. The method is verified against an analytical solution, other lifting-surface codes, and published experimental data for various test cases. A frozen or adaptive wake geometry can be used for both rudder and propeller cases. An explicit trailing edge pressure Kutta condition is used. Surface geometry definition using parametric cubic splines was developed to allow general lifting-surface geometries to be investigated with the capability of dealing with multiple-body problems. From the lifting-surface solution the resultant three-dimensional velocity field can be obtained.

Chapter 5 describes the software implementation of the lifting-surface method for a transputer based parallel computer. The main computational tasks are the calculation of the panel influence coefficient matrix, solution of the resultant system of linear equations, wake adaption, and velocity field generation. A study was carried out into the techniques for solving systems of linear equations. The relative performance of both iterative and direct methods solving the typically dense set of equations generated in lifting surface methods is given. This information can be used as a guide to the performance of similar computational techniques on a transputer based system. A discussion is given of the capabilities of transputer based parallel computers in solving implicit algorithms such as the lifting-surface panel method.

In Chapter 6 the detailed wind tunnel investigation of the interaction between a typical ship rudder and propeller combination is given. The design of the experimental test rig, propeller and rudder models is described. The method of data acquisition and analysis was verified against previously published experimental work. Forces and moments were measured on the rudder using a five-component strain gauge dynamometer. Detailed surface pressure measurements were made on the rudder. A rotating strain gauge dynamometer was used to measure the propeller's thrust and torque.

The validation of the developed numerical model against the experimental results is described in Chapter 7. Results are produced for the four cases of rudder freestream performance, propeller (open-water) performance, influence of rudder on propeller, and influence of propeller on rudder.

Validated numerical model and experimental results were used to carry out a parametric study which is reported in Chapter 8. Results are given for variation in propeller pitch ratio, advance ratio, rudder aspect ratio, chord, span, longitudinal, vertical and lateral separation of propeller and rudder. A final section gives relationships for sideforce for use in ship manoeuvring studies and rudder-propeller design data.

The main conclusions are presented in Chapter 9 with reference to the stated aims and objectives. References are presented at the end of the main body of text and are numbered according to chapter and order of occurrence. Figures and tables are included within the text.

CHAPTER TWO

TRANSPUTER FUNDAMENTALS

2.1 Introduction

This chapter describes the development and terminology associated with the Inmos[2.1] transputer. 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. The techniques used to design a parallel (concurrent) algorithm as opposed to a conventional serial algorithm are discussed.

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.

2.2 Transputers

Transputers are a range of high-performance VLSI(Very Large Scale Integrated) technology devices, developed by Inmos Ltd[2.1], which consist of local memory, four high speed two-way links and a micro-processor unit all mounted on a single silicon chip. Figure 2.1 shows a schematic of the layout of a typical transputer.

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

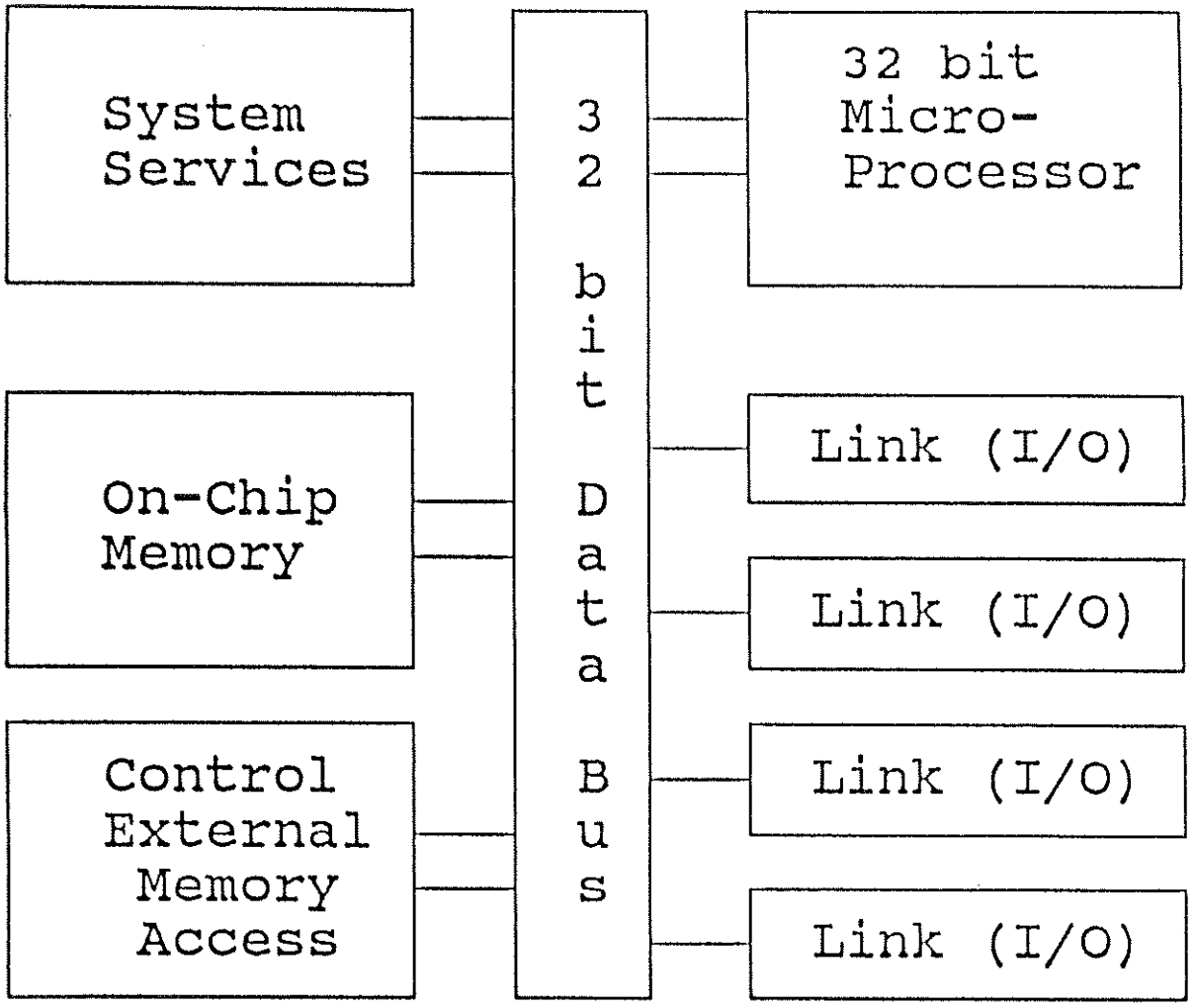


Figure 2.1 Transputer Schematic

of transputers. However, the software developed to run on such machines can be developed on small array machines and then re-configured for the large array machine. The ability to scale the parallel computer to suit the problem is an advantage of transputer based computers.

The transputer is a coarse grained device as it is a powerful micro-processor in its own right. However, the provision of high speed communication links allow even fine grained problems to be tackled in an efficient manner. The granularity, or impedance effect, of matching processes to processors has the result that for a given size and type of computational problem there will be a finite number of transputers to

perform the calculation most effectively. Each transputer spends a maximum amount of the total time processing rather than communicating or waiting for data.

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 and Lonsdale[2.2] investigated the parallelism of a commercial fluid dynamics software package ASTEC[2.3] 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 sequential machines.

As shown in Figure 2.1, each transputer can have external memory assigned to it. However, access to this external memory is slower than to the on-chip local memory. As each transputer is assigned its own memory transputer machines have a distributed memory. If other transputers require access to a particular transputer's memory communication is required between the two transputers. This can occur directly if the two transputers are neighbours or indirectly through information passed through intermediate transputers. Communication between transputers occurs by point to point access across the direct links between the individual transputers.

A range of transputers has been developed. At present, the T800 transputer has the best performance. Hey[2.4] gives typical performance figures for the 20MHz version of the T800 for well-optimised Occam code of 1Mflops (1 million floating point operations per second) and for the same application using presently available Fortran or C compilers of up to 0.5 Mflops. The four serial data links, which operate concurrently with the micro-processor support data transfer of up to 2Mbytes/sec bi-directional communication on each link. Internal memory of 4K of fast access memory (SRAM) is provided, and up to 4GBytes of memory connected to each transputer.

The development of the series of transputer devices and similar circuitry is a continuing process. The next generation of devices is currently under development and it can be assumed that processing speeds and data transfer rates will substantially improve. Therefore, given the continuing development, applications such as that described in this thesis are a valid exercise as they provide necessary information for ensuring that the computational requirements for fluid dynamic problems are considered in future generations of parallel computers.

2.3 Occam2 Language

2.3.1 Introduction

Occam2 is a high level language which exploits the concurrency of the transputer[2.5,2.6]. It is designed to express concurrent algorithms as communicating sequential processes(CSP). Occam2 allows the programmer to produce programs which explicitly deal with the communication necessary in the construction of parallel algorithms. Communication takes place along one-way channels joining two processes. Message protocols define the type of message to be transferred along these channels. When two processes are located on different transputers which need to communicate, the channels are mapped onto the links between the transputers. The Occam2 communication model uses processes executing in parallel which can only transfer information along these one-way channels. The executing processes cannot share memory except through the communication of the information across channel links. Therefore, to use and manipulate the same information two parallel processes require dedicated channel links between them.

As stated in the Occam2 reference manual[2.6] *"the founding principle of Occam is a minimalist approach which avoids unnecessary duplication of language mechanism, and is named after the 14th Century philosopher William of Occam who proposed that invented entities should not be duplicated beyond necessity. This proposition has become known as Occams razor."* This approach has resulted in a hierarchical structure of executing processes. Each process is constructed from a number of smaller processes,

which in their turn use a number of yet smaller processes. This results in a straightforward means of program development. Individual processes are used as building blocks in constructing the overall complex program. This structured approach is ideal for the development of efficient parallel algorithms.

Fundamental to Occam is the ability to verify programs (or processes) through the application of mathematical proof techniques. This feature provides security which is not possible in conventional high-level languages such as C, Fortran or Pascal. A result of the ability to mathematically prove a process is that transformations can be applied to improve the efficiency of code execution.

For this work it was decided to write all software in Occam2. This allows the communications model and concurrency of Occam2 to be fully exploited. Parallel compilers for conventional (sequential) high-level program languages such as C and Fortran do exist. However, as mentioned in the previous section, even at best their performance only achieves half the speed of Occam2. This is not surprising given that Occam2 is designed for parallel processing whereas the necessary forcing of a sequential language into a parallel model will always result in a loss of efficiency. The argument for developing parallel methods using existing languages such as Fortran centres around the previous investment of time and programming effort in developing existing codes. However, the structured model of Occam2 ensures that the translation of existing FORTRAN77 code into Occam2 is a straightforward task. For example the transcription of approximately 1,000 lines of FORTRAN77 code for the finite volume Euler solver described in Chapter 3 was a simple task involving about two days of effort. As is true of most developed software, the greater the structuring of the software (i.e. effective use of procedures/subroutines) the easier the transcription process becomes.

In a similar manner to the evolution of the transputer, the language and software techniques used to control them are continually advancing. A continued insistence on the use of old-style programming languages is a retrograde step. This point is emphasised by Long[2.7] who used LISP in developing software to run on the Connection Machine (CM-2) and found considerable reduction in software development

time on the CM-2 as opposed to that on a classical serial computer.

An introduction to the basic principles and constructs of Occam2 is given in the following section 2.3.2 and in greater detail in the Occam reference manual[2.6] and by May[2.8]. The environment in which Occam2 programs are developed and run is described in section 2.3.3.

2.3.2 Constructs

Occam2 is a fixed format language. The hierarchical structure of Occam2 is enforced in the source code by the use of a double space indentation between each successive level. The number of pairs of spaces indicates the level of the process. The outermost level has no indentation. A useful and unique feature of Occam2 source code editors is that of the "fold" facility. A fold is used to enclose a series of statements or procedures and effectively hide them from view unless required. This allows the overall structure of a program to be seen as a series of nested folds. A closed fold is represented by three dots (...) which may be followed by an identifying comment. The extent of an open fold is delimited by three curly braces: {{{ at the beginning and }}} at the end. The double indentation and fold notation will be illustrated while explaining the main constructs.

The fundamental constructs of Occam2 are the **PAR** (parallel) and **SEQ** (sequential) statements. A **SEQ** command indicates that the statements at the next sub-level are to be executed sequentially i.e. one after another. This is shown in an example of code as:

```
SEQ
  ... do this
  ... do that
```

where the program will execute the statements in fold *do this* and then those in fold *do that*. This is the standard order of execution of for example Fortran or Pascal. The

PAR command states that the following processes at the next sub-level are to be executed concurrently. That is:

```
PAR
  . . . do this
  . . . do that
```

will result in *do this* and *do that* being carried out at the same time. A loop command such as the Fortran 'FOR' or Pascal 'Do' is replaced by either:

```
SEQ i=0 FOR n
  . . . do this
```

or

```
PAR i=0 FOR n
  . . . do that
```

For these statements *do this* will be executed in order n times and *do that* will be carried out n times simultaneously.

A process consists of nested statements. The order of execution of each sub-level is determined by whether it is a PAR or SEQ, for example (as shown below): the folds *do this* and *do something else* (shown open) execute sequentially. In the execution of *do this* the fold *do that* is be carried out simultaneously n times. Following this the algebraic instructions in *do something else* are sequentially executed.

```
SEQ
  {{{. . . do this
  PAR i=0 FOR n
    . . . do that
  }}}
  {{{. . . do something else
  SEQ
    x:=2
    y:=x+2
  }}}
```

Occam2 forbids the manipulation of a variable by two parallel executing processes. This is for deterministic reasons as it would not be possible to decide, until execution, which process would alter the variable first and thereby the final result is not uniquely determined. To aid the execution of **PAR** processes the Occam2 variable type of **CHAN** (channel) is defined. This provides a means of communicating information between two parallel processes. This **CHAN** type of variable allows inter-process communication to be explicitly carried out in a deterministic fashion. The type of information to be transferred across a named channel has to be first specified using the statement **CHAN OF**. For example,

CHAN OF INT *passmessage*:

declares that the information passed through channel *passmessage* consists of a single integer (INT) messages. The character '!' is used to denote that a message is to be output (sent) and character '?' that it is to be input (received) through the specified channel. A particular channel is only for one-way communication between processes. Two-way communication is achieved by using two channels connecting the two parallel processes. The passing of a message through a channel acts to synchronise two processes running in parallel. Two sequential processes can never successfully communicate as the first process will only execute as far as the first channel command and then stop.

A **CHAN** can be declared for any type of Occam2 variable and for mixed types through the use of a **PROTOCOL** definition. A **PROTOCOL** can also be used to define a number of different mixed type messages which can be sent across the same channel using a tagged protocol and a **CASE** statement to differentiate between them.

A simple illustration of a correct communication is given over the page:

```
CHAN OF INT passmess:
PAR
  {{{   first process
  SEQ
    ... do this
    passmess ! 20
    ... carry on
  }}}
  --   (the symbols '--' are used to indicate a comment)
  {{{   second process
  SEQ
    ... do that
    passmess ? X
    ... do something else
  }}}
```

where the first processes executes *do this* and then attempts to communicate the integer '20' out along channel ***passmess***. Simultaneously the second process will execute "do that" and then awaits an integer input into variable X through channel *passmess*. When both processes have reached the channel statement communication occurs and the execution of **carry on** and **do something else** then carry on independently.

Processes can communicate with many other concurrently executing processes along dedicated channels. The **ALT** (alternative) construct is used to accept the first channel communication arriving at a process. This can be used with the **WHILE** statement to produce a multiplexing process (i.e. from many input channels there is a single output channel).

The correct termination of parallel processes is essential. A **PAR** process terminates when all the concurrent sub-processes within it have terminated. A useful multiplex process illustrating the correct termination of a process is:

```
BOOL finito:
SEQ
  finito:=FALSE
  WHILE finito=FALSE
      ALT
      chan1in ? X
          chanout ! X
      chan2in ? X
          chanout ! X
      chan3in ? X
          chanout ! X
      chanfin ? X
          finito:=TRUE
```

where variable X is input on either of channels ***chan1in***, ***chan2in***, or ***chan3in*** and then sent out of the process through channel ***chanout***. This process will continue accepting and sending messages until there is an input on ***chanfin*** and then the boolean(**BOOL**) variable finito is set TRUE and the WHILE loop finishes and the process terminates.

A useful channel (**CHAN**) type is that of **TIMER**. This provides a means of accessing, either setting or reading, the transputers internal clock. The value of the clock is an integer value of the number of 'ticks'. For the low-priority operation of a T800 transputer one tick equals 64 micro-seconds. This allows the time to execute an operation (or process) to be measured. This is illustrated in the following example for timing process *do this*, where ***clock*** is defined as a channel of type **TIMER**.

```
TIMER clock:
INT StartTime,FinishTime,TotalTime:
SEQ
  clock ? StartTime
      . . . do this
  clock ? FinishTime
      TotalTime:=FinishTime MINUS StartTime
```

This section has not been intended as a comprehensive introduction to Occam2 programming but to highlight the important constructs which are used later in describing the software developed for this research.

2.3.3 Environment

An Occam2 program can run either on a single transputer or across many transputers. On a single transputer the processor executes commands sequentially but implements concurrent processes by dividing its time and continually switching between the parallel processes. The Occam2 construct **PRI** is used to state which of two parallel processes is to be given priority by the processor. This utilises the T800 hardware's ability to run at high and low-level priority.

For a multi-transputer system the outermost level of the program is a series of processes running in parallel. Each process will be mapped onto a single transputer. Information passes between the transputers (processes) through channels linking the transputers. At present because transputers have 4 bi-directional links only one channel can be mapped onto each direction of a link. Therefore, each transputer can only have 4 input and 4 output channels. The next generation of transputer based machines will have no such restrictions through the use of the concept of 'virtual channels'. The transputer hardware will effectively multiplex all the channels used onto the number of physical links available.

At present, most single or multi-transputer systems are connected to the outside world (keyboard/display/memory storage) via a host computer system. The means by which programs are developed is reviewed by Hey[2.4]. The software for this research was developed using the Inmos T.D.S. (Transputer Development System)[2.5]. This provides an integrated environment for creating, editing, compiling, and running program across single or multi-transputer system. T.D.S. runs on the transputer connected to the host computer and accesses the facilities of the host computer.

Software may also be compiled using the Inmos Toolsets utility. This compiles

Occam2 code on a conventional sequential computer. To run the executable code on a transputer system it has to be first loaded onto the transputer network through the host computer and then started. This method is used mainly for multi-user transputer systems.

2.4 Transputer Systems

At the University of Southampton there are a number of commercial and development transputer machines which have been used for this work. Table 2.1. lists the capabilities of the machines used.

Table 2.1 Available transputer systems

Machine	Number of Transputers	Memory (kbyte)
Ship Science System	4 x T800	1000
Parsys Supernode	16 x T800	256
Parsys Meganode	128 x T800	256
Meiko Computing Surface	19 x T800	8000

The amount of memory connected to each transputer determines the maximum size of numeric problem that can be run on a given system. The SSTS has five T800 transputers each with 1Mbyte of memory. It provides an excellent low-cost development environment and is also a powerful processing facility in its own right.

The links between individual transputers on the Ship Science Transputer System have to be manually connected to alter the topology of the transputer array. The three

commercial machines used allow the individual transputer link settings to be made from within the software. All machines are accessed via a control (Host) transputer linked to an external computer providing keyboard, screen and disk storage support.

2.5 Concurrent Algorithms

The choice of a particular concurrent algorithm to carry out a task, depends on the size of the individual computational processes and the computer system upon which the algorithm is to be executed. Fundamental to this decision is the degree of independence of the parallel components of the overall process.

An wholly independent process requires no additional information from other processes while it is executing so its execution can proceed without delay. However, if two processes are dependent they effectively synchronise execution when communication is required between them. This may result in one process waiting for the other. Also, the physical time used during the control of the communication of information is not available for the processing of data.

The strategy for the construction of parallel algorithm is dictated by the ease with which the algorithm can be broken into independent processes or processes which require a minimal exchange of information. There are three basic approaches (Hey[2.4]) for writing parallel algorithms.

1) Farm: A central control process sends independent data packets out to worker processes. Each worker processes the data and sends a processed data packet back to the central controller and waits for another data packet to process. As each worker is an independent unit, increasing the number of workers will increase the speed with which data packets can be processed. However, the speed with which the control process can send and receive data limits the obtainable performance. An example from nature is the retrieval by bees of nectar. The larger the number of worker bees the greater the possible amount of nectar which can be collected. However, the actual collection rate is limited by the rate at which bees can enter/leave the hive.

2) Pipe (Algorithmic): Data packets are sent along a line of parallel executing processes. Each parallel process carries out a particular part of the overall algorithm before passing the processed data on to the next process in line. The maximum speed of data processing is determined by the processing time of the slowest individual process. Increasing the number of workers will not necessarily improve performance unless all workers can process data at a greater rate. This is similar to the piped architecture of the present generation of supercomputers.

3) Geometric: The data to be processed by the algorithm is divided into a number of sub-regions and each worker completely processes its allocated sub-region of data. If necessary, data from other sub-regions are communicated between the individual workers. The addition of extra workers either allows a larger data set to be processed in the same time or allocates smaller amount of data to each processor and thus processes the information more quickly. Geometric parallelism is used for problems which can be sub-divided into processes requiring minimal exchange of information. For processes with a small ratio of communications to data processing, the speed of performance should be roughly proportional to the number of allocated workers.

An important consideration in the choice of strategy for the parallel algorithm is the ease with which the software can be developed. Where possible, software should be written to allow scaling with the data set and number of processing units. For example, the algorithm used to process a tightly defined size of problem on a given number of processing units could be optimised to use a pipe approach. However, if at a later date a different parallel computer or size of data set were to be used a major rewriting of software would be required.

2.6 Parallel Algorithms for Computational Fluid Dynamics

The correct choice of parallel algorithm for a fluid dynamic problem is dependent on the specific numeric analysis to be performed. However, a significant

number of numeric methods involve the use of standard techniques and some degree of generalisation about approach is therefore possible.

A wide range of options are available for CFD algorithm development using the most general class of parallel architecture machines (M.I.M.D.), such as transputer networks. However, the granularity of the algorithm compared to that of the transputer network must be considered. For example a Farm approach for explicitly solving the two-dimensional Laplace's equation rapidly becomes communication bound as each task sent out to an individual transputer is processed quickly. The speed of processing is limited by the rate at which information can pass in and out of the central control process.

The type of parallel computer on which the algorithm is to be implemented also influences the approach. For instance, a S.I.M.D. (single-instruction-multiple-datastream) machine such as the Connection Machine CM-2 carries out the same instruction on all its processing units at the same time, leading to algorithm development along a Farm approach as the other algorithms are difficult to implement. The Connection Machine CM-2, details of which are given by Long[2.7] and Wake[2.9] consists of 65,336 processors and is a very fine-grained machine. Long[2.7] used this machine, programmed in L.I.S.P., to carry out massive 3-D Euler/Navier-Stokes problems. It was concluded that the code ran roughly 10 times faster than similar codes on vector supercomputers and that massively parallel machines offer revolutionary as opposed to evolutionary improvements in performance. Wake[2.9] had similar conclusions that such machines are practical and efficient to use after implementing a 3-D rotary Navier-Stokes solver using Fortran 8X on the CM-2. Both these examples are of extreme cases of computational effort and massive scale of problem. It should be noted that medium and small-scale CFD type problems can also make use of the economies and more powerful processing capabilities of smaller scale parallel computers.

The most common approach to flow modelling is Eulerian where flow variables are solved at a fixed field point in space or on a body surface. The numeric techniques

used to solve for the unknown flow variables at a given point can be divided into three general groups:

1) Implicit - where all the variables for all the points within the domain are solved simultaneously by using such techniques as direct matrix inversion or iterative techniques such as Gauss-Siedel elimination.

2) Explicit - where the variables for a particular point are updated based only on information from its neighbouring points. This is carried out for all points in the domain and the process repeated until the solution has converged to a given accuracy.

3) Mixed - where the variables or spatial dimensions are split so that some information is solved by an implicit method and the rest by an explicit stepping method.

The principal considerations for a parallel CFD algorithm, mounted on a transputer based parallel computer, are that:

1) there is an efficient distribution of data throughout the transputer network to minimise data communication requirements;

2) an effective method is used for the calculation and solution of the flow variables which ensures maximum use of the processing capabilities of all the available transputers;

3) it is easy to extract relevant engineering information from the eventual solution.

These three considerations directly equate to the first three stages of carrying out a fluid dynamic analysis, described in Chapter 1 as (1) initialisation, (2) numerical Analysis, and (3) Interpretation. The choice of parallel algorithm has to reflect the demands of all of these stages.

As the number of spatial dimensions and numeric variables (degrees of freedom)

increases the amount of data storage required rapidly increases and the capacity of the computer to process information is limited by its memory storage facilities. The distributed nature of memory on a transputer network lends itself to geometric parallelism. Each transputer is assigned a part of the overall data set. No single central control transputer would be able to cope with the overall memory requirement. Although data is to be geometrically distributed this does not necessarily force the solution algorithm to use a geometric structure.

For the case of an explicit method, geometric parallelism is the obvious technique. This is because each field point only requires information from its nearest neighbours within the data set. Information then only has to be communicated across the geometry sub-domain boundaries when the neighbouring information is located on a different transputer. If the sub-domain area of data is large compared to the amount of data around the sub-domain edge the solution method will be efficient. Efficiency is defined in this context as minimising the amount of time an individual transputer spends transferring/communicating data across its boundaries compared to carrying out the numeric analysis on its sub-domain of data.

For implicit methods the choice of parallel algorithm is not so obvious. This results from the need to solve all points simultaneously. This implies that each point is influenced by all others. For a geometric algorithm the effect of each point's influence has to be communicated throughout the network of transputers. Therefore, a large amount of communication traffic is always required for implicit techniques.

Implicit methods are difficult to efficiently implement in parallel. An important conclusion to be drawn from this is that if an implicit method is made to run efficiently on a parallel computer an explicit technique should be even more efficient. This conclusion is solely based on a comparison of the efficiency of use of the parallel computer and not the numerical efficiency of implicit as opposed to explicit methods.

The efficiency of a parallel CFD algorithm will increase as; the number of spatial dimensions increases; the number of variables (degrees of freedom) per point

increases; and the method of the solution change from implicit to explicit. In general, the number of degrees of freedom at a point is related to the complexity of the numeric formulation. So a steady-state implicit, potential (inviscid, irrotational, incompressible) formulation provides a 'worst-case' test performance of a transputer network in carrying out C.F.D. For instance, Robinson[2.2] showed that an implicit finite-element Navier-Stokes solver using geometric parallelism gave reasonable performance in comparison with a vector Cray2 computer. Although not a 'worst case' the results indicate that a geometric parallelism strategy is viable for implicit methods.

An additional benefit for the use of geometric parallelism for both data distribution and for the solution algorithm is that the problem may be easily scaled. Extra transputers can be added for a given problem size by decreasing the sub-domain mapped onto each transputer. Similarly, increasing the number of field points in the solution results in a larger number of field points per transputer. By ensuring each transputer is assigned the same number of field points the total computational load can be evenly balanced throughout the network, minimising the time individual transputers are dormant.

It can be argued that the increasing sophistication of sequential algorithms results in their greater efficiency. In constrast for parallel computation the trend should be for the design of simple algorithm. Therefore, for general problems (implicit or explicit) a geometric algorithm should be used and this will result in reduced software development time.

2.7 Control of Communications

The efficient communication of information is essential for the development of implicit algorithms. For large arrays of transputers information may have to be transferred across many links through adjoining transputers. The route information takes through a network and the time it takes to reach its final destination are important parameters in determining the performance of the parallel computer.

A process running on a transputer has to carry out two functions: numeric processing, and through-routeing of messages destined for other transputers. It is logical to divide the overall process executing on a transputer into two concurrent processes corresponding to these two functions. One process deals with all messages passing in and out of the transputer's 4 links and the other process carries out all the numeric processing.

The communication requirements of fluid dynamic algorithms will be broadly similar. The process dealing with message passing is referred to as a communications harness. A general purpose communications harness greatly facilitates the development of a wide range of different numeric algorithms. Such a harness was produced as a necessary first step in developing techniques for implementing CFD algorithms on transputer based parallel computers. Chapter 3 details the development of a communications harness to be used for CFD applications. To aid the development of the harness an existing explicit two-dimensional finite volume Euler solver was implemented in parallel and used to verify the approach.

2.8 Measurement of Parallel Algorithm Performance

The assessment of a particular algorithm or more usually a computer's performance is a tendentious subject. This is especially true of commercial machines and applications where a company's future livelihood may be determined by how well its computer performs on a given benchmark test.

The difficulties associated in performance assessment are considerably increased when comparing serial and parallel architectures. If an algorithm is particularly suited to one architecture, a general comparison of the relative merits of the two machines cannot be made. Also of importance in assessing performance is the realism of the operating conditions under which the performance test is carried out. For example, a single-user on a dedicated IBM 3090 will obtain several orders of magnitude better performance than as one of many users in the more common multi-user batch queue environment. The absolute cost of a unit of processing time also influences what is

considered as acceptable performance. The capital outlay, maintenance costs, and personnel associated with a particular computer system dictate the minimum processing cost to obtain a solution within a given time-scale.

Leaving aside the above qualifications, the performance of a given system is simply measured as the total time required to carry out a particular application. Comparing similar programs on serial and parallel machines is a laborious task if it is to be done properly, and from a research viewpoint is of limited long-term use. Technology is continually advancing for all machine architectures and such a comparison exercise will be quickly out of date. As an example of such a comparison Long[2.7] gives data for a comparison of the maximum processing speed of a wide range of current serial and parallel architectures.

A particular program may achieve a certain performance but how efficient is it at using the full capabilities of the computer? This question is especially important when addressing the performance of parallel architecture machines. The communication of data amongst processors and their synchronisation is an overhead in addition to the time required to carry out the numerical analysis. Therefore, an efficient parallel algorithm will be one in which these overheads are minimised.

For networks of transputers where an identical problem can be run across different numbers of transputers, performance information is required to determine whether there is an optimal number of transputers for a given problem size. The efficiency of code (η_c) running on a particular transputer system is defined as:

$$\eta_c = \frac{T_{calc}}{T_{calc} + T_{ovr}} \qquad [2.1]$$

where T_{calc} is the time to carry out all the numeric components of the process and T_{ovr} the associated communications overhead. The more rapid the communications harness, the lower T_{ovr} and the greater the maximum possible efficiency. For geometric

algorithms, where communication only occurs with adjacent transputers, the efficiency η_c will be approximately constant for all transputers. This assumption neglects possible differences in computation time T_{calc} at the domain boundaries. For this case η_c is evaluated by timing the performance of a fixed size problem on one transputer. This gives T_{calc}. Then, by measuring the total time (T_{calc} + T_{ovr}) it takes a multi-transputer system to carry out a problem with each transputer's sub-domain size equal to the total problem size on a single transputer, the code efficiency can be found. Information on η_c for a particular sub-domain size may be used to predict the performance of larger networks of transputers. More sophisticated methods have to be used for predicting the scaled performance of implicit geometric algorithms where the amount of communication and hence T_{ovr} is dependent on the number of transputers in the array.

As has been mentioned already, communication between parallel processes is an overhead to the overall efficiency of a concurrent numerical algorithm. The more quickly messages can be passed between processes (processors), the less delay there is in the calculation and the more efficient the algorithm.

The size of problem assigned to each transputer influences the efficiency. Increasing the size of problem on each transputer increases the code efficiency as proportionally less time is spent communicating. However, the calculation will still take longer. Code efficiency is a useful parameter in determining the number of transputers necessary to perform a calculation and the amount of local memory each transputer requires.

Comparing the overall calculation time for an identical problem run on different sizes of transputer arrays indicates the actual speed-up in performance obtained.

$$S_u = \frac{T_1}{T_n} \qquad [2.2]$$

Speed-up S_u is defined as the ratio of the time taken to perform an identical calculation on one transputer(T_1) to the time taken on n transputers (T_n). A theoretical speed-up of n would occur if no communication took place between transputers. The closeness of the actual speed-up to the theoretical maximum is the conventional measure of the benefits of increasing the number of processors in a parallel architecture machine.

For a constant domain of data, as further processors are added there will be a smaller sub-domain on all transputers. The ratio of communications to calculation increases as there are proportionally more edge nodes. Therefore code efficiency decreases and there is a diminishing return on the improvement in speed-up. This was noted by Amdahl[2.10] who stated that if a computer has two speeds or modes of operation during a given calculation the slow mode will limit the performance. For a transputer based machine the fast mode is the independent numeric calculation on each transputer and the slow mode the requirement for communication or synchronisation. Patton[2.11] discusses in detail performance limits for parallel processors. Work from references[2.12,2.13] shows that a more realistic assessment of parallel performance should be based on the assumption that the problem size is scaled to that of the available processing power. This inverts the convention of measuring speed-up to that of asking how long a given parallel problem would take to run on a serial machine rather than the other way round.

The Occam2 **TIMER** facility allows detailed examination of the time taken for calculation and communication on individual transputers as well as measuring the overall program timings. This allows information on speed-up and code efficiency to be obtained. In this work the times used in calculating code efficiency and speed-up have been given in seconds based on the T800's clock time of 64 μ Sec per tick. This allows the actual performance of the code to be assessed.

In summary, a true bench-mark of the performance of a numerical algorithm is a difficult quantity to assess. The performance of identical code can vary widely when

implemented on different computers. For instance with limited memory machines, a trade-off has to be made between recalculating parameters when they are needed and storing them for the whole calculation. A larger problem can then be solved, but the calculation will take longer. On computers with shared access, the actual speed of solution as opposed to the units of computational time used will depend on the user's priority and how many others are using the computer.

2.9 Conclusion

A transputer is a device which allows the most general class of parallel computer architectures (M.I.M.D.) of arbitrary size to be constructed. Parallel algorithm software development is eased by the use of the high-level computer language Occam2. This gives, for well written code, at least twice the performance obtained with parallel versions of Fortran and C.

The choice of a particular parallel algorithm depends on the application, size of problem, and the capabilities of the parallel computer to be used. On transputer based machines with distributed memory a geometric approach to both the numeric algorithm and data distribution minimises software development time. It also allows scaling of problem size and number of transputers used.

An implicit scheme for solving a fluid dynamic problem with a small number of degrees of freedom per node provides a worst-case test of the efficiency of a parallel computer.

The passing of information around a distributed memory computer, such as a transputer network, is best handled by a communications harness process running in parallel with an application process. A general purpose communications harness for computational fluid dynamics algorithms greatly facilitates the rapid development of a wide range of fluid dynamic algorithms.

CHAPTER THREE
COMMUNICATIONS HARNESS

3.1 Introduction

The main component of the work described in this chapter is the development of a general purpose communications harness (referred to as Harness). Harness is capable of transferring information between various processes executing in parallel on an arbitrary sized array of transputers. It runs in parallel with a numerical fluid dynamic algorithm and controls the transfer of information around an array of transputers. The separation of a particular application into a communications harness process and a numerical process facilitates the development of numerical algorithms. The communication harness can be used for all numerical applications and is independent of the number of transputers used in a particular parallel computer.

An explicit finite volume numerical algorithm for the solution of the compressible Euler equations is a good test of the effectiveness of transputer based parallel computer in solving more general fluid problems. The Euler solver is used to provide a variable computational load for investigating the performance of the communications harness in relation to the numerical algorithm. The alteration of the number of finite-volume cells allocated to each transputer changes the ratio of numerical computation to communication and hence tests the effectiveness of the communications harness.

The development of the communications harness design and its testing using the two-dimensional Euler solver was presented at the 17th International Congress of Aeronautical Sciences in Stockholm, Sweden in September 1990[3.1].

Following the development of the original harness and implementation of the Euler solver a modified harness process was written which significantly improved the performance of the harness. Performance figures for this final harness design are included and compared with those for a similar harness code.

3.2 Communication Harness Definition

The passing of data messages between parallel processes needs to be carefully controlled. The means by which messages pass between transputers should be made independent of the processes running on each transputer. As discussed in Chapter 2 it was decided that a communications harness process should be developed to deal with all types of messages and control their routeing from source to destination through a transputer array.

The function of the harness, as defined in this work, is that of ensuring that messages sent between processes running on different transputers are passed as rapidly as possible across the link(s) joining them. This message passing has to be done in a controlled manner which does not adversely affect the operation of the numerical processes on the intermediate transputers. Additional requirements are a minimal amount of memory for storage and that all messages reach their destination.

One of the principal aims in constructing the harness process was to allow various strategies for solving concurrent fluid dynamic algorithms to be tested. The harness is designed to remove the need for rewriting the inter-process communications when either the topology of the transputer array or the numeric algorithm is altered. The main use of the communications harness is for the implementation of geometrically parallel algorithms However, it also possible to use the harness process for implementing both Farm and Pipe algorithms.

3.3 Original Harness Design

3.3.1 Introduction

The communications requirements of geometrically distributed fluid dynamic algorithms are of a similar nature and can be broadly classed into the following four categories:

1) Nearest neighbour communication: Information sent from one transputer to any of the transputers connected to one of its four wire links, e.g. the updating of node information across a common physical boundary.

2) Directed message: Information sent from one transputer to a specific transputer located somewhere in the array of transputers. The route the message takes through the array has to be controlled to ensure that it arrives at its correct destination.

3) Global message passing: Information to be sent to or collected from all the transputers in an array, e.g. examining global convergence to solution of a series of processes.

4) Process information: The user can only directly access information on the control transputer connected to the transputer array. Therefore, status messages, graphical data, and information from other transputers generated when verifying code have to be sent via the communications harness to the control transputer.

The maximum length of an individual message and the number of messages the harness can deal with at any one time are important design parameters. Large numbers of small messages are liable to increase the ratio of communication time to calculation time, and may overwhelm the harness. However, large messages require greater memory storage and this reduces the size of the maximum data set an individual transputer can operate on.

To aid in the development of the harness the individual components of the harness were written as independent processes. These could be separately tested and if necessary upgraded without altering the overall structure of the harness. The use of this approach reduced software development time and is a main advantage to the use of a structured language such as Occam2. The inherent flexibility of a modular approach also simplifies future changes to the overall structure of the harness.

3.3.2 Control of Communications

The main purpose of the harness process is to control all interprocess communication. The numerical processes simply input or output messages when necessary (similar to READ/WRITE commands in Fortran) and the harness ensures that the information is sent to the correct destination.

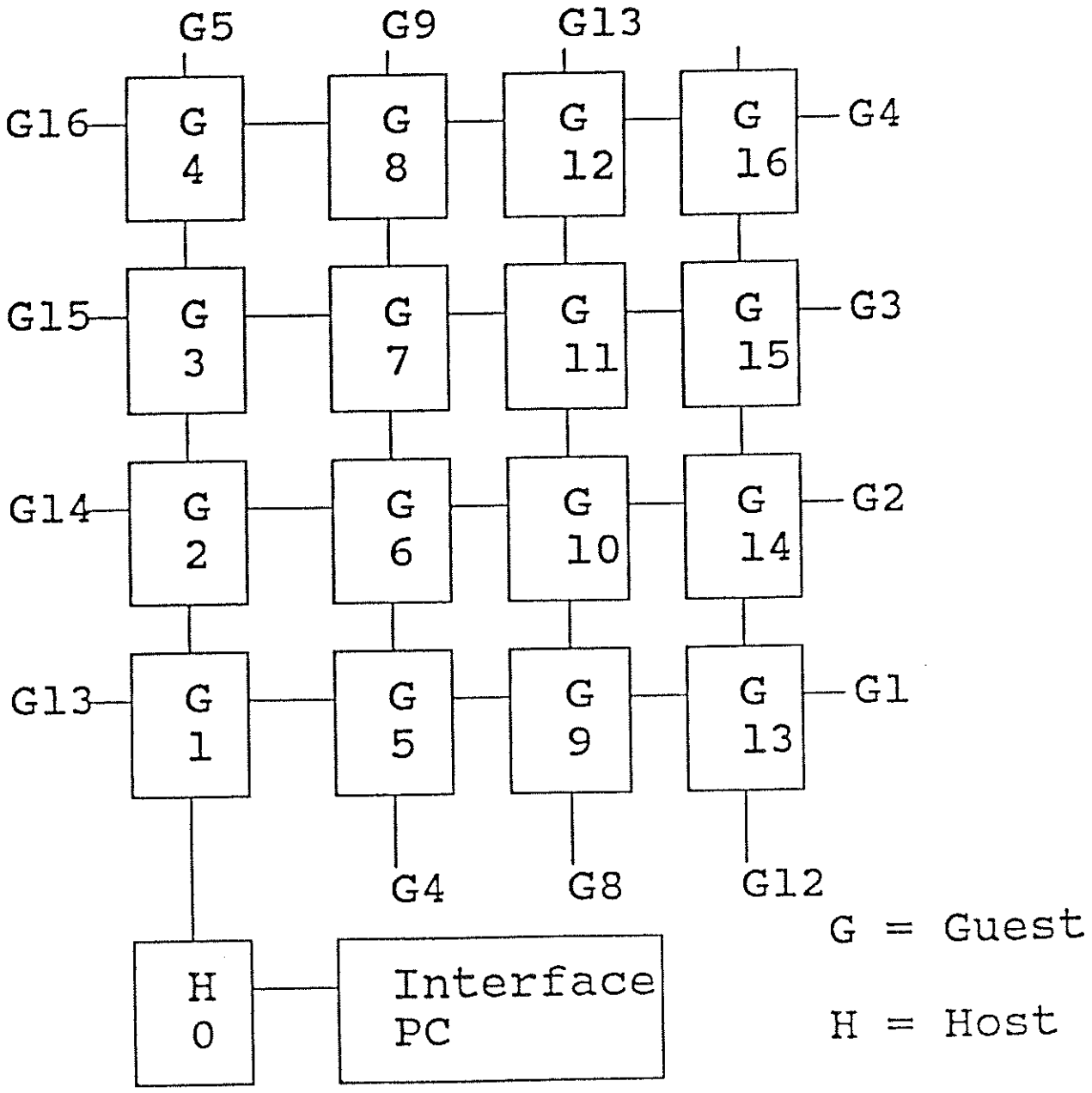


Figure 3.1 Interconnection of an array of transputers

Figure 3.1 shows a representative transputer array. One process (Host) is used to control all the others (Guests). The Host process is mounted on the transputer connected to the PC interface allowing keyboard input, screen display and disk access. The individual Host and Guest processes can communicate with each other only via the Harness which runs in parallel with the application process on each transputer. Harness determines the route each message will take through the transputer array to get to its destination.

3.3.3 Harness Structure

At the outermost level, the code mounted on each transputer consists of two processes running in parallel as shown in Figure 3.2. These are the user's process (Host or Guest) and the harness process. This is written in Occam2 for the Guest process as:

```
PRI PAR
  Harness(Lk1i, Lk1o, ...., Lk3o, Lk4i, Lk4o, inguest, outguest)
  --
  Guest(inguest, outguest)
```

Channels *Lk1i*, *Lk1o* to *Lk4i*, *Lk4o* are the eight channels mapped onto the four input and four output links of the transputer. The two channels ***inguest*** and ***outguest*** link the Guest (or Host) process to Harness. The protocols of these channels are described in the next section. Harness is run at high priority to ensure that messages are throughput as rapidly as possible.

The basic data flow structure of Harness is shown in Figure 3.3. Harness consists of a number of processes running in parallel. The Occam2 structure for Harness is as follows:

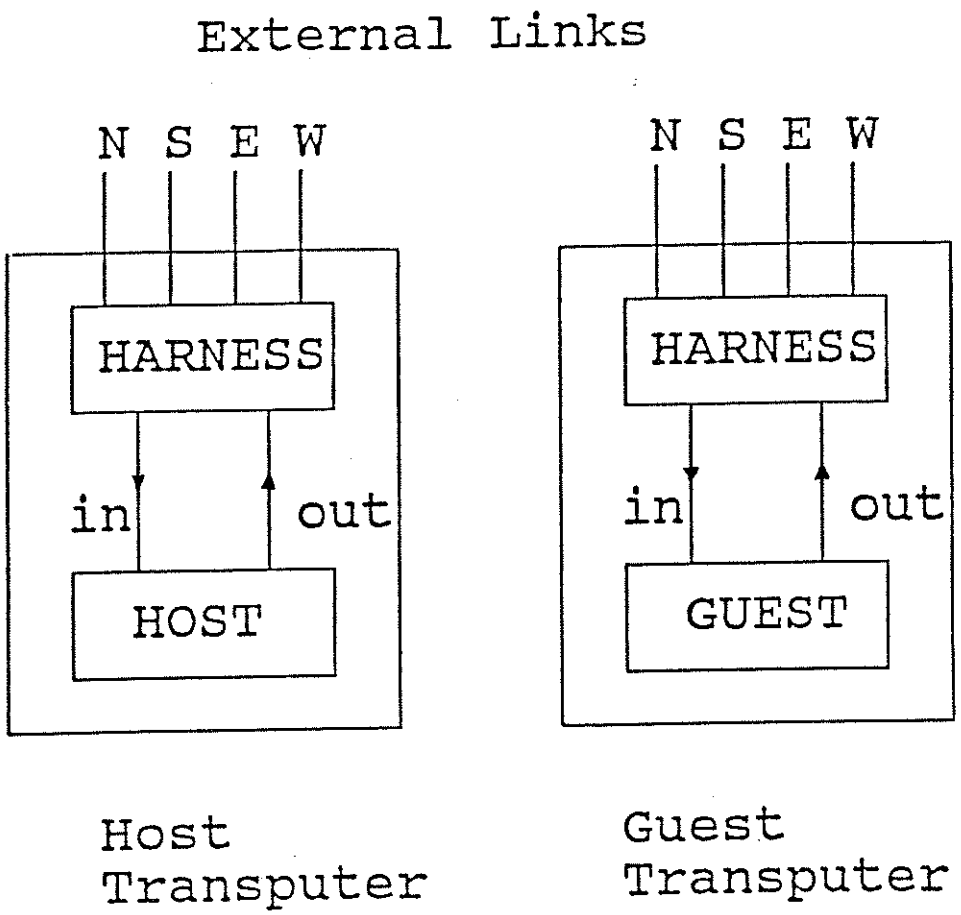


Figure 3.2 Schematic of code running on an individual transputer

SEQ

... channel definitions

PAR

Multiplex(*Lk1i,Lk2i,Lk3i,Lk4i,ToWhichBuffer,f1*)

WhichBuffer(*ToWhichBuffer,ToBuffer1,ToBuffer2,f2*)

Buffer(*ToBuffer1,ToRte1,f3*)

Buffer(*ToBuffer2,ToRte2,f4*)

Encode(*outguest,ToRte3,f5*)

Route(*ToRte1,ToRte2,ToRte3,Lk1o,Lk2o,Lk3o,Lk4o,todec,f6*)

Decode(*todec,inguest,f7*)

FinishHarness(*f1,f2,f3,f4,f5,f6,f7*)

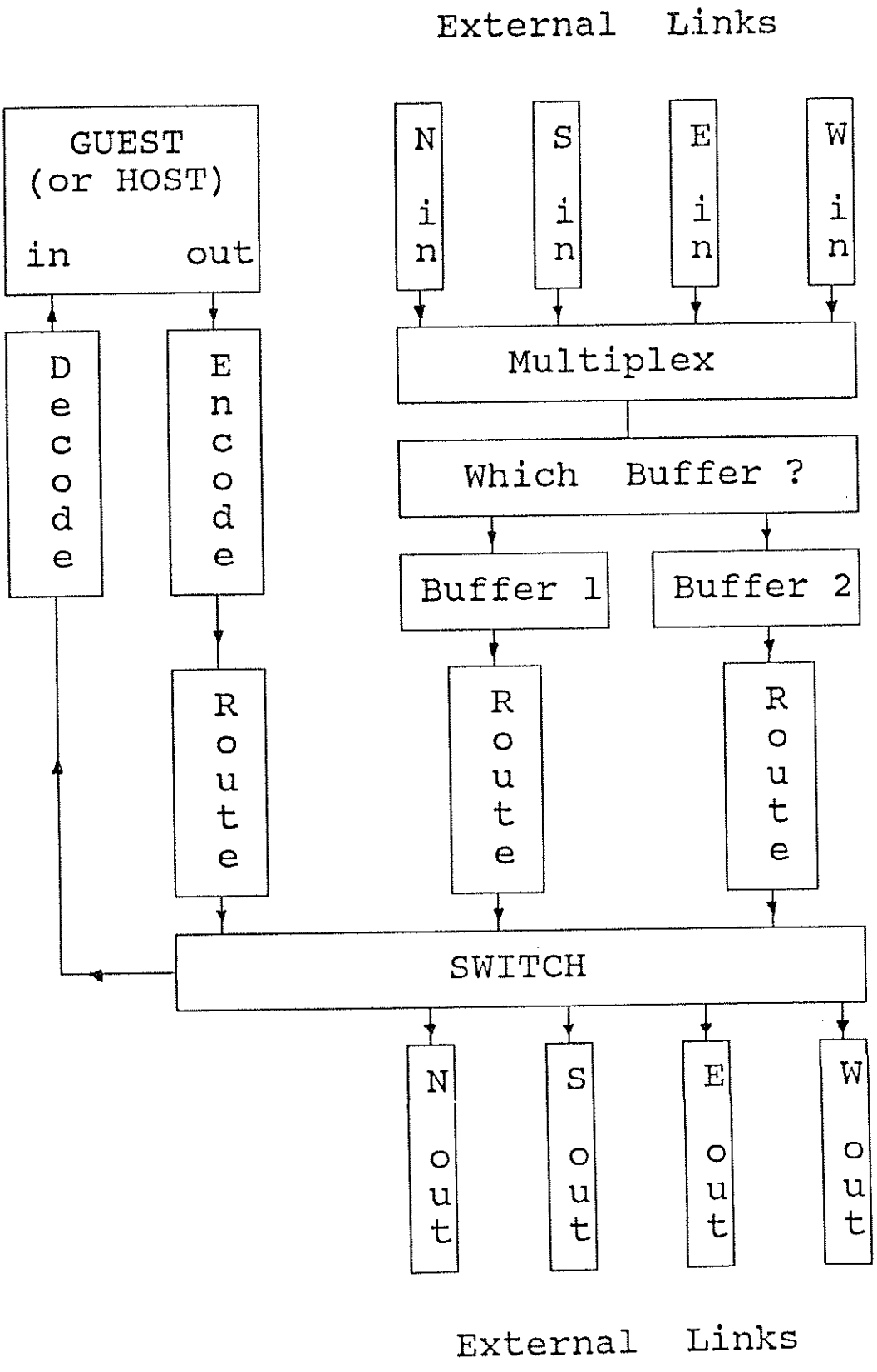


Figure 3.3 Harness data flow schematic

Channels *f1,...,f7* are of type BOOL and are controlled by process FinishHarness to switch down all the other parallel processes to ensure correct termination when an application is finished. All the other channels are of the same protocol as ***inguest*** and *outguest*.

Data flows into the harness process from any of the four external links connected to other transputers(channels *Lk1i* to *LK4i*). All four channels are multiplexed onto a single channel *ToWhichBuffer*, reducing the need for a buffer for each individual link. The Multiplex process, as shown below, is simply an ALT statement which passes an incoming message to the WhichBuffer process.

```
SEQ
  finito:=FALSE
  WHILE finito=FALSE
    ALT
      f1 ?
        finito:=TRUE
      Lk1i ? <message>
        ToWhichBuffer ! <message>
      ... identical for other 3 input channels
```

The WhichBuffer process allocates incoming messages to the first available empty buffer. At present, the harness uses just two input buffers. The Buffer process, when it is empty, accepts a message from WhichBuffer and then passes it on to the Route process.

Messages from the Guest (or Host) process are encoded into a standard form in the Encode process which also acts as an input buffer. A Route process is assigned to each input Buffer which determines the destination of a message. Based on the message destination, it decides which external link should be used to pass on the message or whether the message is for the resident Host (or Guest) process. Route sends the message to the Switch process which ensures that the message is passed out

of the Harness either on one of the four external links or to the Decode process. The Decode process translates messages into a form readable by the Guest (or Host) process.

The structure of the harness ensures that only relevant messages are sent to the resident Guest (or Host) process on each transputer. Other messages are passed through the Harness, leaving by the link determined by the routeing strategy.

All harness processes run in parallel with each other. Occam2 currently allows processes to be assigned one of two priority levels (High or Low). The overall Harness process and all its component processes are set to High priority, so that incoming messages are dealt with immediately they arrive. This ensures that other transputers further along the array will have to wait a minimum amount of time for messages.

Appendix B details the important design parameters of Harness.

3.3.4 Message protocol

Different types of messages are allowed in Occam2. The content of messages is defined by protocols. The Harness process uses two simple message protocols:

1) User Protocol. This allows the Guest (or Host) process to send messages of type: integer (INT), boolean (BOOL), real(REAL32), and string ([]BYTE)variables; and one or two-dimensional arrays of real numbers. Appendix B lists the Occam2 definition of this protocol. Larger data structures can be built up from a number of messages containing different parts of the structure. A component of the protocol defines the destination of a message as either to be sent to the transputer connected to the North, South, East, or West links, or the Host process. An individual Process number can also be stated and this is used when the Host process communicates to individual Guest processes or Guest processes communicate with each other. This allows all the four type of messages for computational fluid dynamic problems previously defined to be

sent. The protocol is used for the inguest and outguest channels connecting the Harness and Guest (Host) processes.

2) Harness Protocol. This comprises a header of 25 bytes which defines the type of message, the identity number of the destination and source transputers (see Appendix B). The remainder of the message is a variable length one-dimensional array of bytes. The use of this protocol simplifies the passage of messages through the Harness, each component process only having to deal with a variable length string of bytes. The Harness process uses this protocol to pass messages through it and across the transputer's external links.

The Guest and Host processes use the User Protocol to send and receive messages. The protocol allows flexibility about the sorts of messages which can be sent between processes. As well as data the protocol also allows development messages (Debugging information) to be sent. In the Harness process the Encode and Decode processes use the protocol definitions to translate messages between the two protocols. The Harness Protocol is used for passing messages around the array of transputers. The Route process controls the definition of the 25 byte header and uses it to determine the route and destination of messages.

3.3.5 Routeing Strategy

Figure 3.4 illustrates the simple strategy chosen for routeing messages through a rectangular array of transputers. Messages from the Host process or between guest processes pass along the lowest row of transputers until they reach the column which includes their destination.

Each transputer is assigned an unique identity number (IdNo) based on its position within the transputer array. The Host transputer identity number is defined as zero. For a rectangular array of transputers a grid scheme is used to define a particular transputers identity number. That is, IdNo = (I*Ny) + (J+1), where I is

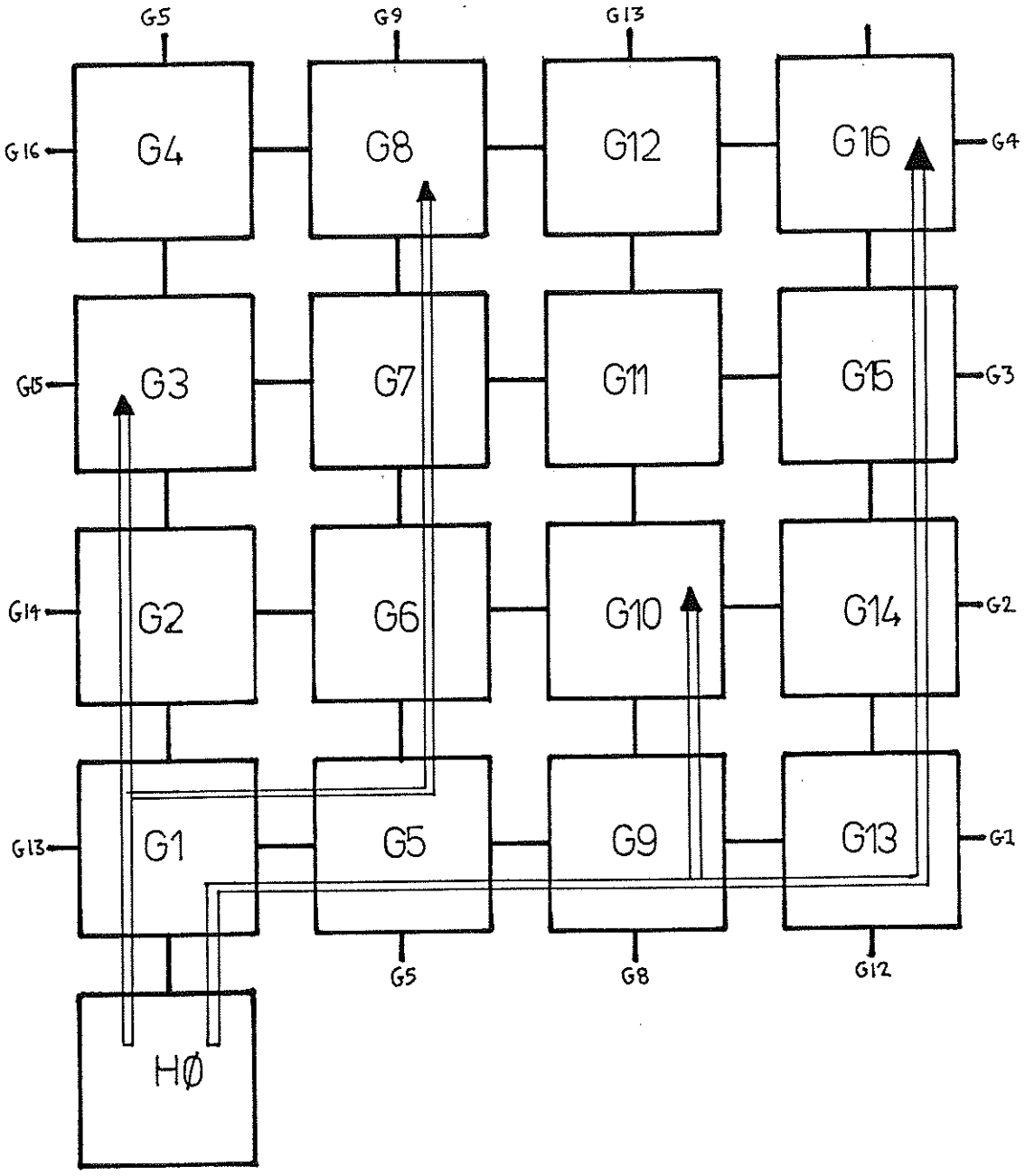


Figure 3.4 Routeing strategy for square array of transputers

the column number of the transputer and J is its row number. Nx is the number of transputers in each row and Ny the number in each column. The row index J varies between 0 and (Ny-1) and column index I varies between 0 and Nx-1. The unique relationship between I, J and IdNo allows the Route process to use the Harness protocol header information to determine the output link on which the message should

be sent on.

Nearest neighbour communication occurs directly between adjacent transputers. For computational problems with a high density of messages a more subtle routeing strategy might be required to avoid bottle neck problems. However, the grid based routeing strategy has the advantage of simplicity and hence has a low computational penalty in determining the correct link.

Figure 3.4 also shows the connection of end of row transputers to each other to provide a closed ring. This provides a short route for messages between the end of row transputers.

3.3.6 Communications Overheads

The time each transputer spends dealing with communications rather than executing the numerical algorithm is an overhead to efficient numeric processing. The Harness on each transputer has to process messages at high priority to allow the overall calculation to proceed without delay. Therefore, Harness has to process messages with as little computational requirement as possible. Similarly, the time a user process (Guest or Host) is idle waiting for an incoming message is an overhead and reduces the efficiency of the code. Good practice is to ensure that there is always something for each GUEST process to carry out while awaiting incoming messages. This effectively 'hides' communication time and minimises the associated delays. As there is a finite time associated with sending a message along a transputer link generally it is better to send one long message rather than many shorter messages.

3.3.7 Buffering

The amount of buffering required depends on the likely amount of communication and the effectiveness of the routeing strategy in avoiding bottle-necks. The number of buffers is restricted by the need to provide storage space for each

additional buffer. A nearest neighbour problem gives a light communications load across a transputer array, with messages only crossing between adjacent transputers. A simple routeing strategy and a small number of buffers can adequately control communications.

A potential problem when controlling concurrent communicating processes is deadlock. This occurs when every process is waiting to receive a message and no messages are sent to allow the calculation to proceed. The provision of buffers and a routeing strategy which avoids bottle-necks minimises the risk of deadlock. However, even harnesses using 'deadlock' free routeing strategies will fail if a user process generates messages which no process is willing to receive. Eventually all buffers fill up with redundant messages and deadlock occurs. Additional buffering requires additional message storage, as mentioned earlier, the harness design used in this work uses two buffer processes. Careful thought in the design and implementation of the Guest and Host process is the best method of avoiding deadlock and in reducing communication overheads.

3.4 Validation of harness approach

3.4.1 Introduction

An explicit two-dimensional finite volume scheme for solving the compressible Euler equations was used to validate the approach of separating the software into parallel processes of a numerical algorithm and communications harness. This Euler solver has been implemented to run across a variable sized array of transputers. The method used is based on a cell-vertex scheme and a multiple-grid scheme may be used to accelerate convergence. The use of a geometric parallelism allocates the same number of cell-vertices to every transputer in the array. The explicit scheme only requires information to be transferred between neighbouring transputers. This allows the performance of the transputer network to be evaluated by either altering the number of transputers for a given number of cell-vertices or for a fixed number of

transputers by altering the total number of cell-vertices.

A two-dimensional explicit finite volume scheme developed by Ni[3.2] for solving the compressible Euler equations was implemented. A Fortran77 version[3.3] of this method was available and this was translated into Occam2 to form the basis of the Guest and Host processes.

3.4.2 Explicit Two-Dimensional Finite Volume Euler Solver

3.4.2.1 Formulation

The basic formulation consists of an solution of the unsteady Euler equations where **U**, **F**, **G** are four component vectors which correspond to the mass, cartesian x and y momentum, and total energy conservation equations. This can be written in compact vector form as:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} + \frac{\partial G}{\partial y} = 0 \qquad [3.1]$$

where the state vector **U** has components

$$U = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} \qquad [3.2]$$

the flux vectors **F**, and **G** have components

$$F = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u(E + (\frac{p}{\rho})) \end{bmatrix} , G = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho v(E + (\frac{p}{\rho})) \end{bmatrix} \qquad [3.3]$$

and the total energy per unit mass is given by

$$E = e + \frac{1}{2}(u^2 + v^2) \qquad [3.4]$$

These are coupled with the auxiliary equations for a perfect gas:

$$p = (\gamma - 1)\rho e \quad and \quad e = C_v T \qquad [3.5]$$

A steady-state solution is obtained by explicit time-marching from an initial condition and applying appropriate boundary conditions at the edge of the physical domain. Expressed in integral conservation form the equation becomes:

$$\frac{\partial}{\partial t}\int\int U\,dA + \oint (F\,dy - G\,dx) = 0 \qquad [3.6]$$

Ni's method uses a cell-vertex method where at each time-step the integral of fluxes through the cell edges are summed around the cell to give a first-order difference in state vector Δ U at the cell centre using the finite-volume approximation.

$$\Delta U_c = \left[\left(\frac{F_1 + F_2}{2} \right) \Delta y - \left(\frac{F_3 + F_4}{2} \right) \Delta y + \left(\frac{G_1 + G_2}{2} \right) \Delta x - \left(\frac{G_3 + G_4}{2} \right) \Delta x \right] \frac{\Delta t}{\Delta x \Delta y}$$

[3.7]

Where C refers to the centre of the finite-volume cell and 1,2,3,4 the four corners of the cell. The difference in fluxes at the cell centre is distributed to the four corners of the cell using Ni's distribution formula for each corner:

$$\delta U_1 = \frac{1}{4}\left(\Delta U_c - \left(\frac{\Delta t}{\Delta x} \right) \Delta F_c - \left(\frac{\Delta t}{\Delta y} \right) \Delta G_c \right)$$

$$\delta U_2 = \frac{1}{4}\left(\Delta U_c - \left(\frac{\Delta t}{\Delta x}\right)\Delta F_c + \left(\frac{\Delta t}{\Delta y}\right)\Delta G_c\right)$$

$$\delta U_3 = \frac{1}{4}\left(\Delta U_c + \left(\frac{\Delta t}{\Delta x}\right)\Delta F_c + \left(\frac{\Delta t}{\Delta y}\right)\Delta G_c\right) \qquad [3.8]$$

$$\delta U_4 = \frac{1}{4}\left(\Delta U_c + \left(\frac{\Delta t}{\Delta x}\right)\Delta F_c - \left(\frac{\Delta t}{\Delta y}\right)\Delta G_c\right)$$

The signs of the second-order correction terms ΔF and ΔG depend on the position of the cell centre relative to the corner and has an effect similar to the use of upwind or downwind differencing.

$$U_1^{m+1} = U_1^m + (\delta U_1)_A + (\delta U_1)_B + (\delta U_1)_C + (\delta U_1)_D \qquad [3.9]$$

The state vector U_1 at each vertex is updated by summing the corrections from the four neighbouring cells A,B,C,D.

3.4.2.2 Time step and Numerical smoothing

To converge to a steady-state solution local time-stepping is used. The time step for each finite-volume cell is set to the maximum that will keep the numerical scheme stable. The formula given by Ni[3.2] is used.

$$\Delta t \leq \min\left[\frac{\Delta x}{|u| + a}, \frac{\Delta y}{|v| + a}\right] \qquad [3.10]$$

Where a is the local speed of sound for each cell.

It is necessary to provide numerical smoothing to capture shock features and prevent oscillatory growth of errors. The method used is that given by Ni where smoothing terms are introduced into the distribution formula as follows:

$$\delta U_1 = \frac{1}{4}\left(\Delta U_c - \left(\frac{\Delta t}{\Delta x}\right)\Delta F_c - \left(\frac{\Delta t}{\Delta y}\right)\Delta G_c\right) + \mu\left(\overline{U} - U_1\right) \qquad [3.11]$$

where

$$\overline{U} = \frac{1}{4}\left(U_1 + U_2 + U_3 + U_4\right) \qquad [3.12]$$

and

$$\mu = \sigma\left(\left(\frac{\Delta t}{\Delta x}\right) + \left(\frac{\Delta t}{\Delta y}\right)\right) \qquad [3.13]$$

The artificial damping factor σ is in the range 0 to 0.1.

3.4.2.3 Initial and Boundary conditions

The cell-vertex scheme ensures that the state vector values U are calculated at the boundaries of the physical domain. As an initial condition the flow is assumed to be at the freestream value. The finite volume approximation to the integral equation expresses the difference in state vector U at a cell-centre in terms of fluxes across the four edges of the cell. Six different types of boundary condition have been implemented for cell edges which are on the boundary of the domain of interest. Mapping sub-regions of the physical domain onto arrays of transputers is thereby simplified. The join between two sub-regions on different transputers is classed as a boundary to the sub-region and is specified during the initial problem set up. The boundary conditions are:

(1) Internal, where two edges join on the same transputer;

(2) Normal, a boundary between sub-regions across which fluid can flow;

(3) Inflow and (4)Outflow, for flow in and out of the domain which is nearly normal to the freestream direction, updated using the method of characteristics to determine the domain of influence;

(5) Solid, is treated as a reflective boundary with flow kept tangential to the surface (Hall[3.4]);

(6)Free, for boundaries at a distance where changes to the far-field can be assumed negligible

3.4.2.4 Convergence

The calculation is assumed to have converged to a steady-state when the maximum change in the x-momentum (ρu) component of the state vector **U** is everywhere less than 1.0×10^{-5}.

3.4.2.5 Multiple Grid

To improve the rate of convergence of the flow solver the Multiple-Grid scheme of Ni[3.2,3.5] was implemented. The coarsest level of updating information is restricted to the area of grid mapped onto each transputer and the boundary information is treated in the same manner as for the transfer of information on the finest level of calculation. Only minimal modification to the code was required to fully implement the Multiple-Grid scheme. Figure 3.5 shows the saw-tooth cycle for transferring cell-vertex corrections δ **U** to coarser and coarser grids. After each transfer operation, the corrections were interpolated back to the finest grid and the state vector **U** updated.

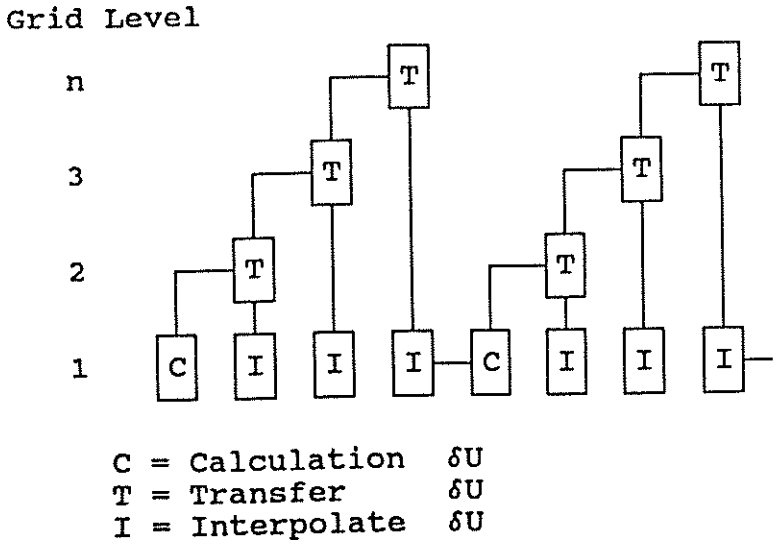


Figure 3.5 Multiple-grid saw tooth cycle for updating

3.4.3 Parallel Implementation of Euler Solver

3.4.3.1 General

The implementation of Ni's method for solving the compressible Euler equations is straightforward. The method used is based on geometric parallelism with a Host process controlling a number of Guest processes. Each Guest process is assigned a sub-region of the physical domain containing the same number of finite-volume cells. A two-dimensional problem has at most four boundaries between sub-domains. These boundaries are directly mapped onto the four external links of a transputer. Figure 3.6 illustrates the overall scheme, with the left and right halves of the figure indicating the role of the Host and Guest processes in the overall calculation.

3.4.3.2 Initialise and Setup Network

Each Guest process is told the dimensions of the transputer array and its individual identity number. This information is used to initialise the Routeing process and for determining the identity number of a Guest transputer and the identity number of its four neighbours.

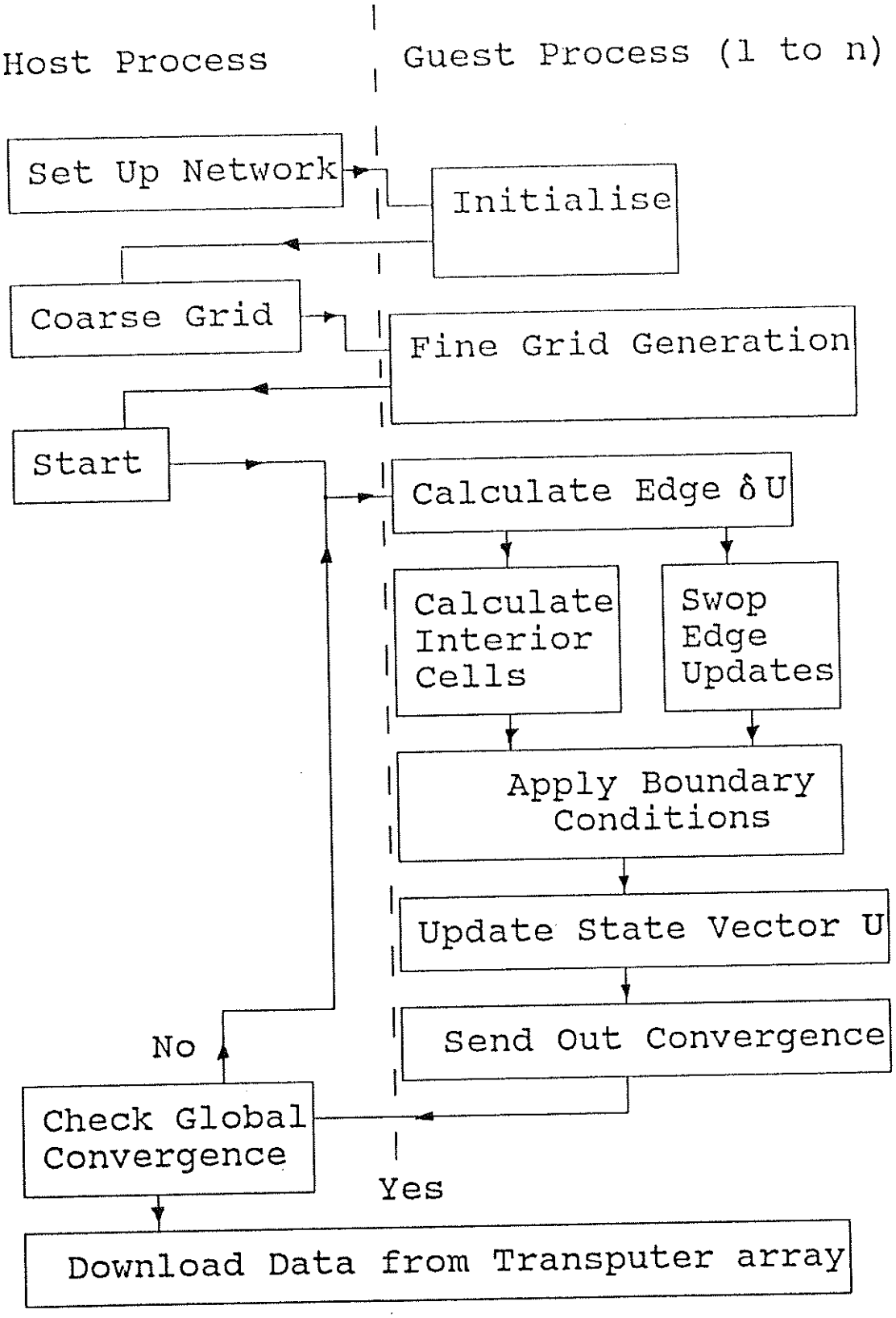


Figure 3.6 Flow chart of Euler solver code showing roles of host and guest processes.

3.4.3.3 Grid Generation

The definition of the solution domain for an explicit finite volume scheme requires a grid mesh mapped over the physical space. The generation and quality of this grid are important parameters in the efficiency and accuracy of the numerical scheme. The requirement for the same physical problem to be solved across different sizes of transputer arrays meant that the pre-processing of the solution grid was most easily carried out by making it an integral part of the Euler Solver.

A simple input file format was adopted. For each geometric problem, four external edges are defined, each edge corresponds to a single type of boundary. As a first approximation the required number of interior nodes are produced by direct interpolation between the four edges. The initial grid is refined using the elliptic grid generation method described by Steger and Sorenson[3.6]. A Successive Over Relaxation (S.O.R.) scheme is used to elliptically converge the grid. The edge values of the grid are held fixed. Further details are given in Appendix C.

The actual generation of the grid is divided into two steps. On the host process a coarse grid is elliptically refined. This grid is then subdivided into a number of regions equal to the number of transputers available. The four edges of each of these sub-regions along with the corresponding boundary types are sent to the Guest processes. On each Guest the process is repeated with the required mesh refined using the elliptic grid generator. The coordinates of the edge values on adjacent transputers are identical and are held fixed.

The format of the grid generation process allows a physical problem to be automatically matched to the number of transputers available. More complex geometries can be solved by breaking the overall physical domain into several segments each mapped onto a number of arrays within the transputer array.

3.4.3.4 Parallel Numerical method

On each Guest process the first operation at each time-step is the calculation of δ U updates for all the edge cells. Next the interior cell updates to the state vector U are calculated. In parallel with this for any edges in common with other transputers δ U values for the edge are exchanged. The communication of edge updates in parallel with the calculation of the state vector updates effectively hides the communication time.

The exchanging of information across a sub-region boundary takes place in two stages. First North-South edge information is exchanged and then East-West information. This ensures that all the information for corner nodes is sent to all four possible transputers upon which the corner node is defined.

The number of messages each transputer needs to send and receive at each time step is proportional to the number of Normal sub-region boundaries with other transputers. For each edge on the transputer which is a Normal boundary, two-dimensional arrays containing δ U updates are sent.

After message communication and calculating the interior updates, the boundary node state vectors are updated using the assigned boundary conditions. Finally, the actual interior state vectors are updated.

3.4.3.5 Global Convergence

For a distributed problem the determination of convergence requires that the maximum convergence on all transputers is known and that when all the individual maxima are below the convergence threshold the calculation is stopped. This is achieved by all the Guest transputers sending convergence information back to the Host process. The Host then confirms that the solution has converged or allows the

calculation to proceed. The global convergence test causes a delay in execution while the transputers await the instruction to proceed or terminate. The delay is dependent on the number of transputers and the average number of transputers each message has to pass through to arrive at the Host. The larger a transputer array the greater the delay associated with a global convergence test. A method to reduce this delay is to carry out a convergence check after a regular number of time steps. Effectively this convergence check synchronises all the Guest processes at regular intervals. It should be noted that a transputer and its four neighbours are locally synchronised through the exchange of information at the state-vector updating phase of the numeric calculation.

3.4.3.6 Data Retrieval

If required the steady state solution obtained on each transputer can be downloaded onto the PC terminal for subsequent processing. For large sizes of transputer arrays the amount of data storage required is at least equal to the total memory capacity of the whole transputer array.

3.4.4 Test Case Results

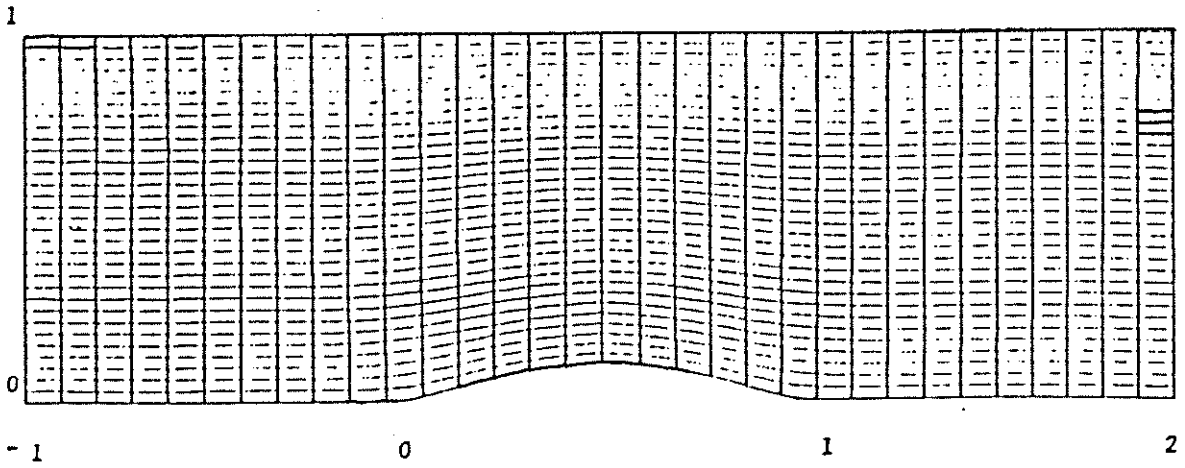


Figure 3.7 Grid for 10% thick circular-arc bump in channel

Results are given for two test geometries representative of both internal and external two-dimensional flow. These are the flow over a circular arc-bump in a channel and an aerofoil (NACA0012) at incidence. Figure 3.7 shows the overall grid used to solve a 64 by 64 cell problem flow over a circular-arc bump placed in a channel with a height equal to the chord of the bump. The maximum height of the bump is 10% of its length.

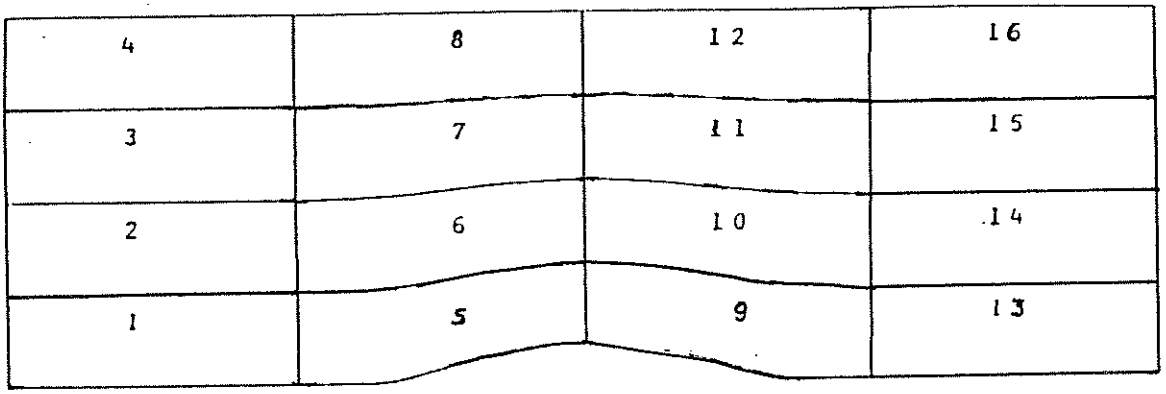


Figure 3.8 Sub-division of channel grid onto 16 transputer array

Figure 3.8 shows the sub-division of the grid corresponding to a 16 transputer array. The numbering of the sub-divisions corresponds to the layout of transputers in Figure 3.1.

Results are shown for two different inflow Mach Numbers corresponding to those used in Ni[3.2]. Figure 3.9 is for a subsonic inflow with Mach number equal to 0.5 and shows the velocity on the upper and lower edges of the channel.

A transonic Mach number of 0.675 was used to generate Figure 3.10.

A NACA00 2 aerofoil section was also tested. The grid generated for the NACA 0012 aerofoil section is shown in Figure 3.11 for a 64 by 64 grid. The corresponding sub-division of the grid onto sixteen transputers is given in Figure 3.12.

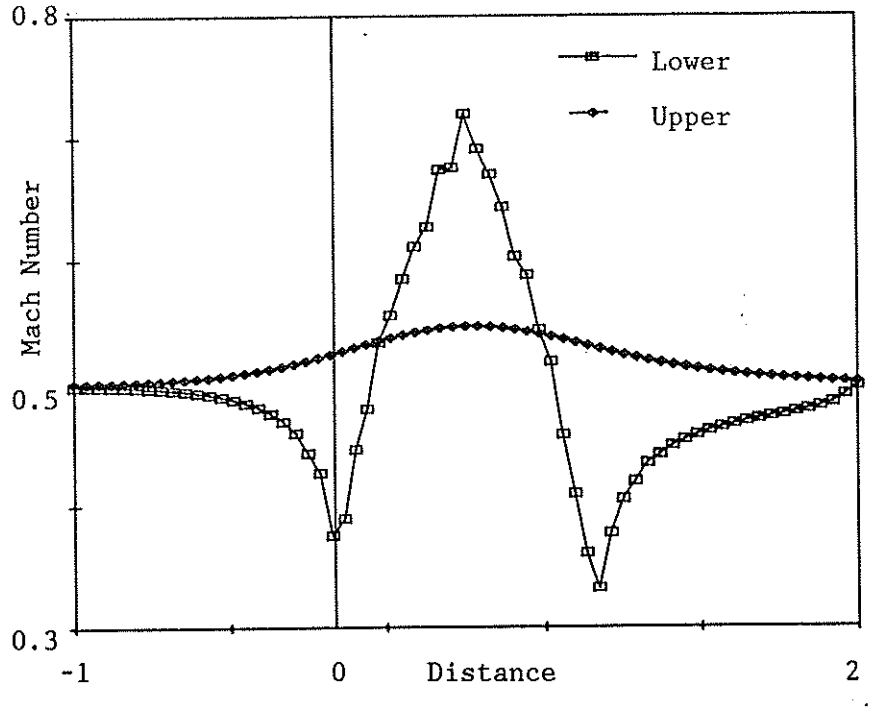


Figure 3.9 Subsonic flow over 10% bump in channel at Mach number of 0.5

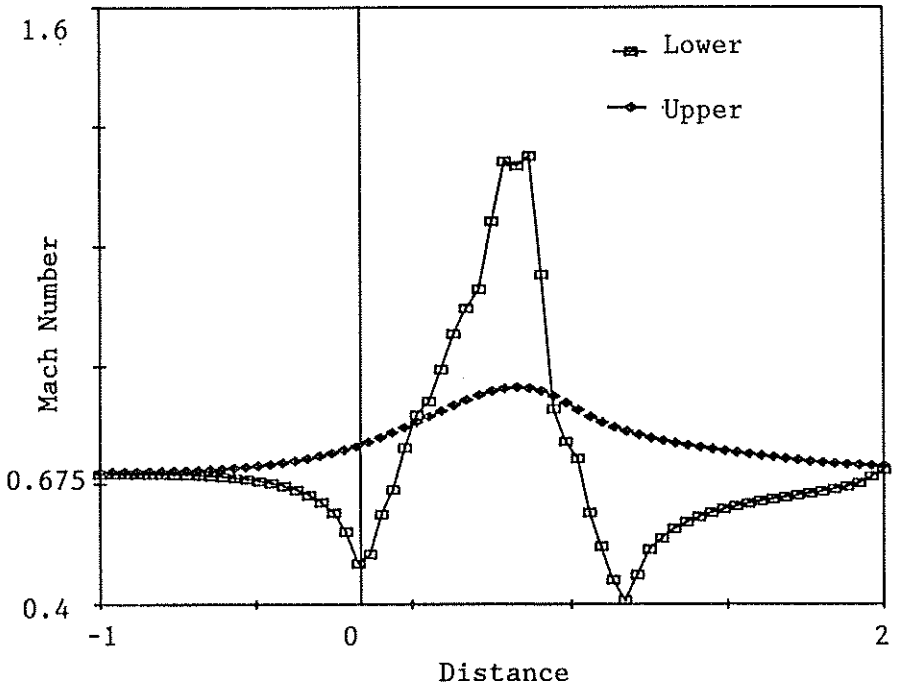


Figure 3.10 Transonic flow over 10% bump in channel at Mach number of 0.675

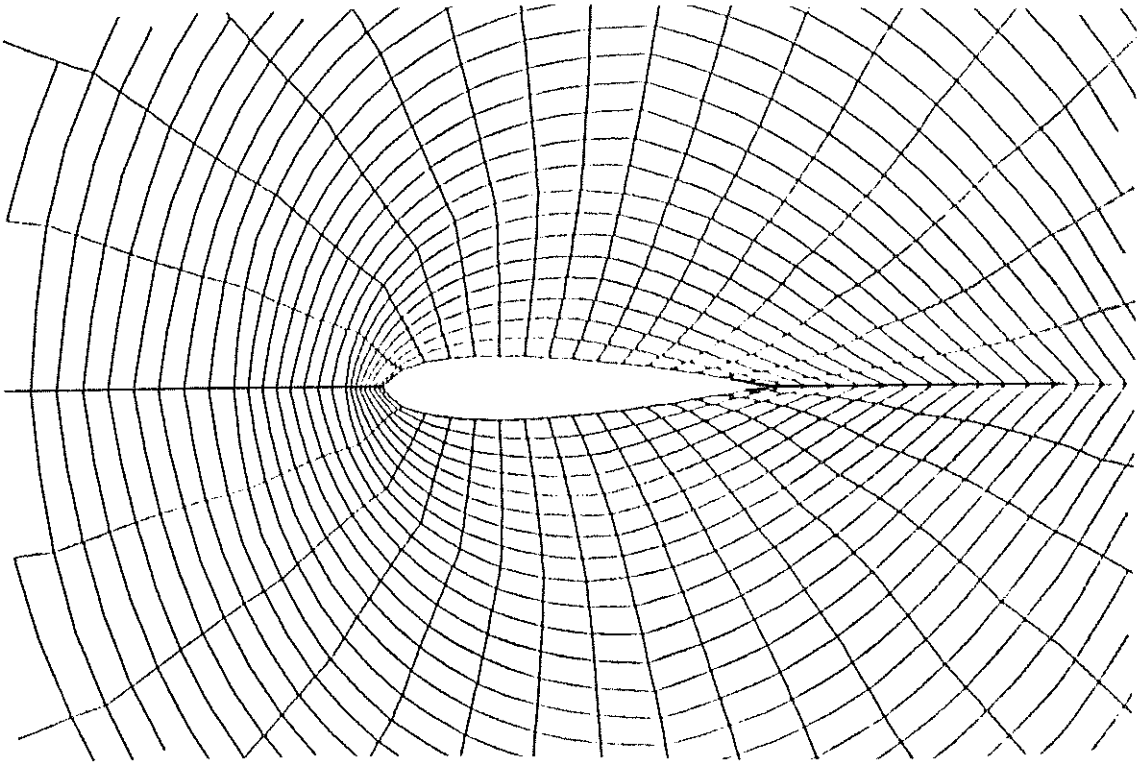


Figure 3.11 Grid for NACA0012 aerofoil with mesh size of 64 by 64

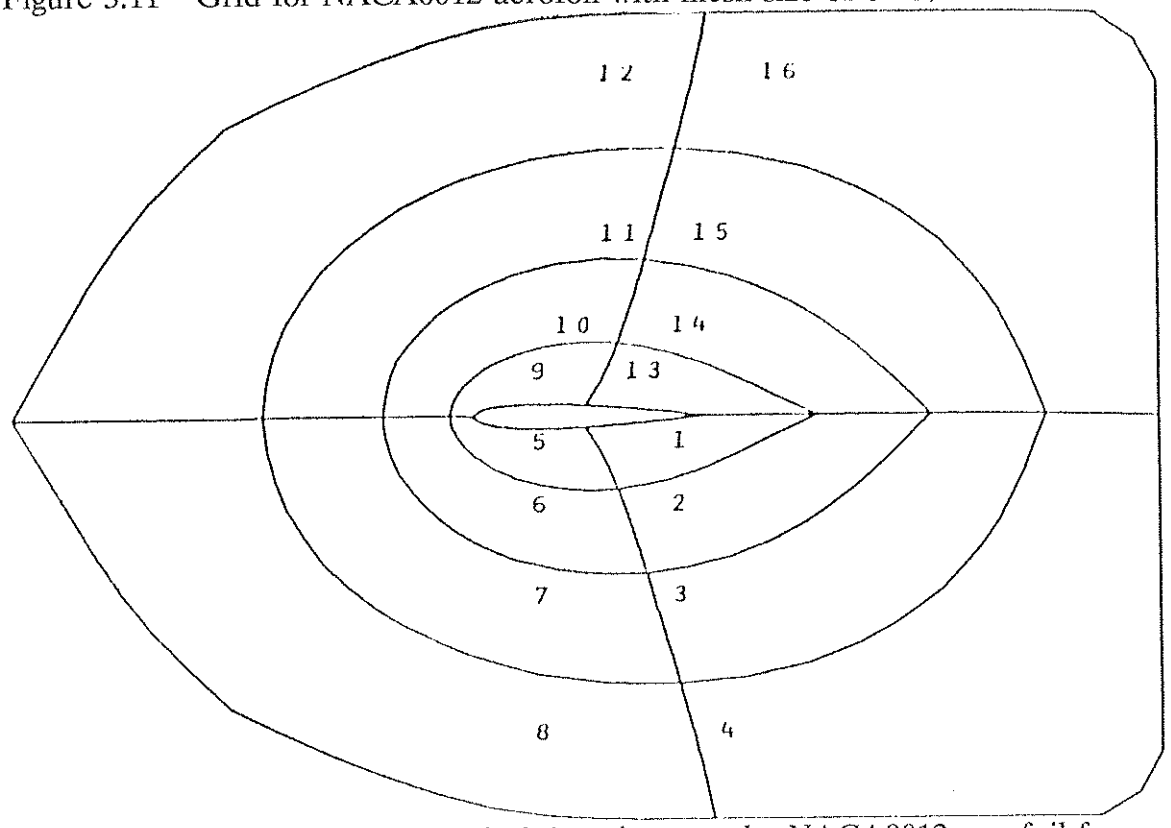


Figure 3.12 Sub-division of physical domain around a NACA0012 aerofoil for an array of 16 transputers

The resultant grid is an 'O' grid with two sides of the transputer array joined on a line downstream from the trailing-edge. Iso-mach contours are shown for a test case given by Jameson[3.7] with a freestream Mach Number of 0.5 and angle of attack of 3° in Figure 3.13.

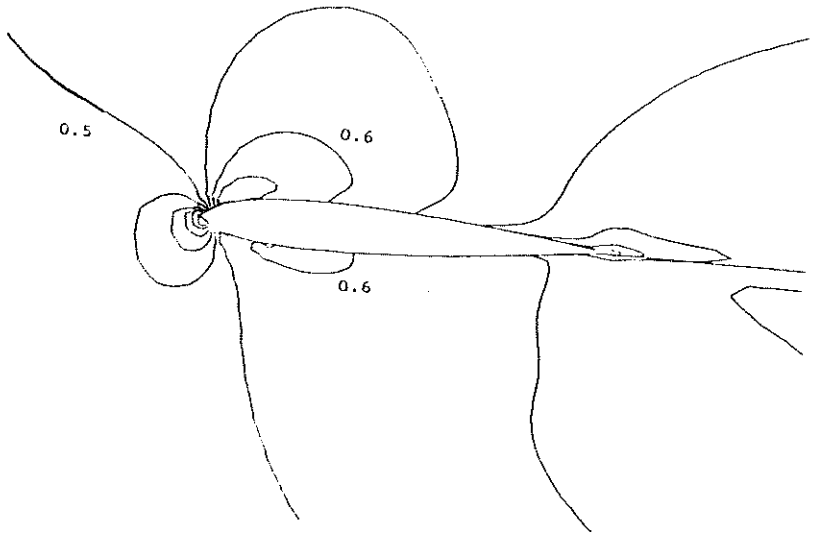


Figure 3.13 Iso-mach contours around a NACA0012 section at incidence of 3° and at 0.5 Mach No.

3.4.5 Code Performance

Results are presented for the performance of the Euler Solver and Harness on square arrays of up to 16 transputers. A standard test case for transonic flow at a Mach number of 0.675 over the 10% thick circular-arc bump was used for all the performance measurements. No Multiple-Grid acceleration was used. The data presented is based on the average time to perform one iteration with the same number of finite volume cells on all transputers in the array.

In addition to the average time per iteration, the average amount of time each transputer spent communicating convergence information to the Host transputer was measured. The nearest neighbour communication takes place in parallel with the calculation and cannot be measured. The nearest neighbour communication time on each Guest is equal to the difference in time taken to carry out an identical calculation with and without communication across sub-region boundaries.

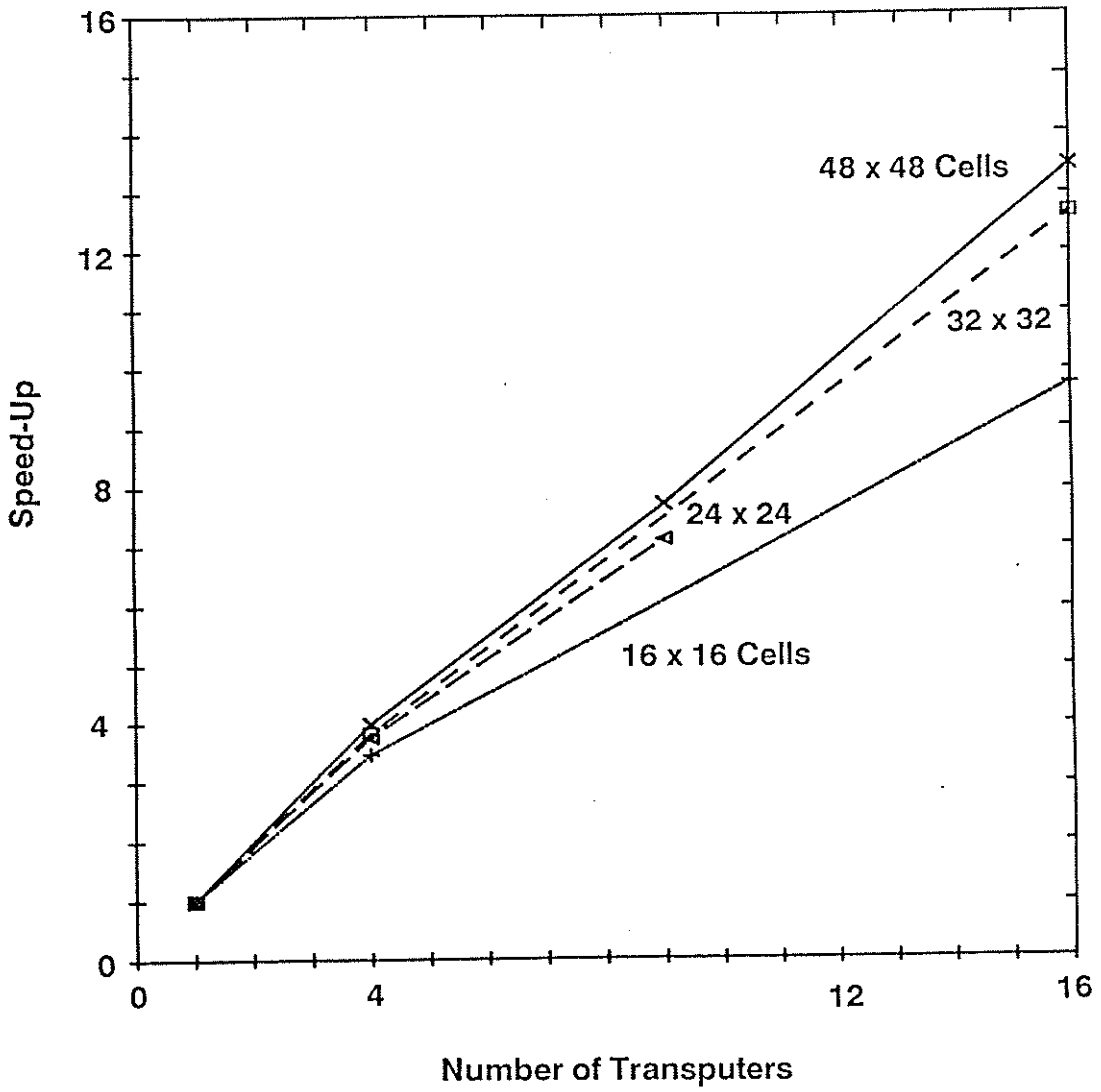


Figure 3.14 The speed-up Su of Euler Solver against number of transputers for a given overall number of finite volume cells

Figure 3.14 shows the Speed-Up Su obtained when an identical problem was run on different numbers of transputers. It can be seen that as the overall problem size is increased (more finite-volume cells per transputer) a greater Speed-Up was obtained. With 16 transputers a Speed-Up of 13.4 was obtained for a 2304 cell problem compared with 9.7 for a 256 cell problem.

In Figure 3.15 the Code Efficiency η_c is plotted for different numbers of transputers and amounts of finite volume cells per transputer. As expected, the smaller the number of cells allocated to an individual transputer the lower the Code efficiency

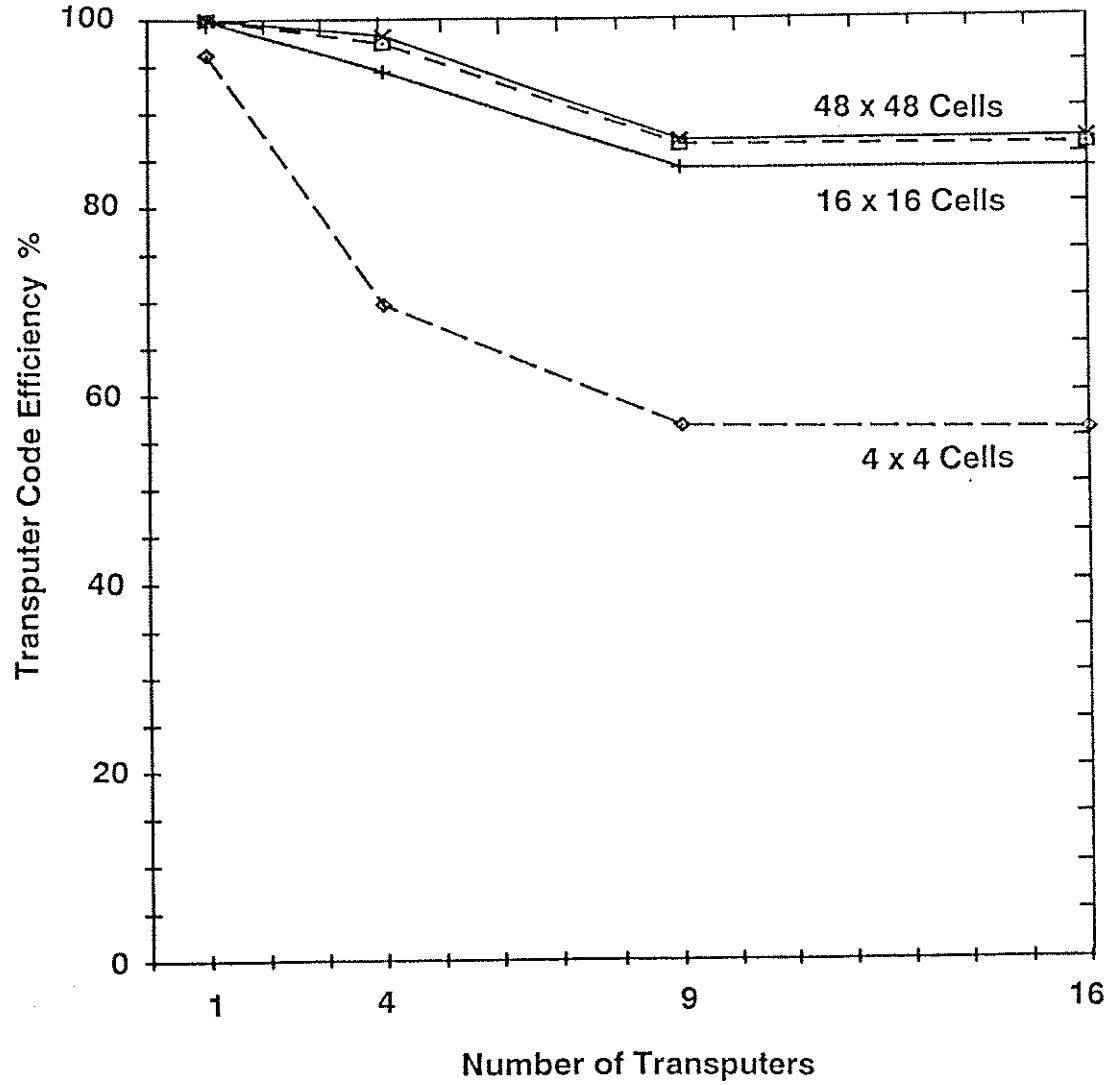


Figure 3.15 Code efficiency against number of transputers for different numbers of finite volume cells per transputer

as the transputer spends a proportionally greater amount of time communicating. For only one transputer the Code efficiency is close to a 100% as the only communication which occurs is the passing of convergence information to the Host. As the transputer array size increases, the code efficiency drops as the transputers have to communicate nearest neighbour information. It is interesting to note that after 9 transputers there is no appreciable decrease in Code Efficiency with 16 transputers. The amount of time to carry out a global convergence check is small compared to that for nearest neighbour exchange. Therefore, the individual Code efficiency of a transputer array will be determined by the slowest Guest process. In the case of 4 transputers, all 4 have to

exchange information across two edges, however, for 9 and 16 transputer arrays the slowest processes are the interior transputers which have to exchange information across all four edges. The Code-Efficiency and speed-up will be determined by the time-delay associated with communication across all four edges of a Guest process. This explains why there is no appreciable decrease in Code Efficiency between 9 and 16 transputers. This allows predictions for the Speed-Up of larger transputer arrays to be made as the individual Code efficiency is independent of the transputer array size.

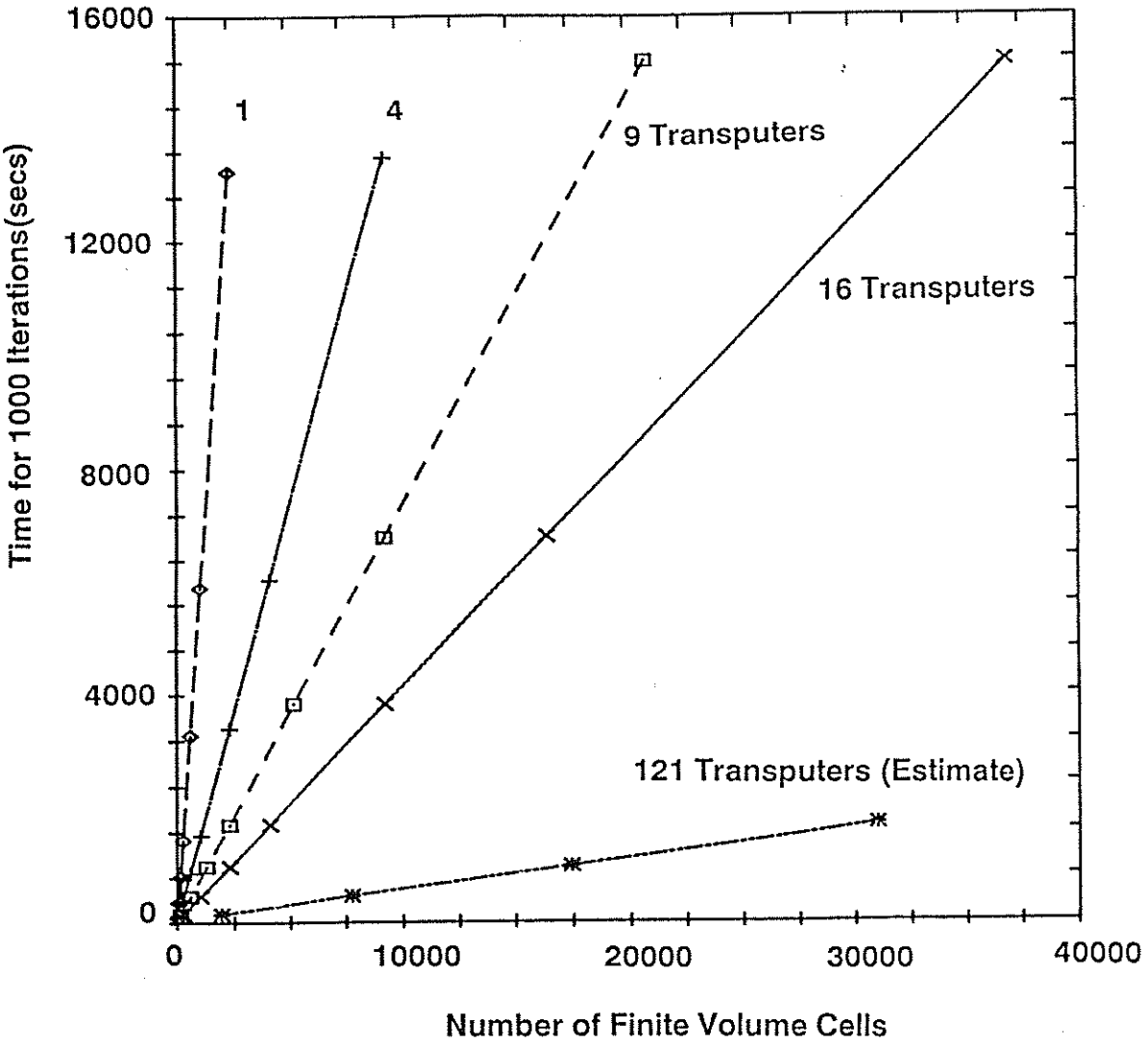


Figure 3.16 Performance chart of time to carry out 1000 iterations against the overall number of finite volume cells for different sizes of transputer array

A performance chart for a given number of finite-volume cells and the physical time to carry out a thousand time-steps on a specific number of transputers is shown in Figure 3.16. An estimated plot for a 121 transputer array is included. This is based on individual transputers in a 16 and 121 transputer array having the same Code Efficiency. For each size of transputer array there is a linear relationship between the number of cells and the time to carry out a thousand iterations. Increasing the transputer array size increases the speed of solution but each individual transputer's code efficiency decreases. The minimum number of cells assigned to an individual transputer is 16 so for a given array size there will be a minimum problem size. Correspondingly, the amount of memory assigned to a transputer determines the maximum number of cells which can be allocated to a transputer. For a given computational load per cell the total number of finite-volume cells and the amount of memory assigned to each individual transputer will determine the optimum array size.

3.4.6 Conclusion

The effectiveness of the parallel implementation of the Euler solver was measured. It was found that the speed-up in overall time to convergence obtained by increasing the number of transputers was dependent on the physical size of the problem. The larger the number of transputers, the more efficiently each transputer was used. With a small domain of 16 by 16 finite-volume cells the code ran 9.7 times faster on 16 transputers compared to a single transputer. For a 48 by 48 cell problem a speed-up of 13.4 was obtained with 16 transputers. For transputer arrays with more than 4 transputers the time to carry out one time step was inversely proportional to the number of transputers in the array.

3.5 Modifications to Harness Design

Following the successful implementation of the Euler solver, using the communications harness, possible improvements to the design of the harness were considered. The aims of the improvements were to:

1) improve speed of data transfer and
2) reduce the memory requirements of the communications harness code.

Several alternative strategies were considered to modify the data flow through the harness. However, the first step of analysis involved an investigation of the breakdown of time spent by the individual processes within the harness. It was soon discovered that up to 90% of the total time was spent in the Encode and Decode processes in translating REAL/INT numbers into strings of bytes. The original reason for using the Harness protocol as a simple string of bytes was to simplify the data flow structure while allowing easy debugging during the development of harness procedures.

To remove the inefficient translation process it was decided to define a new protocol which would combine the User and Harness protocol and would use a single tagged protocol for passing all types of User messages through the Harness process and across the transputer networks. The encode and decode processes would now consist of copying the relevant data into or out of the User protocol tags and into the Harness tag. These are processes with a minimal time requirement.

A major benefit of this approach was a significant reduction in the data storage requirement of the Harness process. An individual Real32 number could occupy a string length of up to 20 bytes which for large arrays of numbers required large amounts of memory. The new protocol reduced the overall harness memory requirement by approximately 50%.

Figure 3.17 plots the ratio of the original harness time T_{orig} to that of the modified harness T_{mod} against the number of real numbers in a message. The separate lines show the effect of increasing the number of messages passed. These timings were measured using the TestBed program described in the next section. From the plot it can be seen that the greater the length of message the greater the improvement speed-up obtained by using the modified harness as the inefficiency of the encode/decode process has been removed. In particular, for a large number of messages the speed up

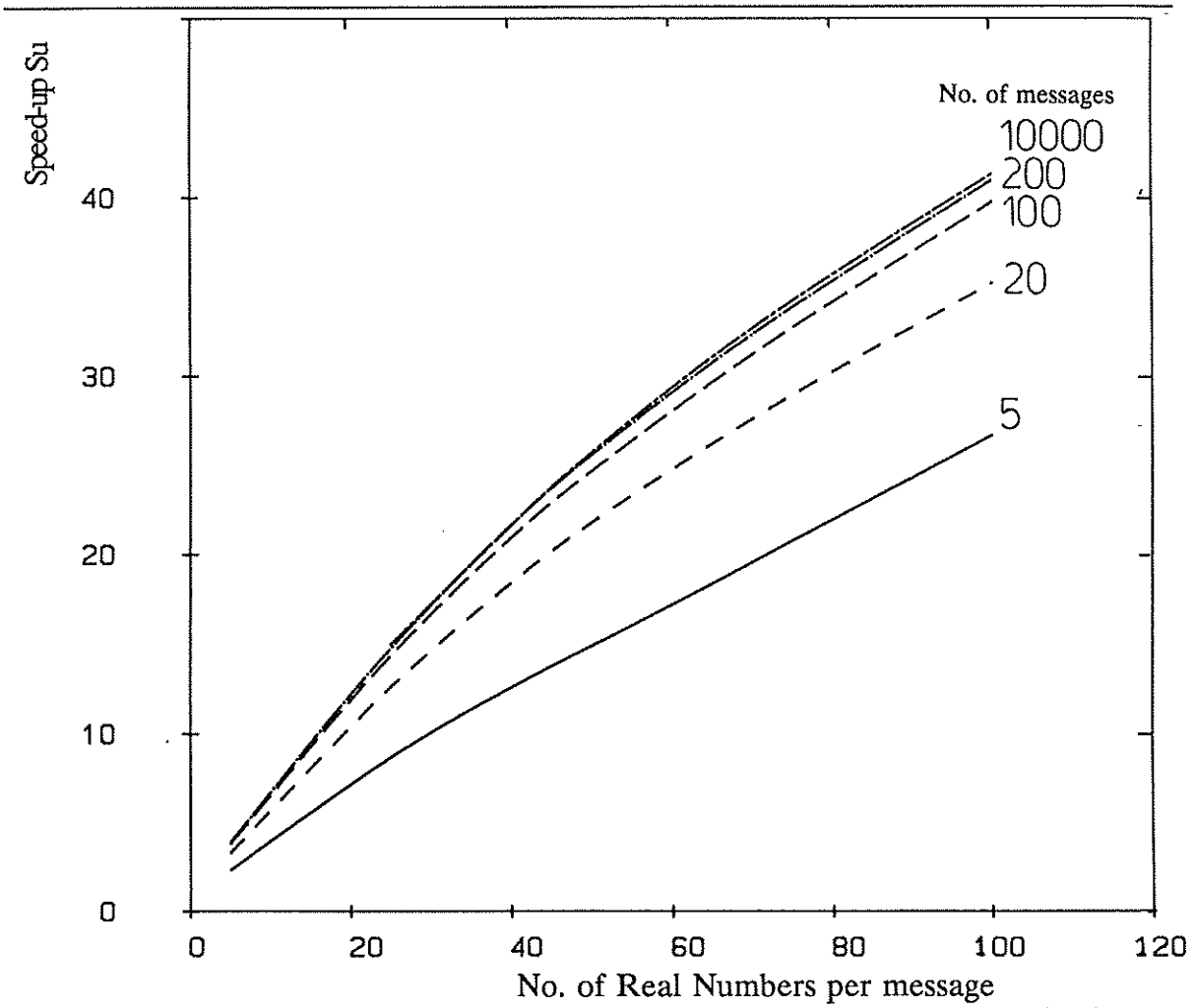


Figure 3.17 Comparison of peformance of original and modified communications harness

was up to 40 times, i.e. the modified harness design took $1/40^{th}$ of the time to pass a given number of messages. Also, the larger the number of messages passed during the test the greater the speed-up. However, the change with increasing the number of messages passed was small compared to that obtained with increasing the length of message.

The analysis of the harness data flow structure showed that it was unlikely that further large savings of time could be made without significant effort and with guaranteed improvements. An interesting phenomenon of parallel processing is that more sophisticated strategies may not actually work any better and it is not possible to predict improvements. In the author's experience, the simpler the approach the better. Therefore, the existing harness data flow structure and methodology was retained.

3.6 Performance of Final Harness Design

3.6.1 Introduction

The quantification of the performance of the Harness process requires the measurement of the time taken to pass messages between transputers across a network. The time taken for a message to cross a number of links between transputers will depend on: the length of the message, for example the number of real numbers in the message; how much other message traffic is also passing across the network; and the additional amount of numeric processing carried out by processes running in parallel to the Harness process.

As each transputer's internal clock is independent it is not possible to measure the absolute time for a message to pass between two transputers by comparing the difference in time on their two clocks. However, if the message is returned to the original transputer the time for the total journey out and back can be recorded and the average journey time between the two transputers obtained.

3.6.2 Message Passing Testbed

A program (TestBed) was written to measure the performance of the harness process. This allowed a series of message timing experiments to be carried out. The structure of the program is identical to that used for numeric applications. That is:

```
PRI PAR
  Harness(Lk1i, ...,Lk4o, toguest, fromguest)
  --
  TestBed(toguest, fromguest)
```

The TestBed process contains a number of different tests for evaluating the performance of the Harness. As a test message is sent out the local transputer's low-

priority clock is measured and the time is included with the message. When the message returns the time is recorded again and the average time for the journey to the destination is half that of the difference between the start and finish times. This time does include the time taken for the message to be received and returned by the destination TestBed process although for solitary messages (no other messages on the network) this should be small. TestBed consists of three parallel processes:

```
PAR
  ... generate message(s)
  --
  ... send message
  --
  ... receive message
```

Messages arriving in TestBed are tested to see if they originated from the particular transputer or if they are to be returned to their origin. If they are returned messages their elapsed journey time is added to the running total time to be averaged when all messages have been sent and returned.

A single message type of a one dimensional array of real numbers was used. For comparison with other harness performance tests each real number occupies 4 bytes and the total message length in bytes can thereby be found. The number of real numbers per message for these tests could be varied between 1 and 110.

These tests were carried out on the Ship Science Transputer System and allowed message timings over up to 4 links. The tests carried out were:

1) Link Tests: In these tests solitary messages were sent from the host transputer(0) to the other four transputers (1..4) only along the up and down links. This gave a maximum number of links for the message to pass across of 4. As only one message is crossing the network at a time and the guest transputers are simply waiting to return

a message this test should measure the maximum message speed of the Harness. By varying the length of message and the number of links a performance relationship can be obtained and compared with other harness codes. The times obtained were averaged over 10,000 separate messages.

2) Message Density: All four guest transputers broadcast fixed length messages to either their nearest neighbours or to any of the transputers in the network. Each transputer sends a certain number of messages, the actual processor number being randomly assigned. By altering the total number of messages each transputer sends out the performance of the network with differing degrees can be assessed.

3) Processor Delay: This is the same as for the message density test. However, in addition there is a fixed time length loop delay before sending out each message. This acts to simulate a numeric processing load in TestBed.

3.6.3 Performance Results

Table 3.1 gives the average time in T800 clock ticks (1 tick = 64μ seconds) to send a message of given length in bytes (b) over a variable number of links (L). using this information a relationship was found for the time to pass the message in terms of L, b and their product L.b. This is given, where t is in milliseconds as:

$$t = 525 + 320.L + 0.86.b + 1.4.L.b \qquad [3.14]$$

The message length b in bytes is related to the length of the one-dimensional array of 32bit real numbers (Nr) as follows

$$b = 53 + 4.Nr \qquad [3.15]$$

It should be noted that this relationship includes the time to pass the message in and out of the TestBed process and also the time taken to interrogate the transputer's internal clock. However, the relationship gives a good basis for estimating the required communication time of numerous processes. For instance, this information is necessary

Table 3.1 Average time to pass 10000 messages across a number of transputer links

Message Length		Time to pass message (T800 ticks)			
bytes(b)	No. Real32	1 Link	2 Link	3 Link	4 Link
53	1	14.63	22.17	25.94	33.48
93	10	15.91	24.36	28.58	37.06
153	25	18.03	27.98	32.98	42.98
253	50	21.56	34.05	40.33	52.83
353	75	25.09	40.10	47.64	62.70
493	110	30.04	48.55	57.89	76.50

when estimating the performance of implicit algorithms where the volume of communications traffic per transputer increases with the size of the transputer array. This is opposed to that for an explicit algorithm, as seen with the Euler solver, whose volume of communication traffic per transputer remained constant.

Figure 3.18 plots to a base of message length in real numbers the time taken to pass messages for differing message densities. It can be seen that beyond a total number of messages of about 50 the system saturates and that increasing the number of messages does not significantly increase the message time. The ratio of saturated message time to that for a single message is about 3.5 and this is not strongly dependent on message length. This degradation in performance will only be important when a process is awaiting information with no numeric processing to carry out. Normally the extra delay will be hidden as there is information available for processing. A similar effect was found when a random length delay was introduced between messages. The time to send a message increased to a limit which was independent both of the message length and the number of messages. It is thought that this limit was determined by the hardware switching between parallel processes on the transputer itself.

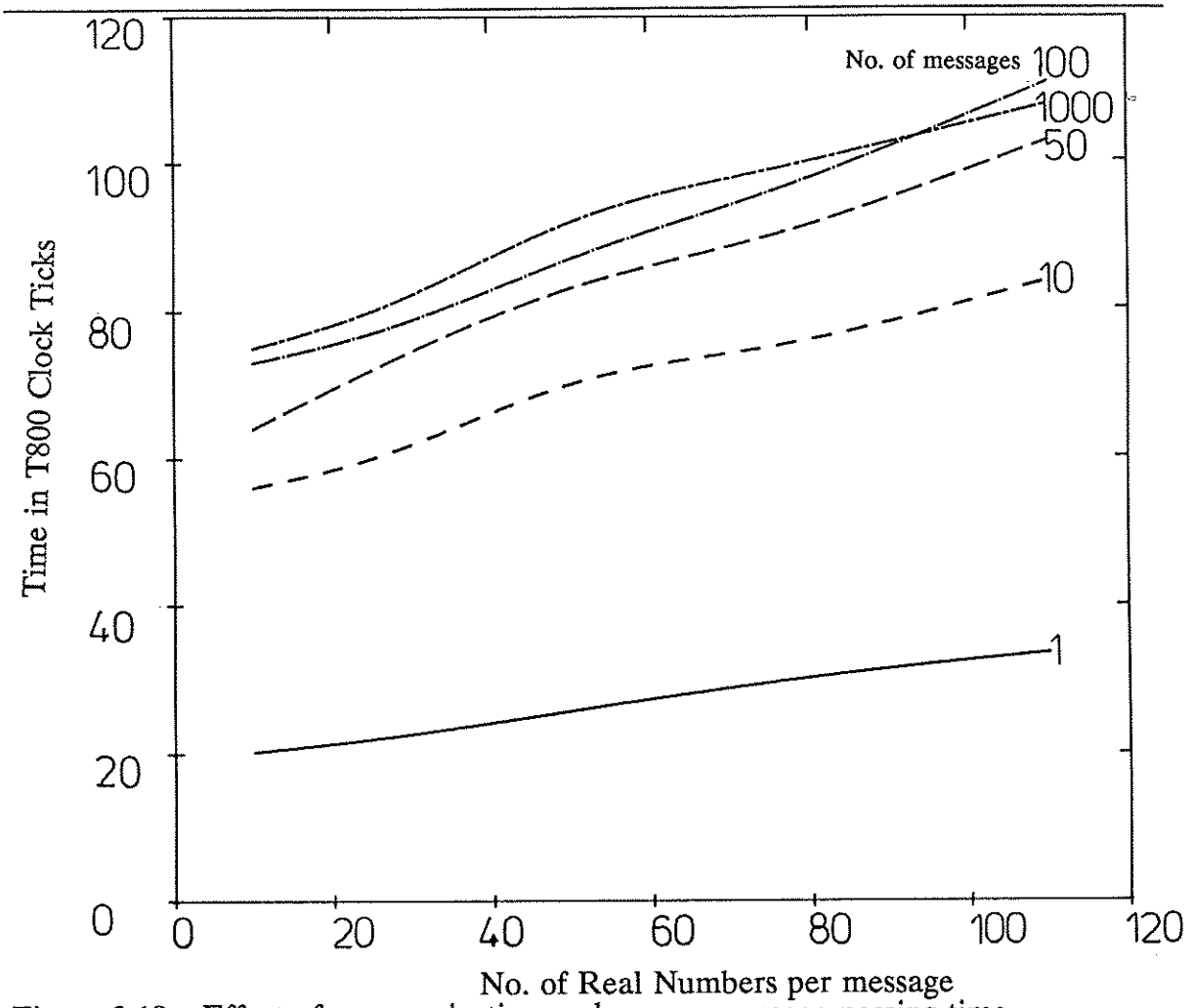


Figure 3.18 Effect of communications volume on message passing time

3.6.4 Discussion

Hey[3.8] gives figures for the harness (named TINY) developed by Clark as part of the Edinburgh transputer initiative. This gives a relationship for the message passing time in micro-seconds over varying numbers of transputer links and message size of:

$$T = 32.7 + 23.4L + 0.2b + 0.7Lb \qquad [3.16]$$

Figure 3.19 compares the above relationship with that obtained for the Harness process. The vertical axis is the ratio of the time taken by TINY and that for the C.F.D. Harness to pass a message. The horizontal axis is the message length in terms

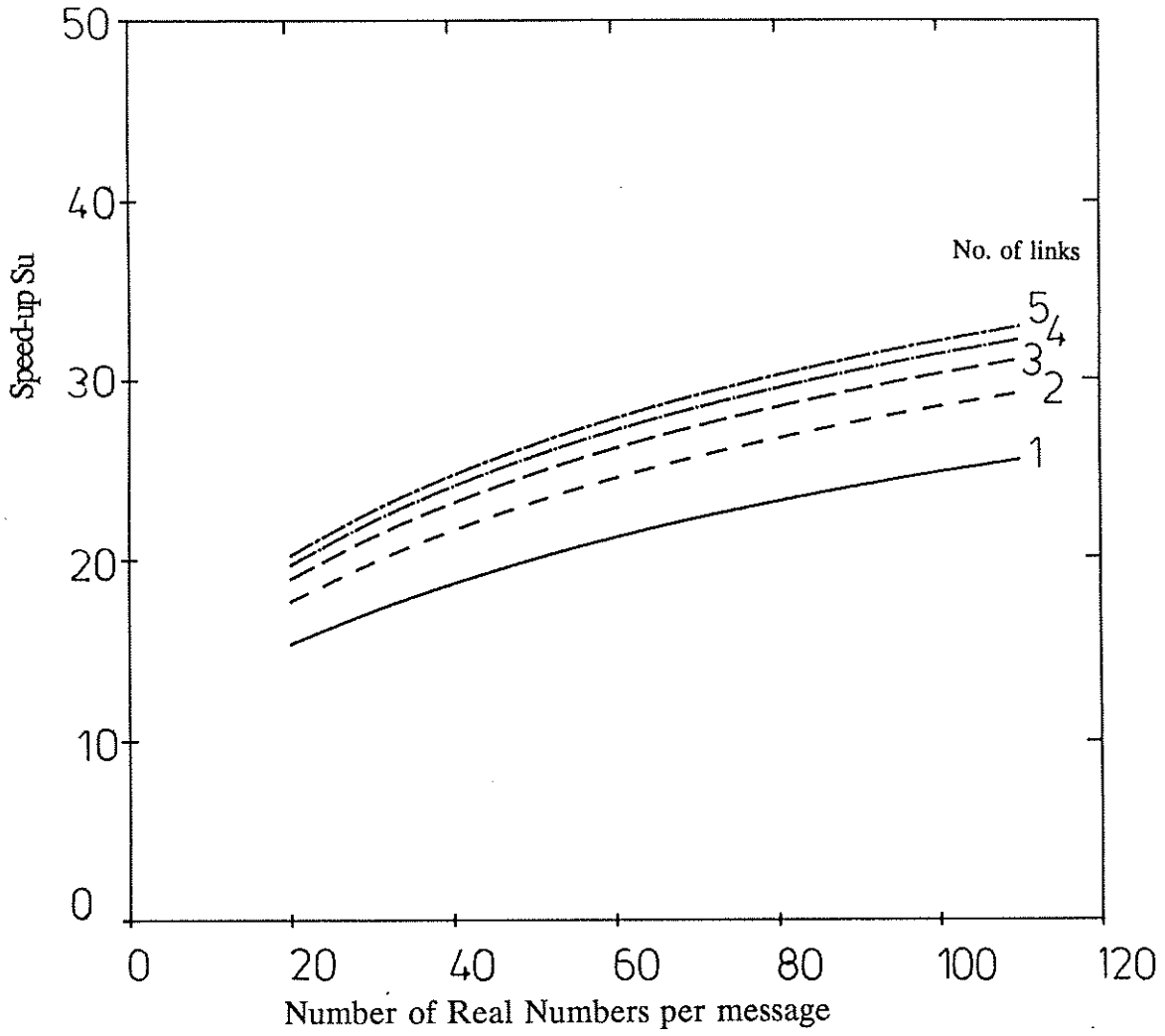


Figure 3.19 Performance comparison between the developed CFD harness and TINY

of real numbers. Different lines show the effect of increasing the number of transputer links. From the figure it can be seen that the CFD harness is between a 15th and 30th of the speed of the TINY software. However, as noted earlier, the figures for the CFD harness include the time for the data to be processed by the testbed process and it is not clear whether the figures given for TINY include such things as transcribing between protocols. Overall the CFD harness developed was considered adequate for use with the moderate sized arrays of transputers available for use in this work and that it would be suitable for use with the implicit algorithms necessary for carrying out a lifting-surface panel method. The development of these algorithms is described in

Chapter 5. The timing information for the CFD harness can be used to estimate the performance of these implicit algorithms as the message passing experiments were carried out under realistic conditions.

3.7 Conclusion

The main part of the work described in this chapter has been the development of a general harness process to control communications between transputers. Using Harness, numerical algorithms can be made to run on variable sized array of transputers without the need to write communications software for each application.

It was found that the implementation of a representative explicit numerical fluid dynamic algorithm on a variable-sized array of transputers was a straightforward exercise.

The communications harness, written to carry out the necessary inter-transputer communication, greatly eased the development of the numerical algorithm. The development of Harness also allows other computational fluid dynamic algorithms to be easily implemented at a future date.

The explicit two-dimensional Euler solver provided a good method of testing the performance of an array of transputers. It was found that for transputer array sizes greater than four the individual Code efficiency η was independent of the transputer array size. This implies that the speed of solution should be inversely proportional to the number of transputers which is as expected for geometric parallelism of an explicit algorithm.

Following the implementation of the Euler solver the original harness design was modified. This gave a large improvement in harness performance and a 50% reduction in memory requirement. The speed of communication tests carried out on the harness allow the performance of implicit algorithms using the harness process to be estimated

for a given size of transputer array.

The modified harness is considered to have an acceptable performance which is able to cope with the demands of the implicit algorithms required to carry out a lifting surface analysis. The theoretical basis of the lifting surface panel method is given in the next chapter and its implementation across an array of transputers in Chapter 5.

CHAPTER FOUR
LIFTING SURFACE PANEL METHOD FOR MODELLING SHIP RUDDERS AND PROPELLERS

4.1 Introduction

This chapter describes how the interaction between a ship rudder and propeller is to be based on separate lifting surface panel solutions of the flow around a rudder and propeller geometry. The two solutions are coupled through the iterative modification of their respective inflow velocity fields. This technique, referred to as the Interaction Velocity Field (IVF) method is based on work carried out on a frequency-domain panel method for modelling aircraft propeller-wing interaction.

The theoretical basis of the lifting surface panel method is given and uses a perturbation potential formulation. An explicit condition to enforce zero pressure loading at the trailing edge is used. An important part of the work has been the use of a geometry definition scheme using parametric cubic splines. Also described is a method by which the shape of the trailing wake sheet can be iteratively adapted.

A final section describes the verification of the panel method against the analytic solution for flow over an ellipsoid, flow over a circular plan form wing, and flow over a rectangular NACA0012 section wing.

4.2 Review

Only a limited amount of theoretical work has been carried out with regard to the modelling of a ship rudder and propeller (see Chapter 1). However, the similar problem on aircraft of wing mounted propellers and the resulting interaction between the wing and propeller has received considerable attention.

In a recent paper by Cho & Williams[4.1] the propeller-wing interaction was

investigated using a frequency domain panel method. A linearised compressible potential theory was used to determine the unsteady aerodynamic coupling between the wing and propeller. An iterative approach was used whereby the propeller and wing are solved in isolation and the effect of the interaction is obtained using a Fourier Transform of their respective induced velocity fields. The lifting-surface method used gave good agreement with measured mean loads on a wing/propeller system. A principal conclusion was that useful performance predictions could be made where circumferentially averaged (isolated) propwash is imposed on the wing and where a circumferentially averaged (isolated) wingwash is imposed on the propeller to get the modified steady blade/rudder loading.

Previously Kleinstein [4.2] had investigated the effect of a propeller on the aerodynamic characteristics of a high aspect-ratio wing. Using a lifting-line approach with the propeller stream represented by an equivalent circular jet of uniform velocity, it was found that a better approximation to propeller-wing interaction was obtained than that by Koening [4.3]. Koening's classical theory represents the propeller as a uniform two-dimensional velocity field of height equal to the propeller diameter. A drawback to the work was the lack of experimental studies to verify the results.

Kroo[4.4] studied the problem of minimising induced loss for the integrated system of propeller and wings. In a simplified model using a fourier series wing lift distribution the stagger theorem of Munk [4.5] was used. In its generalised form this states that "*the net force in the stream-wise direction is independent of the stream-wise position of surfaces with a given circulation distribution*". This implies that the wing optimum circulation distribution (minimum drag) can be computed with the propeller far upstream and therefore removing the need to calculate the effect of the rudder on the propeller and that of the propeller's bound vorticity of the wing. This simple analysis showed that the wing's optimal lift distribution significantly differed from that of the elliptic load distribution of an isolated wing. The wing acts to restore the swirl energy otherwise lost from the propeller's wake. Again, no experimental comparison with theory was made.

An experimental investigation and computational analysis was carried out by Witkowski et al [4.6] at Purdue University. These tests were carried out with a configuration for a tractor mounted 12" diameter propeller upstream of a rectangular wing with geometric aspect ratio of (33/4). In the computer analysis two methods were used. A semi-empirical method involved the superposition of experimentally correlated propeller wake velocity onto a uniform flow across a wing modelled using a vortex lattice method. This gave a fast means of achieving results which predicted ΔC_D to between 10% and 30%. The other method used a vortex-lattice to simultaneously model the wing and propeller. In this, the propeller is assumed to have a wake on a rigid helix of constant diameter and the wing wake is assumed flat and in the stream-wise direction. The calculation was repeated for a range of propeller blade phase angles and the quasi-steady forces obtained by averaging over a complete revolution. This method gave a better comparison for ΔC_D of between 6-12%.

Although the work of Witkowski [4.6] and also that of Lorber [4.7] in looking at helicopter rotor-fuselage interaction have been moderately successful in modelling the complete interaction geometry, a fundamental problem remains in the specification of the propeller wake geometry. At best, only an educated guess is possible in stating the wake geometry, and vortex-lattice methods can be sensitive to the location of collocation points. As most experimental results are of time-averaged quantities it is difficult to justify the use of combined geometry models where a non-physical description of the wake structure has to be used to obtain results. The approach used by Cho and Williams [4.1] would appear to be the most promising with the separation of the problem into the modelling of a propeller with imposed velocity field and a rudder with imposed velocity field. This allows the considerable expertise of modelling isolated rudders and propellers to be fully used and the modelling of the interaction can be concentrated on developing adapted velocity inflow fields.

No mention has been made so far of more sophisticated modelling of the flow using a Euler or Navier-Stokes formulation. This is due to the large computational requirements of such methods combined with the difficulties of modelling a propeller

helical wake. At the present time the methods for modelling steady-state isolated propeller all require massive computational resources. Therefore, the available methods for modelling the propeller are restricted to some form of lifting surface, lifting-line, or blade-element momentum approach.

The balancing of the need for physical realism with the complexity of numerical formulation and hence computational requirement is dependent on the final use of the computed solution. To develop a parametric relationship for manoeuvring studies and design purposes, as opposed to studies of noise and vibration, only quasi-steady total forces and surface pressure information are required for the rudder and propeller. This information may be derived from a lifting-surface panel method. This is a potential formulation which assumes the flow is inviscid, incompressible and irrotational. Panel methods for estimating forces were first used by Hess and Smith [4.8] and have been the subject of considerable development.

The separation of the modelling of the rudder and propeller allows the most appropriate method for modelling each to be chosen. Molland [4.9] obtained a good comparison for a lifting-line model of free-stream semi-balanced skeg rudders. However, a propeller inflow results in considerable span-wise cross-flow which cannot be adequately modelled using a lifting-line except thorough considerable empiricism. It is also not possible to directly determine the local chordwise position of centre of pressure for a given span position. This is required for an understanding of the rudder stock torque and for sizing the rudder control gear. Therefore, a surface panel method was chosen to model the rudder.

A lifting-surface approach to modelling a propeller has been successfully developed at M.I.T. by Kerwin et al [4.10,11,12]. A variety of formulations and wake models have been used. On the other-hand blade-element momentum theory and Goldstein corrections, (eg. Eckhart and Morgan[4.13]),can be used to calculate the propeller race information necessary for an inflow field to the rudder. However, this approach uses considerable empiricism and it is difficult to take account of the

upstream influence of the rudder on the propellers performance. In this work it is proposed to use the lifting surface formulation for modelling both the rudder and the propeller. This approach will give the necessary information at a reasonable computational cost for calculating the interaction effects and the performance characteristics of a ship rudder and propeller.

In respect of the secondary aim of this work of assessing the performance of transputer arrays in solving fluid dynamic problems an implicit method for the solution of the inviscid, incompressible, irrotational steady-state scheme such as a lifting-surface panel method provides a rigorous test.

4.3 Theory

Based on the review of literature, a reasonable compromise between computational effort and physical accuracy in modelling the flow interaction was concluded to be the use of a lifting surface panel method. The same panel method would model in isolation the individual rudder and propeller. The interaction between the two lifting surfaces is accounted for by the modification of their respective inflow velocity fields.

In a lifting surface panel formulation the approximation of the full Navier-Stokes equation assumes that the flow is inviscid, incompressible and irrotational and satisfies Laplace's potential equation:

$$\nabla^2\phi = 0 \qquad [4.1]$$

A detailed description of the method and a review of its historical development is given by Hess[4.14]. Lamb[4.15] showed that a quantity satisfying Laplace's equation can be written as an integral over the bounding surface S of a source distribution per

unit area σ and a normal dipole distribution per unit area μ distributed over the S. If $\mathbf{v}$ represents the disturbance velocity field due to the bounding surface (or body) and is defined as the difference between the local velocity at a point and that due to the free-stream velocity then:

$$\underline{v} = \nabla\phi \qquad [4.2]$$

where ϕ is defined as the disturbance potential. This can be expressed in terms of a surface integral as:

$$4\pi\phi = \int\int_{S_B} [\frac{1}{r}\sigma + \frac{\partial}{\partial n}(\frac{1}{r})\mu]dS + \int\int_{S_W} \frac{\partial}{\partial n}(\frac{1}{r})\mu\, dS \qquad [4.3]$$

where S_B is the surface of the body and S_W a trailing wake sheet, as illustrated in Figure 4.1 for a two-dimensional flow. In the expression r is the distance from the point for which the potential is being determined to the integration point on the surface and $\partial/\partial n$ is a partial derivative in the direction normal to the local surface. A dipole distribution is used to represent the wake sheet. Hess [4.16] showed this can be directly related to the vorticity distribution used in vortex lattice methods (VLM).

The conditions imposed on the disturbance potential are that (from Hess[4.14]):

1) the velocity potential satisfies Laplace's equation everywhere outside of the body and wake;

2) the disturbance potential due to the body vanishes at infinity;

3) the normal component of velocity is zero on the body surface;

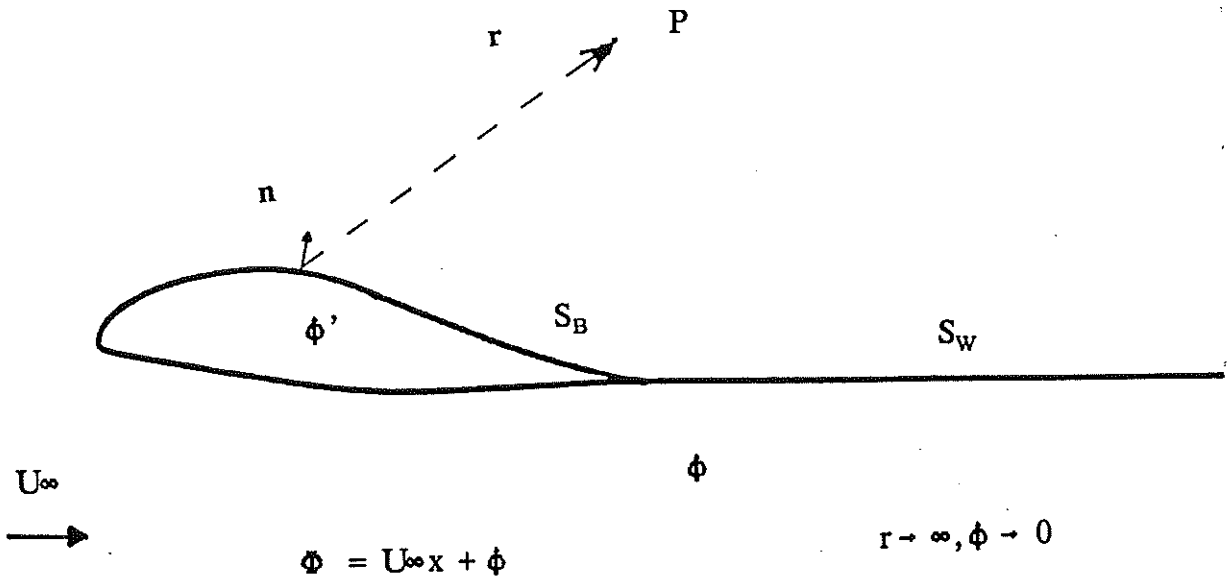


Figure 4.1 Two-dimensional lifting surface schematic

4) the Kutta-Joukowsky condition of a finite velocity at the body trailing edge is satisfied.

5) the trailing wake sheet is a stream surface with equal pressure either side.

For a steady-state solution the wake dipole strength distribution is uniquely determined by the application of the Kutta condition at the body trailing edge. As conditions (1) and (2) are satisfied as functions of μ and σ, conditions (3) and (4) are used to determine μ and σ on the body. The Kutta condition only applies at the trailing edge and some other relationship has to be used to uniquely determine the distribution of μ and σ over the body. The numerical resolution of this non-uniqueness is referred to as the singularity mix of the lifting-surface method.

The choice of numerical representation of the body integral singularity mix and boundary conditions are described in the following sections.

4.4 Morino's Method

Lee[4.17] carried out a two-dimensional investigation into four possible schemes for the solution of Lamb's equation. In this, the body source strength per unit area σ was defined as:

$$\sigma = \frac{\partial \phi}{\partial n} - \frac{\partial \phi'}{\partial n} \qquad [4.4]$$

where $d\phi/dn$ is the normal derivative of the velocity potential on the outer surface of the body and $d\phi'/dn$ is the normal derivative on the interior of the body surface. Similarly the body dipole distribution per unit area μ was defined as:

$$\mu = \phi - \phi' \qquad [4.5]$$

where ϕ and ϕ' are respectively the disturbance velocity potential on the exterior and interior of the body surface. Equation [4.3] can then be written as:

$$4\pi\phi(p) = \iint_{S_B} [(\phi(q) - \phi'(q))\frac{\partial}{\partial n}\frac{1}{R} - (\frac{\partial \phi}{\partial n} - \frac{\partial \phi'}{\partial n})\frac{1}{R}\ ]\ dS + \iint_{S_W} \Delta\phi \frac{\partial}{\partial n}(\frac{1}{R})dS \qquad [4.6]$$

As the flow within the body surface S is non-physical the choice of different internal velocity potential ϕ' will result in different singularity mixes. These were classified, by Lee, as follows.

1) Total velocity: If for Equation[4.6] the normal derivative with respect to the field

point is taken a velocity formulation results. The internal potential is specified such that on S_B the potential derivative is continuous so that:

$$\frac{\partial \phi}{\partial n} = \frac{\partial \phi'}{\partial n} \qquad [4.7]$$

The body source strength becomes identically zero and [4.6] becomes:

$$2\pi \frac{\partial \phi}{\partial n} = \iint_{S_B} \mu \frac{\partial^2}{\partial n_p \partial n_q} \frac{1}{R} dS + \iint_{S_W} \Delta\mu \frac{\partial^2}{\partial n_p \partial n_q} \frac{1}{R} dS \qquad [4.8]$$

This is an integral equation of the first kind for the unknown dipole strength μ when the dipole strength is given by:

$$\mu = \phi + \phi_\infty \qquad [4.9]$$

where ϕ_∞ is the inflow or free-stream potential

$$U_\infty = \nabla\phi_\infty \qquad [4.10]$$

It should be noted that Hess and Smith's original velocity method was based on a vortical formulation of [4.8] rather than a dipole distribution.

2) Perturbation velocity: For the velocity formulation, where the equation [4.6] is differentiated normal to the field point, if the internal potential ϕ' is set to zero it can be shown that $d\phi'/dn = 0$ on S_B. Therefore [4.6] becomes:

$$4\pi \frac{\partial \phi}{\partial n} = \iint_{S_B} \sigma \frac{\partial}{\partial n}\left(\frac{-1}{R}\right) dS + \iint_{S_B} \mu \frac{\partial^2}{\partial n_p \partial n_q} \frac{1}{R} dS + \iint_{S_W} \Delta\phi \frac{\partial^2}{\partial n_p \partial n_q} \frac{1}{R} dS \qquad [4.11]$$

where $\sigma = d\phi/dn$ and $\mu = \phi$. Given the source strength from the normal boundary condition this again is an integral equation of the first kind for the unknown dipole strength μ.

3) Total Potential: In equation[4.6], if the inflow velocity potential can be expressed as:

$$\nabla\phi_\infty = U_\infty \qquad [4.12]$$

and then the internal velocity potential is set constant and equal to the negative of the external free-strean potential then Equation [4.6] becomes [4.14]:

$$\phi' = -\phi_\infty \qquad [4.13]$$

$$2\pi\Phi_p = 4\pi\phi_\infty + \iint_{S_B} \phi(q)\frac{\partial}{\partial n}\frac{1}{R}dS + \iint_{S_W} \Delta\Phi\frac{\partial}{\partial n}\frac{1}{R}dS \qquad [4.14]$$

where Φ is the total potential. This represents the body as a normal dipole distribution and is a Fredholm integral equation of the second kind for the total potential Φ.

(4) Perturbation Potential: If the internal potential ϕ' is set to zero and from that $d\phi'/dn$ is zero, Equation [4.6] becomes:

$$2\pi\phi = \iint_{S_B} (\phi\frac{\partial}{\partial n}\frac{1}{R} - \frac{\partial\phi}{\partial n}\frac{1}{R})dS + \iint_{S_W} \Delta\phi\frac{\partial}{\partial n}\frac{1}{R}dS \qquad [4.15]$$

As $d\phi/dn$ is known, as it has to satisfy the normal surface boundary condition, this expression is also a Fredholm integral equation of the second kind for the dipole strength μ. This dipole strength is equal to the external value of the potential on the body surface.

The conclusion of Lee's study was, that for lifting surfaces which have both thin and thick sections (e.g. propeller blades), the perturbation potential method(4) taken from the work by Morino and Kuo[4.18] was the most suitable. The principal advantages of this method are that because panel potential (scalar) rather than velocity (vector) influence coefficients are calculated only a third of the memory requirement for the method is needed. Also, the perturbation potential influence coefficient is an order less singular. Kerwin and Lee [4.11] used this method and found it robust in their investigation of ducted propellers.

Maskew[4.19] used Morino's method which is a low-order panel method as opposed to higher-order formulations where the source/dipole strengths are not assumed piece-wise constant over a panel. The simplicity of low-order panel methods and their low computing cost allows great flexibility in their applications. In conclusion Maskew stated that a low-order method using Morino's formulation gave comparable accuracy to higher-order formulations while avoiding many problems associated with other low-order methods.

Margesson et al.[4.20] compared five production surface panel methods, three of which used low-order singularities. All but one used as a basis the formulation due to Morino and the other a higher-order formulation due to Hess. The two higher-order Morino formulations provided the same accuracy with lower panel numbers but at greater computational cost.

Morino's numerical procedure is based on representing the body surface by a series of N quadrilateral panels each with an unknown but constant dipole strength per unit area. The vertices of these panels are located on the actual surface of the body. The wake sheet is represented by M panels placed on the stream-surface from the trailing edge of the body surface. Its dipole strength per unit area is related to the difference in dipole potential at the trailing edge. In Morino's work the wake strength μ_w was equated to the difference in potential between the upper and lower surface at the trailing edge.

That is:

$$\mu_w = \phi_u - \phi_l \qquad [4.16]$$

On the body surface the source strength per unit area is prescribed by satisfying the condition for zero normal velocity at the panel centroid:

$$\sigma_s = \bar{U} \cdot \bar{n} \qquad [4.17]$$

where **n** is the unit normal outward from the panel surface and U the specified inflow velocity at the panel centroid.

The numerical discretisation of equation[4.15] gives for the potential at the centroid of panel i as:

$$\phi_i = \frac{1}{2\pi}\sum_{j=1}^{N} \left((U_\infty . n_j) S_{ij} - \phi_j D_{ij} \right) + \sum_{k=1}^{M} \Delta\phi_k W_{ik} \qquad [4.18]$$

where for panel j:

S_{ij} is the source influence coefficient of a unit strength panel;

Di_j the dipole influence coefficient;

W_{ik} the influence of the constant strength wake strip extending to infinity.

As there are N independent equations corresponding to the N body surface panel centroids, Equation[4.18] is closed and can be evaluated. Expressed in matrix form it becomes:

$$[D_{ij}]\underline{\phi} + [W_{ik}]\underline{\Delta\phi} = [S_{ij}] (\underline{U_\infty}.\underline{n}) \qquad [4.19]$$

For Morino's original trailing edge Kutta condition, which directly relates $\Delta\phi$ to the difference in trailing edge panel potential, the matrix expression [4.19] can then be directly solved to give the vector of dipole potentials $\boldsymbol{\phi}$. Numerical differentiation of dipole potential along the body surface allows the surface velocity and hence pressures on the surface to be evaluated.

The method used for the evaluation of the individual influence coefficient elements of the matrices S_{ij}, D_{ij} and W_{ik} are described in the next section.

4.5 Newman Panel

At the heart of a lifting surface panel method is the efficient calculation of the potential (or velocity) influence coefficients at a field point due to a particular panel's source or dipole distribution. Newman[4.21] derived expressions for calculating the exact influence coefficients of a constant strength distribution of sources and normal dipoles over a quadrilateral panel. The method of calculation of the dipole influence coefficient avoids the use of numerical integration. The approach of Newman was different from that used originally by Hess and Smith[4.8] although the form of the exact source influence coefficient is algebraically similar.

As the distance of the field point from the panel centroid increases approximate expressions are used for the influence coefficients to reduce computational effort. Following Hess & Smith, Newman also gives arbitrary order multi-pole expansions. These ensure that at greater distances from the panel the accuracy of the source and dipole influence coefficient is maintained while at the same time the computational time is reduced.

The scheme used in this work was to choose the scheme to calculate the influence coefficients based on the ratio of the distance between the point of interest and panel centroid divided by the length of the panel main diagonal (referred to as B).

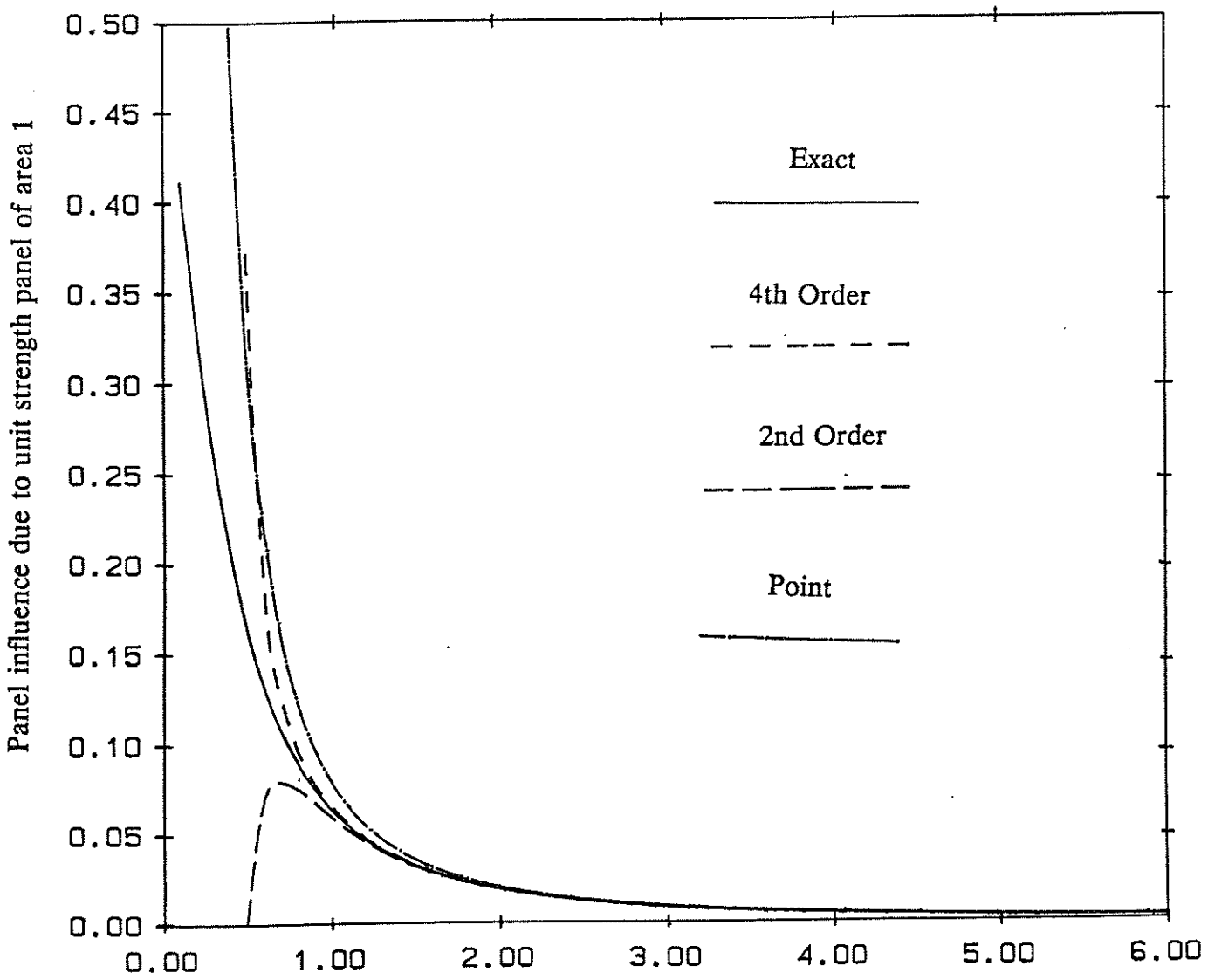


Figure 4.2 Comparison of scheme for calculating the dipole influence coefficient against distance above the centre of a square panel

Newman's exact formulation is used when B is between 0 and 2, a 4^{th} order multi-pole approximation between 2B and 2.45B, a 2^{nd} order multi-pole approximation between 2.45B and 4B, and a point source/normal dipole for distances greater than 4B. Figure 4.2 shows the influence coefficients for this composite scheme compared to that for the exact, 4^{th}, 2^{nd}, and point expressions for a field point at different heights above the centroid of the panel. Appendix D provides the mathematical derivation of the exact,

multipole and point expressions Their Occam2 interpretation is given in Ref. [4.22].

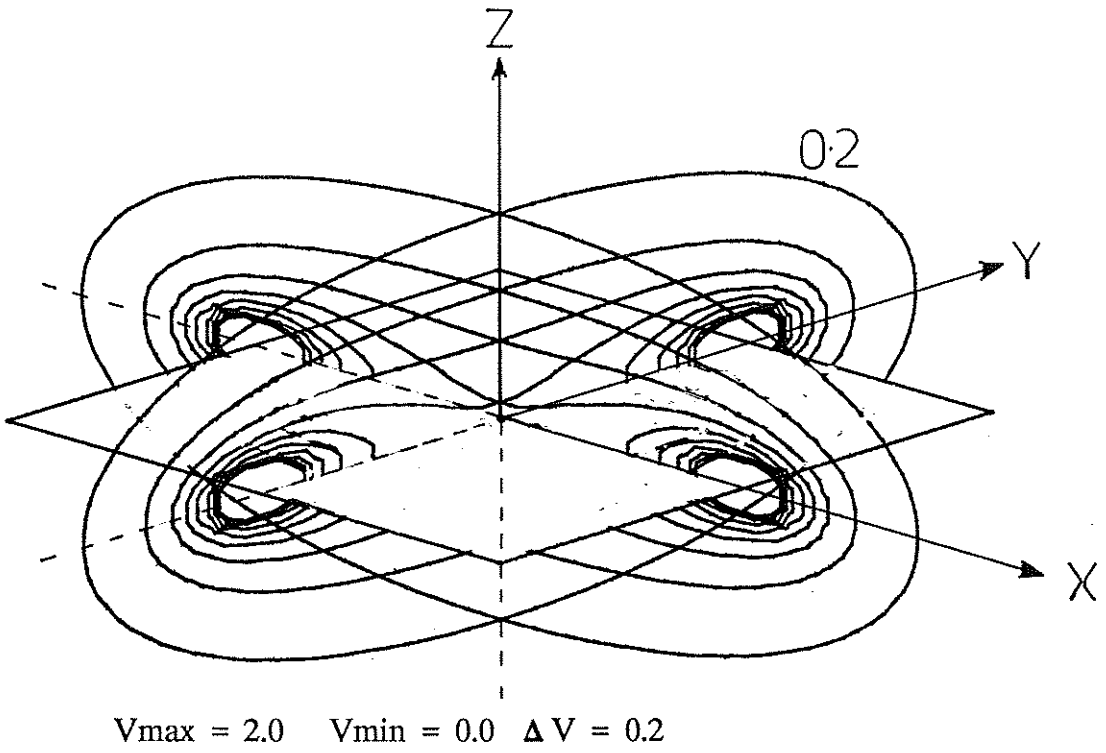


Figure 4.3 Isometric contour plot showing total velocity lines for two planes through a square dipole panel

The modelling of the interaction requires the evaluation of total velocity in the field domain away from the lifting surface. The expressions given by Newman are for the potential influence only. For potential flow, the velocity is the local gradient of the potential. Therefore, the velocity influence due to a panel has been derived by applying the Grad operator ∇ to the source and dipole influence expressions. These relationships are also given in Appendix D. Figure 4.3 shows, for two planes through a square panel, an isometric contour plot showing lines of constant total velocity for a uniform dipole distribution. The correspondence between a uniform dipole distribution and an element of four vortex lines can be directly seen. A cutoff is applied for the evaluation of the velocity influence within a distance of 5% of the length of an individual edge.

The multipole expressions require the calculation of 13 coefficients based on the geometrical properties of the panel. To minimise computational time it would be sensible to only ever calculate these values once for each panel. However, the memory requirement to do this severely limits the maximum number of panels for a problem and it was decided to recalculate the coefficients for the four panel vertices every time a panel was used. Table 4.1 provides timings for calculating these geometrical coefficients for a single panel and also for the four different methods for calculating the source and normal dipole influence coefficients.

Table 4.1 Comparison of time taken to calculate dipole influence coefficients using different schemes between a single T800 transputer and VAX 11/750.

	Clock Ticks T800 (low)	Time (mSec)	Vax 11/750 (Newman [4.21]) (mSec)
Geometry	119	7.62	-
Exact	27	1.73	1.5
4th Order	17	1.09	0.4
2nd Order	12	0.77	0.2
Point	9	0.58	-

These timings were obtained using a single T800 transputer running at low priority and are averaged over a large number identical calculations. The figures are given in both units of the T800 internal clock and milliseconds. Also shown are the approximate values given in Newman[4.20] for calculating the source and normal dipole influence coefficients in single precision on a VAX 11/750.

It can be seen that:

1) Calculation time of the geometrical coefficients is only 4.4 times that of calculating the exact influence coefficient. Therefore, if at any one time a number of field point

influence coefficients can be calculated using information for a single panel there will not be that large a computational penalty for recalculation.

2) As expected an individual T800 micro-processor is slower than the VAX 11/750 but only by a small amount.

4.6 Kutta Condition

For a steady-state solution, the dipole strength of the trailing wake sheet has a constant strength in the stream-wise direction. This strength is directly related to the circulation around the lifting surface. The original Kutta condition, implemented by Morino, involved setting the trailing wake sheet dipole strength equal to that of the difference in perturbation potential at the trailing edge. This implies that the pressure difference at the trailing edge would be close to zero. Lee showed that a source term should also be included to ensure that there was zero difference in <u>total</u> potential caused by the difference in source strength of the two trailing edge panels and gave the expression for this as

$$\Phi_s = \bar{U}_o.\bar{r} \qquad [4.20]$$

For three-dimensional flow, **r** is the vector between the centroids of the two trailing edge panels. With this additional term, when significant cross-flow occurs at the trailing edge, the upper and lower panels will not necessarily be at the same pressure and a non-physical trailing edge pressure loading occurs. This was seen by Lee as the need to explicitly equate the upper and lower panel pressure using an iterative scheme to correct the dipole wakestrength based on a factor K multiplying the pressure loading at the trailing edge from the previous iteration. Only sketchy details are given by Lee into the actual evaluation of K. However, the scheme is described as a Newton-Raphson method and from this hint an expression for K has been derived.

The method developed in this work uses the description given in [4.11, 4.17] as

a basis. An explicit condition of no pressure loading is enforced across the upper and lower trailing edge panels. That is, the pressures are equal:

$$\Delta p_{te} = p_u - p_l = 0 \qquad [4.21]$$

If it is assumed that Δ Cp is primarily a function of the local trailing edge wake sheet strength $\Delta\phi$ an iterative Newton-Raphson approach is suggested to determine the wake strength for the point of zero pressure difference at the trailing edge. That is:

$$\Delta\phi^{k} = \Delta\phi^{k-1} - \frac{\Delta p(\Delta\phi^{k-1})}{\frac{d\Delta p}{d\Delta\phi}} \qquad [4.22]$$

where the trailing edge pressure loading Δ p is the difference in pressure between the upper and lower panels at the trailing edge,[4.23], and Lee's K is the inverse of the derivative of Δ p with respect to the wake strength $\Delta\phi$.

$$\Delta p = p_U - p_L \qquad [4.23]$$

Substituting for pressure in terms of surface velocity V gives

$$\Delta p = (V_L \cdot V_L) - (V_U \cdot V_U) \qquad [4.24]$$

Differentiating [4.24] with respect to a change in wake strength $\Delta\phi$ gives

$$\frac{d\Delta p}{d\Delta\phi} = 2\left(V_L \cdot \frac{dV_L}{d\Delta\phi} - V_U \cdot \frac{dV_U}{d\Delta\phi}\right) = \left(\frac{1}{K}\right) \qquad [4.25]$$

By deriving an expression for surface velocity in terms of the velocity influence sum of all the source and dipole panels, and wake strips (see next section 6, equation[4.30]) and then differentiating with respect to the wake sheet strength and assuming that the principal influence on a pair of trailing edge panel's is due to the attached wake strip then $dV/d\Delta\phi$ can be expressed as:

$$\frac{dV_L}{d\Delta\phi} = V_{WL} \quad and \quad \frac{dV_U}{d\Delta\phi} = V_{WU} \qquad [4.26]$$

where V_{WL} and V_{WU} are the velocity influence coefficients of the wake strip attached to the panels at their respective centroids. All the components can then be numerically evaluated and hence the wake strength updated.

The zero'th order (k=0) approximation for the wake strength is taken to be the original Morino kutta condition:

$$\Delta\phi^0 = \phi_u - \phi_l \qquad [4.27]$$

as ϕ_u and ϕ_l are unknown then the numerical Equation[4.19] is arranged with the unknowns on the left hand side:

$$[D_{ij} + W_{ik}]\,\underline{\phi} = [S_{ij}]\,\underline{U_\infty \cdot n_j} \qquad [4.28]$$

Once the solution vector ϕ is obtained this is used to calculate Δ Cp at the trailing edge. Using Equation[4.22] the correction to the wake strength is found. This correction vector of known strength is multiplied by the wake strip influence coefficient matrix W_{ik} and applied to the right hand side of the equation. This modifies Morino's original matrix expression to:

$$[\,D_{ij} + W_{ik}\,]\,\phi = [\,S_{ij}\,]U_\infty \cdot n_j - [\,W_{ik}\,]\left(\frac{d\Delta\phi}{d\Delta p}\Delta p\right)^k \qquad [4.29]$$

The process is repeated until the pressure loading at the trailing edge has been removed to any significant degree.

4.7 Adaptive Wake Model

The accurate modelling of a wake sheet convected downstream from a lifting surface is difficult and the subject of much research. The zero thickness wake sheet should be aligned tangential to the local flow to ensure that there is no pressure jump across the sheet. The specification of the wake geometry cannot be known a priori as

its location will depend on the eventual flow solution which is dependent on the actual wake geometry. There are schemes which iteratively adapt the wake shape until a converged shape is obtained with the wake everywhere aligned with the local flow. These schemes greatly increase the computational cost of solution and are prone to problems of sensitivity to the initial geometry, especially where considerable wake roll-up occurs. Many of these problems originate from the non-physical representation of the wake as a zero thickness constant strength sheet extending to infinity. However, a correct geometrical representation of the wake sheet is necessary if good numerical results are to be obtained.

In the Morino formulation the wake sheet is represented as a series of trailing dipole panels. The influence coefficient of a wake strip W_k extending from the body trailing edge to infinity is the sum of the influence coefficients of all the panels making up the strip. For practical calculations the panelled wake strip will only extend a finite distance downstream. This distance can be determined by carrying out sensitivity studies where the length of wake strip is increased until no appreciable change in body dipole strengths occurs. Another, more physical approach, is to use a far-field approximation for a wake sheet extending to infinity from a given distance downstream of the body. This method was used by Lee.

For non-rotating lifting surfaces where the principal aim is to obtain surface pressure and hence body forces the wake sheet evolution downstream is not critical to the final solution. For these cases a fixed (or frozen) geometry wake can be prescribed. Usually downstream of the body the wake is aligned in the free-stream direction and close to the body the wake is aligned with the direction of the lifting surface chord.

The accurate representation of the wake sheet generated by a rotating lifting surface, such as a propeller blade, presents major difficulties. Considerable work has been done in developing propeller wake models. These usually involve some amount of empiricism and wake-adaption techniques. In Lee's work the prescribed propeller wake geometry was generated using a previously developed lifting-surface code[4.12]

and experimental data. In a recent paper by Maitre and Rowe[4.23] a numerical method for iteratively relaxing an initial wake following the propeller geometric pitch was found to converge. It gave good results, although was not completely free to adapt near the hub where the wake radius was held constant.

In this work, to allow for possible variation in propeller design, it was decided to investigate a wake adaption method based on that by Maitre and Rowe. The wake is divided into a near and far region. In the far region the movement of each wake panel is based on that of the last panel in the near region. For the near region the velocity at each panel centroid is calculated by summing the velocity influence coefficients of the body source and dipole panels and the wake dipole panels. The total velocity at the panel centroid is the sum of the body rotational velocity, inflow velocity and the disturbance velocity:

$$V_T = V_o + r \times \omega + \sum_{j=1}^{N} \left(\phi_j V_{D_j} + (V_o \cdot n_j) V_{S_j} \right) + \sum_{k=1}^{M} \left(\Delta\phi_k V_{W_k} \right) \qquad [4.30]$$

The total velocity $\mathbf{V}_t$ is translated into cylindrical coordinates where **a** is a unit vector in the direction of rotation, **r** a unit vector in the direction of the panel centroid perpendicular to **a**, and **t** forms the orthonormal set **t** = **a** x **r**. This gives

$$\begin{aligned} V_a &= V_T \cdot a \\ V_r &= V_T \cdot r \\ \omega &= \frac{(V_T \cdot t)}{R} \end{aligned} \qquad [4.31]$$

where R is the radial position of the panel centroid. As the velocity of the panel nodes is required this is taken to be the sum of the cylindrical components of velocity of the surrounding panels divided by the number of panels surrounding the node. That is

$$V_{node} = \frac{1}{N_p} \sum_{1}^{N_P} V_{centroid} \qquad [4.32]$$

The positions of the nodes of a wake strip are altered recursively starting from just aft of the trailing edge using the following relationships which give, in cylindrical coordinates (r,aθ), for node k+1(Maitre).

$$\begin{aligned} r^{i}_{k+1} &= r^{i}_{k} + v_r \Delta t_k \\ \theta^{i}_{k+1} &= \theta^{i}_{k} + \omega R \Delta t_k \\ a^{i}_{k+1} &= a^{i}_{k} + v_a \Delta t_k \end{aligned} \qquad [4.33]$$

The convective time step in this work is taken to be

$$\Delta t_k = \frac{L}{|V_T|} \qquad [4.34]$$

where L is the straight-line distance between two successive nodes. Using this method it ensures that the stream-wise length of the panel sides remains constant.

This method was also used for non-rotating bodies by working in cartesian coordinates for the recursive relationships. That is :

$$\left[x_{k+1}, y_{k+1}, z_{k+1} \right]^{i} = \left[x_k, y_k, z_k \right]^{i} + \left[u, v, w \right] . \Delta t \qquad [4.35]$$

4.8 Geometrical Definition

4.8.1 Introduction

Morino's method locates quadrilateral panels on the actual body surface. For lower-order formulations these panels consist of a number of connected straight lines which form a closed surface. Generally, only quadrilateral panels are used and where possible one of the principal directions of the panels is aligned in the flow direction.

An accurate geometrical definition of a three-dimensional body as a closed surface constructed from quadrilateral elements is a crucial component of a lifting-surface analysis. How easily arbitrary bodies can be defined will determine the

usefulness of a numerical analysis code. Lee [4.17] tailored the flow solver for a particular geometry e.g. semi-span wing or propeller blade. This approach is of limited use and a better approach is that used in the commercial code Quadpan[4.19] where an input file is used which contains the four vertices for each panel. The preparation of this pre-processing file is especially time consuming as a new file has to be created for every change in panelling density. However, such an input file format allows arbitrary bodies to be tested without recourse to creating individual executable code for every geometry.

In this work one of the principal features is the investigation of the performance of a lifting-surface code on a transputer network. Therefore, it is necessary to have a simple means of scaling the overall problem size by altering the number of panels used to define a lifting-surface. Therefore, the decision was made to combine the two approaches described and generate the actual panel vertex coordinates within the program but use a pre-processing file to define the number of bodies and their individual geometry. This allows a problem to be scaled by using the internal panel generator to produce a different number of panels for the same overall body geometry. In addition the pre-processing input file can be used to define the interaction velocity field.

4.8.2 Parametric Cubic Spline

A variety of means are available for defining a three-dimensional surface (or body). A ship hull form is conventionally defined using a series of lines which lie in parallel planes. These lines, whether waterlines, buttocklines or transverse sections, are themselves defined in terms of an ordered set of coordinates. A mathematical relationship can then be used to generate the curved lines between the coordinates and hence specify a three-dimensional surface.

An extremely useful and straightforward means of relating the line coordinates to the curve passing through them is that of a parametric cubic spline procedure. A

spline approximation is defined (Kreyszig [4.24]) as a piece-wise polynomial approximation to a curve. Each segment of a line is represented as a polynomial . For a cubic spline at the end of each segment the gradient and curvature of the polynomial expression are matched to the adjoining polynomial expressions. This results in a curve made up of a series of cubic lines i.e.

$$y = k_1 t^3 + k_2 t^2 + k_3 t + k_4 \qquad [4.36]$$

where the values of the constants k_1,k_2,k_3 and k_4 are solved using the end conditions of gradient and curvature continuity. Defining a value of the parameter t will uniquely define the value of the y coordinate. Similar relationships for the x and z coordinates allow a three-dimensional curve to be uniquely determined by the single parameter t. This parameter is the distance along the original curve and this is usually approximated as the straight-line distance between points. That is:

$$\Delta t = \sqrt{(X_1 - X_2)^2 + (Y_1 - Y_2)^2 + (Z_1 - Z_2)^2} \qquad [4.37]$$

For the purposes of this work a surface definition using parametric cubic spline procedures provides an accurate approximation to a three-dimensional surface. The end condition used throughout this work was that of zero curvature.

4.8.3 Body Definition

The facility to define a number of bodies or separate parts of the same body independently allows complex geometries and flows to be investigated. If each body is defined relative to its own body coordinate system there has to be a means of relating the coordinate systems of all the bodies to the overall cartesian coordinate system in which the panels are defined. A set of four vectors (**S**, **P**, **O**, **A**) were used to carry out this transformation. The definition of these vectors allows great flexibility for the parametric studies of complex geometries.

Analytically the flow solution for incompressible potential flow can be made non-

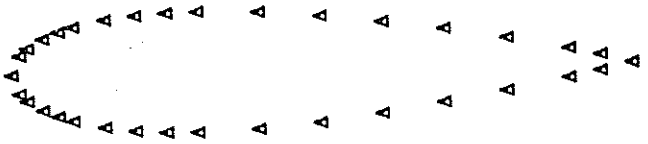
dimensional and independent of physical scale. However, to allow direct comparison with experiment it was decided to use the actual physical dimensions for the numerical analysis. Often geometries, for example rudder sections, are given in terms of non-dimensional percentage chord. To allow the checking of data and also the alteration of such parameters as rudder aspect ratio it was decided to define body coordinates in terms of a scale vector **S**:

$$S = \begin{bmatrix} X_{sca} \\ Y_{sca} \\ Z_{sca} \end{bmatrix} \qquad [4.38]$$

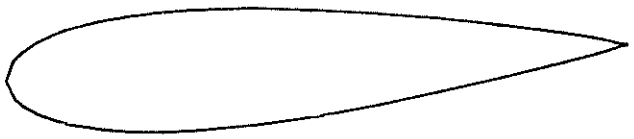
where X_{sca}, Y_{sca}, Z_{sca} are a scale factors for each of the body cartesian coordinates.

A body pivot vector **P** is defined which, in the body coordinate system, locates the position about which the body can be rotated. An ordered rotation vector **A** defines these angular rotations about the pivot in degrees. These rotations are in order respectively about the body X,Y and Z axis. For example, for a rudder with a pivot at 25% of its chord, it can be set at any angle of incidence by changing the value of Z-axis rotation angle component of vector **A**. Effectively, this allows the individual body coordinate systems to be rotated with respect to the overall coordinate system.An absolute offset vector **O** relates the origin of each body coordinate system to that of the overall coordinate system.

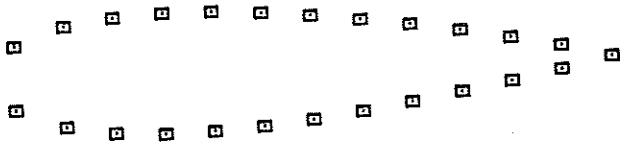
Each individual body (or part body) is defined in the same manner as that of a ship hull form; as an ordered series of lines with each line containing an ordered set of three-dimensional points. For a closed lifting body such as a rudder or wing, a wake sheet will be connected to the trailing edge and it is therefore sensible to start and finish each body definition line at the trailing edge. For non-lifting bodies this is not essential. Figure 4.4 illustrates the process of defining a rudder as a series of parametric cubic splines. The order of the series of lines and points is important in determining the direction of the panel surface normal used in calculating panel



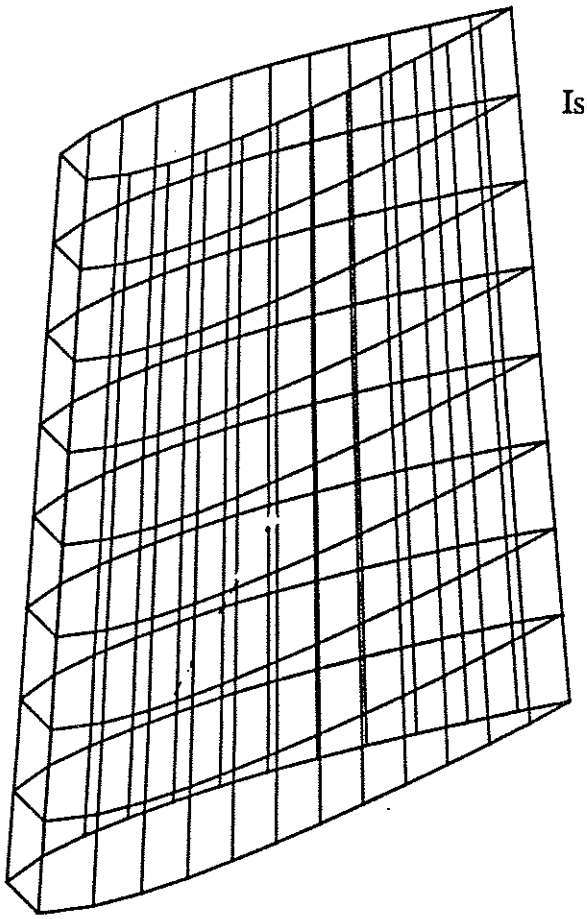
(1) Definition Points



(2) Spline through points



(3) Panel nodes



Isometric view of panelled rudder

Figure 4.4 Process for defining a rudder surface using parametric splines

influence coefficients. The lines are ordered so that the normal vector to a panel always faces out into the exterior flow field.

The numerical discretisation of the body geometry into a number of quadrilateral panels requires the number of panels in the section data direction (parameter t) Nt and in the line direction (parameter s) Ns. The process of generating the Nt by Ns panels is carried out by:

1) Producing (Nt + 1) coordinates for each line of section data using a parametric cubic spline through the section data. The distribution of points within a section can either be spaced at intervals of Δ t, where

$$\Delta t = \frac{|T|}{N_t} \qquad [4.39]$$

and T is the total parametric length of the line or where Δ t is some function of t. For closed bodies the start and finish point of each line are made identical.

2) A parametric cubic spline is formed in parameter s by using the i^{th} point from each of the section data splines. Each of these new cubic splines is used to define (Ns+1) points. Repeating this for all (Nt+1) points generates the coordinates vertices for all the required panels.

A wake sheet can be panelled in a similar manner. To ensure accurate matching of a body trailing edge and its wake sheet the number of section lines must be the same for body and wake. Also, the section lines should have identical start coordinates. This ensures that an identical s parameter spline will be produced and sub-divided so that wake strip coordinates correspond to those of the trailing edge panels.

4.8.4 Flow Definition

The disturbance velocity field generated by a body is superimposed on the

velocity field existing in the absence of the body. For many problems this inflow velocity field is a constant velocity in the free-stream direction throughout the domain and can be directly specified. In the case of a rotating body the velocity on its surface will be the vector sum of the free-stream velocity and the body's rotational speed. That is:

$$\underline{V} = \underline{U}_{\infty} + \underline{r} \times \underline{\omega} \qquad [4.40]$$

where **r** is the position vector of the panel centroid from a point at the origin of the axis of rotation. The angular velocity $\boldsymbol{\omega}$ vector is the scalar speed of rotation in the direction of the axis of rotation.

The modelling of the interaction between a ship rudder and propeller requires a spatially varying flow definition in the absence of a lifting-surface. So in addition to the uniform free-stream velocity and where necessary rotational velocity a spatially varying velocity distribution is needed. To facilitate modelling, a velocity field definition is used whereby the u,v and w components of velocity corresponding to velocities in the x,y and z directions are specified at a uniform spacing within a three-dimensional block. This is illustrated in Figure 4.5. A cylindrical and cubic block were respectively used for the propeller and rudder inflow fields.

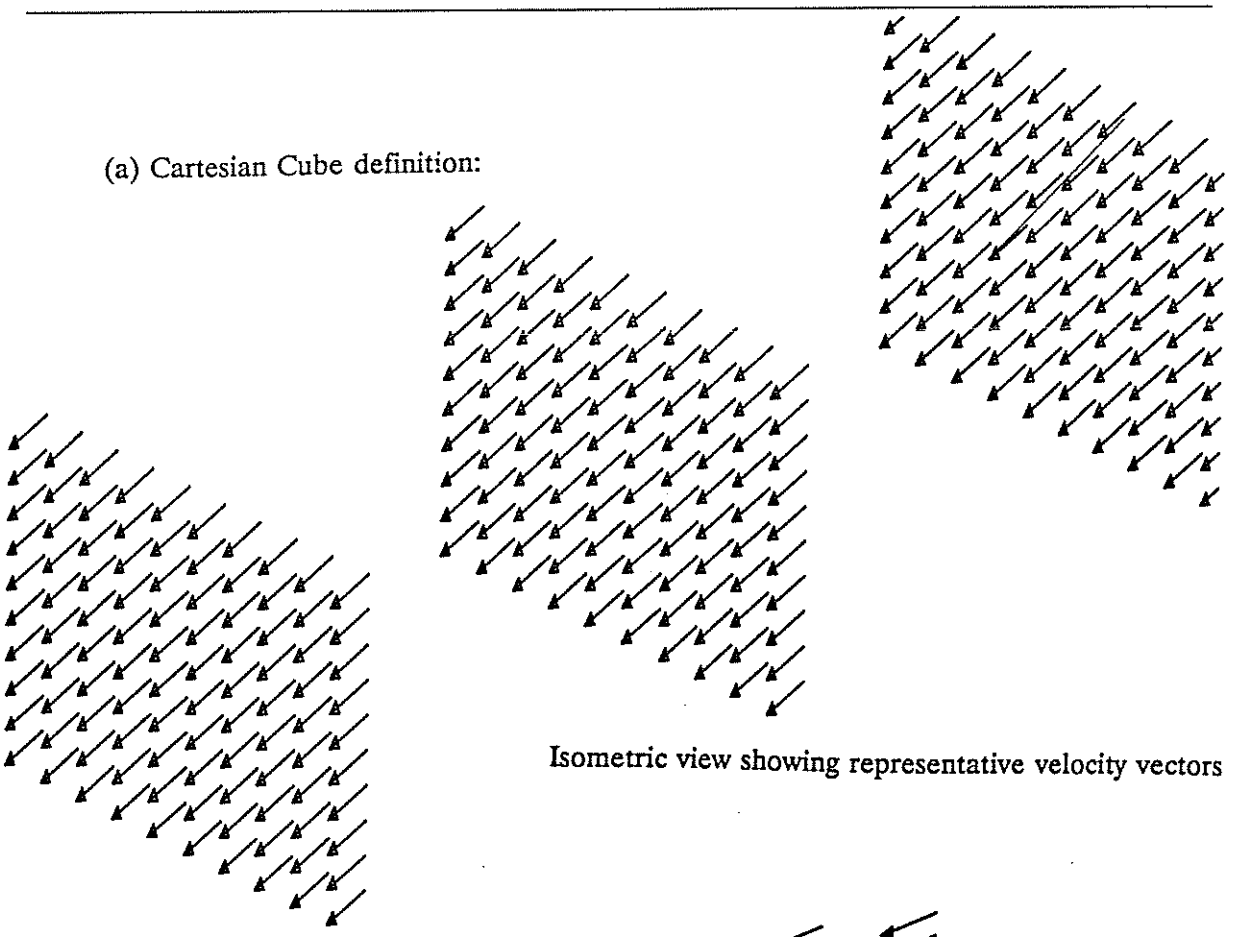
Theoretically, the interaction velocity field could be replaced by the location of suitably placed source/dipole panels of known strength and used directly in the calculation of influence coefficients. However, it is more convenient to specify a velocity field. If this velocity field satisfies Laplace's equation it is identical. That is if:

$$\nabla^2\phi = 0 \qquad [4.41]$$

then as

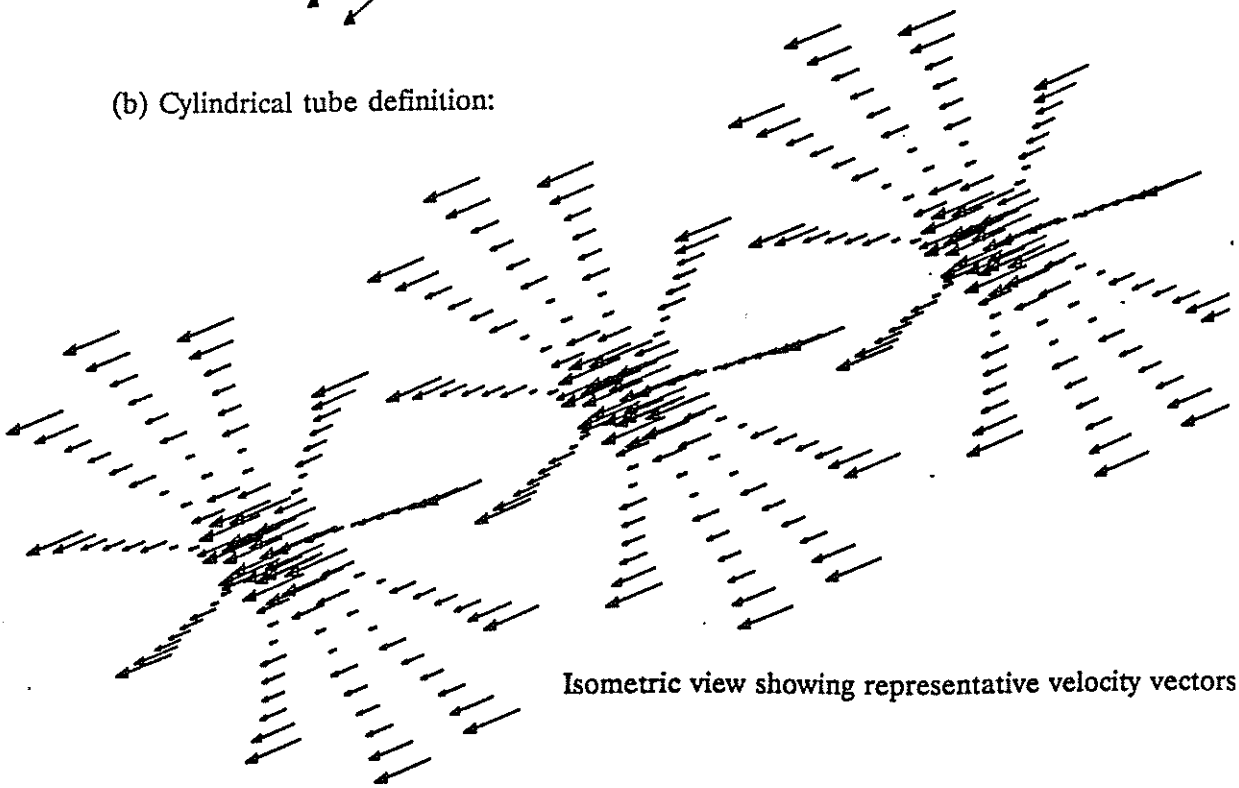
$$u = -\frac{\partial\phi}{\partial x},\ v = -\frac{\partial\phi}{\partial y},\ and\ w = -\frac{\partial\phi}{\partial z} \qquad [4.42]$$

(a) Cartesian Cube definition:



Isometric view showing representative velocity vectors

(b) Cylindrical tube definition:



Isometric view showing representative velocity vectors

Figure 4.5 Velocity Field definitions: Cube and tube

to satisfy Laplace's equation the interaction velocity field **Ui** must satisfy:

$$\nabla \cdot \underline{U}_i = 0 \qquad [4.43]$$

or expressed as the sum of velocity differentials should everywhere equal zero:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} + \frac{\partial w}{\partial z} = 0 \qquad [4.44]$$

It is probably acceptable (and more practical) to allow a certain amount of deviation from this condition. Possible difficulties will arise from the use of circumferentially averaged quantities in deriving interaction velocity fields and numerical problems close to panel edges.

4.9 Calculation of Aerodynamic Coefficients

The numerical solution of Morino's method gives a result vector which specifies a dipole strength at the centre of each panel. As explained previously, this corresponds to the potential ϕ on the surface of the body. To obtain practical engineering information from this surface potential distribution a numerical differentiation has to be carried out. The differentiation gives the disturbance velocity tangential to the panel surface. The total velocity at the panel centroid $\mathbf{U}_t$ is the vector sum of the tangential disturbance velocity $\mathbf{U}_d$, and the normal component of the body surface $\mathbf{r} \times \boldsymbol{\omega}$ and interaction field velocity $\mathbf{U}_i$.

$$U_t = U_d + \left(U_i - (U_i \cdot n)\, n \right) + \left(r \times \omega - \left((r \times \omega) \cdot n \right) n \right) \qquad [4.45]$$

where **n** is a unit vector normal to the panel surface.

There are two methods of obtaining the disturbance tangential surface velocity[4.18]. That is either by fitting a parametric cubic spline through the panel centroids and using the cubic polynomial constants to obtain the gradient and hence

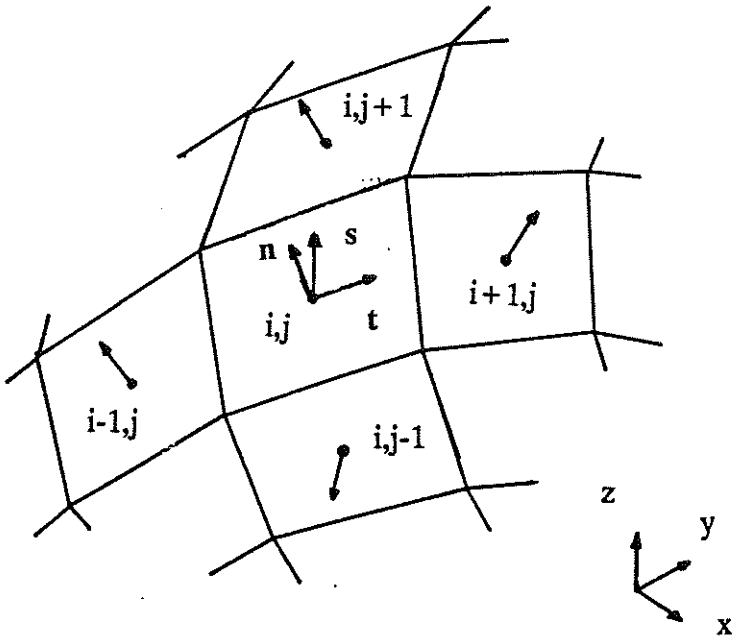


Figure 4.6 Detail of surface panel vectors for calculating surface velocity

velocity, or by a finite difference approach. The spline approach requires the assemblage of information from all the panels in a particular parametric direction. On the other hand, the finite difference method only requires information about its four neighbouring panels. This is shown in Figure 4.6, with the two unit vectors s and t in the local parametric directions. The velocity in the t and s directions are then obtained using a second order central difference:

$$u_t = \frac{\left(t_{i,j} - t_{i-1,j} \right)\left(\Phi_{i+1,j} - \Phi_{i,j} \right)}{\left(t_{i+1,j} - t_{i-1,j} \right)\left(t_{i+1,j} - t_{i,j} \right)} - \frac{\left(t_{i+1,j} - t_{i,j} \right)\left(\Phi_{i,j} - \Phi_{i-1,j} \right)}{\left(t_{i+1,j} - t_{i-1,j} \right)\left(t_{i,j} - t_{i-1,j} \right)} \quad [4.46]$$

and

$$v_s = \frac{\left(s_{i,j} - s_{i,j-1} \right)\left(\Phi_{i,j+1} - \Phi_{i,j} \right)}{\left(s_{i,j+1} - s_{i,j-1} \right)\left(s_{i,j+1} - s_{i,j} \right)} - \frac{\left(s_{i,j+1} - s_{i,j} \right)\left(\Phi_{i,j} - \Phi_{i,j-1} \right)}{\left(s_{i,j+1} - s_{i,j-1} \right)\left(s_{i,j} - s_{i,j-1} \right)} \quad [4.47]$$

The finite difference was chosen because it is easier to implement and, for panels which closely follow the curved surface, of comparable accuracy to that of the parametric spline method.

Having determined the surface velocity in the parametric coordinate system a

transformation has to be carried out to give the surface velocity components in the overall coordinate system. Unit vector **s** and **t** are not necessarily orthogonal and therefore the velocities are first transformed into an orthogonal system with one direction normal to the panel. The u,v and w components can then be found. The combined expression as given by Lee[4.17] becomes:

$$\underline{U}_d = \frac{\frac{d\phi}{dt}\left(t - (s.t)\, s\right) + \frac{d\phi}{ds}\left(s - (s.t)\, t\right)}{||\, sxt\,||^2} \qquad [4.48]$$

Knowing the disturbance velocity $\mathbf{U}_d$ and hence total velocity $\mathbf{U}_t$ allows the local non-dimensional pressure coefficient Cp to be found.

$$Cp = 1 - \left|\frac{U_t^2}{U_\infty^2}\right| \qquad [4.49]$$

The integration of the pressure distribution over the N panels defining the body surface allows the total potential pressure force **F** on a body to be evaluated as a vector sum, where A_i is the area of the i^{th} panel and $\mathbf{n}_i$ the direction of its unit surface normal,

$$F = \frac{1}{2}\rho\, U_\infty^2 \sum_{i=1}^{N} C_{Pi} \,.\, A_i .\, n_i \qquad [4.50]$$

and N the number of panels on the body.

The calculation of the pressure components of the non-dimensional body force and moment coefficients requires a further transformation into the correct body coordinate system. For example, as shown in figure 4.7 for a ship rudder at incidence with the x direction in the free-stream direction and z vertical the lift is the **j** component of **F**. That is:

$$C_L = \frac{F \,.\, j}{\frac{1}{2}\rho\, U_\infty^2 s\bar{c}} \qquad [4.51]$$

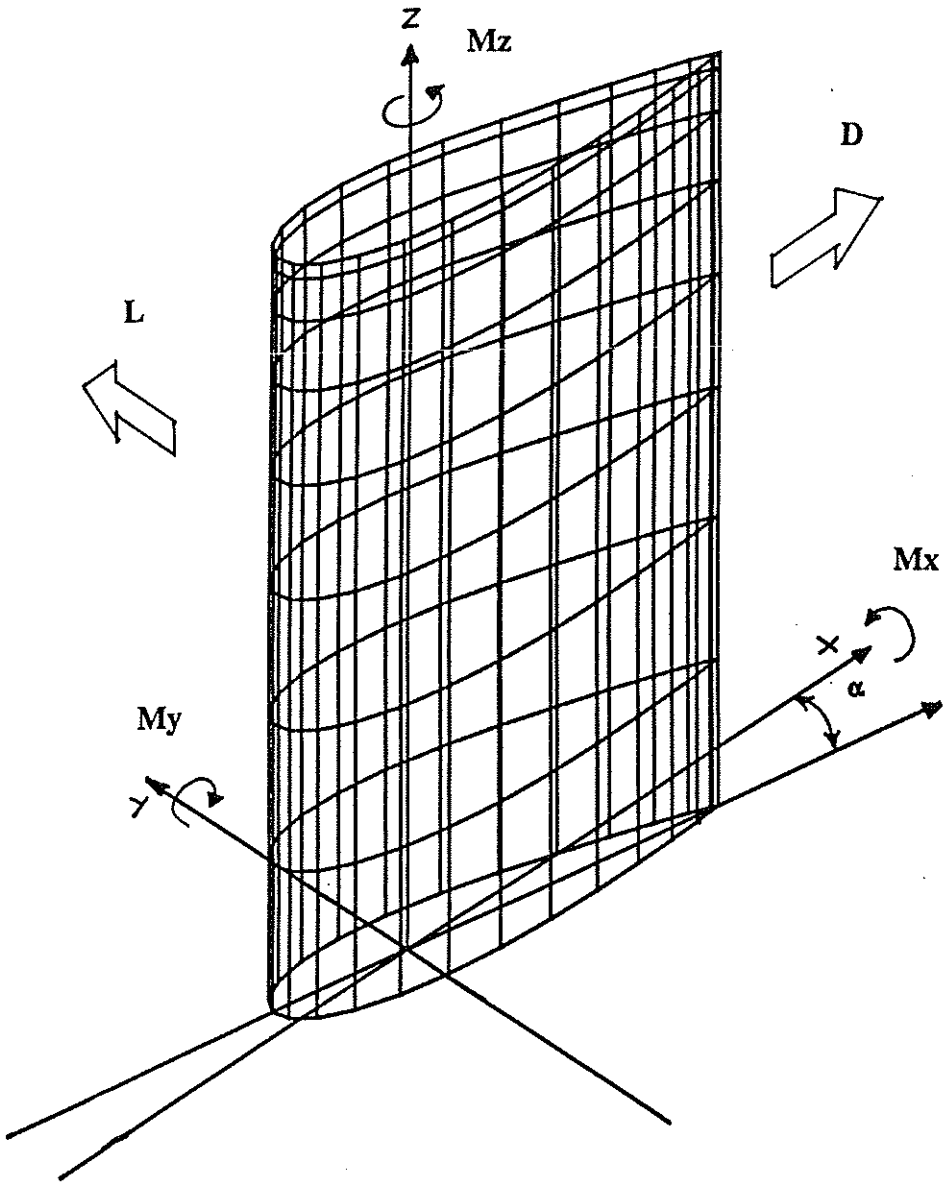


Figure 4.7 Direction of forces and coordinate system on a rudder at incidence

where s and c are respectively the rudder span and mean-chord. The pressure component of Drag is correspondingly:

$$C_D = \frac{F \cdot i}{\frac{1}{2}\rho U_\infty^2 s\bar{c}} \qquad [4.52]$$

An estimate of the viscous skin friction force acting on a lifting-surface can be found by using the panel surface velocity and distance from the leading edge to estimate the skin friction coefficient C_f. This gives a viscous force contribution equal to:

$$F_{visc} = \frac{1}{2}\rho \sum_{i=1}^{N} C_f A_i (V_T \cdot V_T) \mathbf{v} \qquad [4.53]$$

where **v** is a unit vector in the local flow direction. The skin friction coefficient is calculated in terms of local Reynolds number:

$$Rn = \frac{| U_\infty | s}{\nu} \qquad [4.54]$$

where s is the distance to the leading edge. The expressions used for C_f are from Schlicting[4.25]:

$$\begin{aligned} Rn < 3x10^5 \quad & C_f = 0.664 Rn^{-0.5} \\ 3x10^5 \le Rn < 1x10^7 \quad & C_f = 0.074 Rn^{-0.2} - 1050 Rn^{-1} \end{aligned} \qquad [4.55]$$

Combining the viscous and pressure contributions gives the total force $\mathbf{F}_T$ acting on the body as:

$$F_T = F + F_{visc} \qquad [4.56]$$

It should be noted that this neglects the change in pressure force due to viscous effects. Similar expressions are derived to give the total moment acting about the body pivot **P**, where $\mathbf{L}_i$ is the position of the panel relative to the origin of the body axes.

$$M = \frac{1}{2}\rho U_\infty^2 \sum_{i=1}^{N} C_{Pi} A_i (L_i - P) \times n \qquad [4.57]$$

$$M_{visc} = \frac{1}{2}\rho \sum_{i=1}^{N} C_f A_i (V_T \cdot V_T) (L_i - P) \times \mathbf{v} \qquad [4.58]$$

$$M_T = M + M_{visc} \qquad [4.59]$$

4.10 Verification

4.10.1 Introduction

To verify the numerical implementation of the assembled theoretical method a number of trial geometries were tested. The geometries used are the same as those by Lee and allowed comparison both with Lee's work and the original source. The test cases presented are to verify the numerical analysis procedures.

It was not possible to carry out extensive sensitivity studies on these test cases, for instance on panelling density, due to a maximum limit of 400 on the number of panels during the development phase of the code. However, it was considered that if the results sufficiently replicated those by Lee then such tests were unnecessary.

4.10.2 Ellipsoid

Lamb[4.15] gives an analytical solution for the surface potential of an ellipsoid geometry. For a non-lifting configuration (symmetrical flow) a direct comparison between the potential found using the numerical method and the analytical solution can be made.

The ellipsoid used for this test has a surface defined by:

$$x^2 + y^2 + c^2z^2 = 1 \qquad [4.60]$$

where the value of c was taken as 0.1. The perturbation potential on ellipsoid surface given by Lamb is:

$$\phi\,(\,x, y, z\,) = U_\infty\, x\, K \qquad [4.61]$$

where

$$K = \frac{c}{2-\alpha_o} D \qquad [4.62]$$

$$D = \int_0^\infty \frac{d\lambda}{(1+\lambda)^{\frac{3}{2}}(1+\lambda)^{\frac{1}{2}}(c^2+\lambda)^{\frac{1}{2}}} \qquad [4.63]$$

$$\alpha_o = cD \qquad [4.64]$$

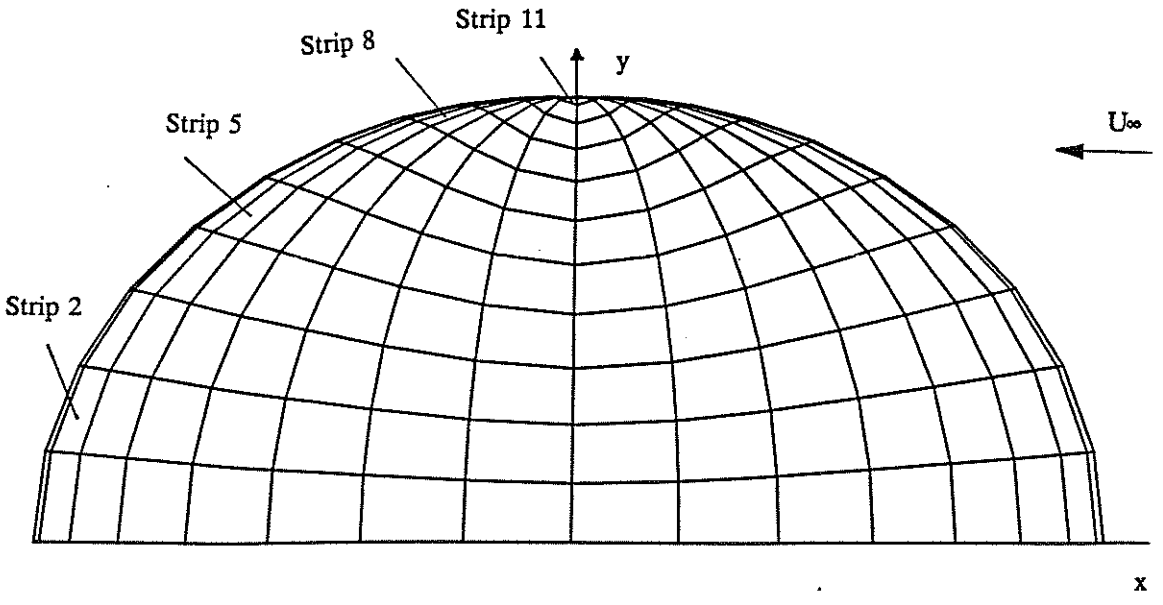


Figure 4.8 Panelling arrangement for an ellipsoid

Figure 4.8 shows the geometric representation of the ellipsoid which had 32 chordwise and 12 spanwise panels with a cosine and sine panel distribution respectively. The sinusoidal panel distributions concentrates the panels in the areas of large variation

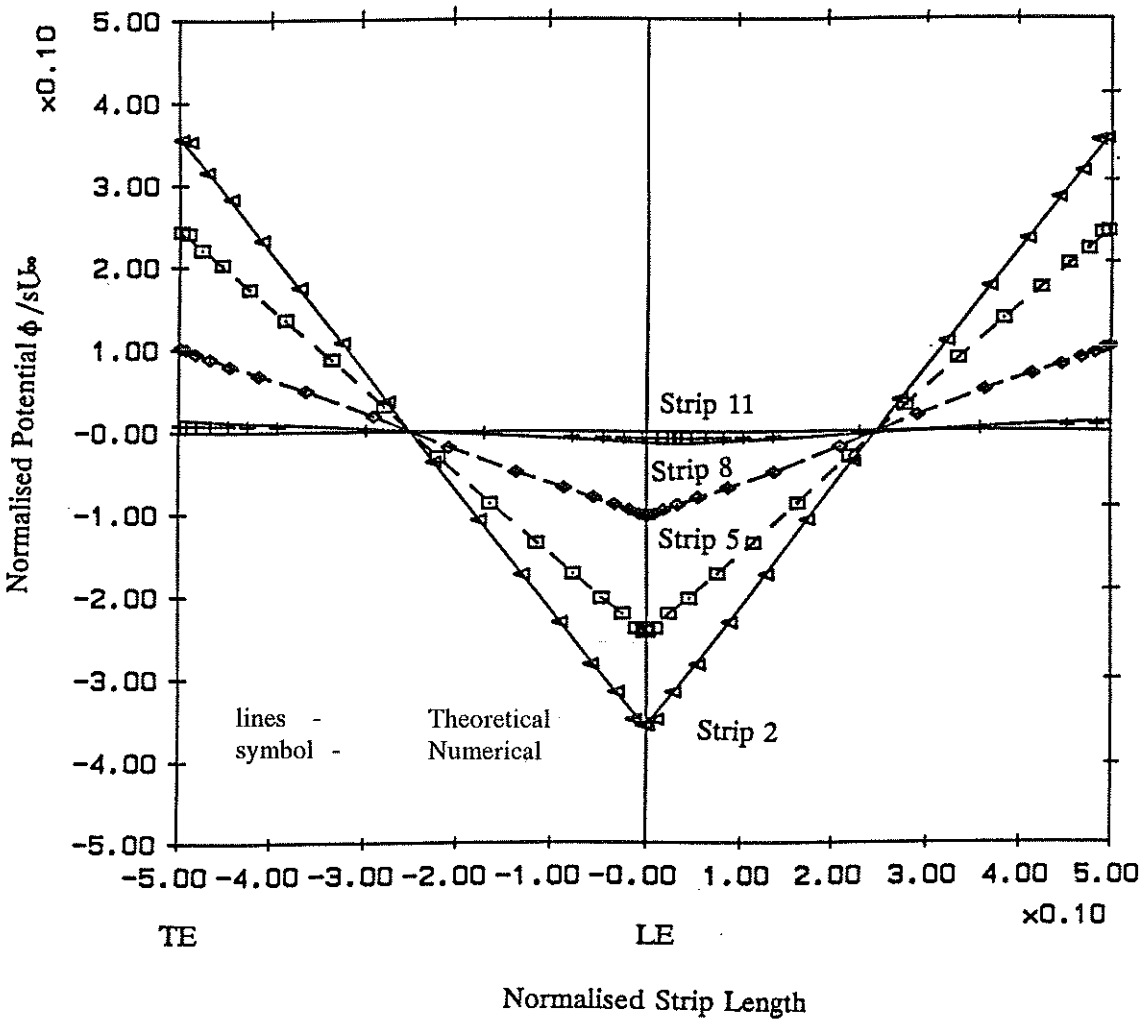


Figure 4.9 Comparison of theoretical and numerical potential for an ellipsoid with a=1, b=1 and c=0.1

of surface curvature and potential. In Figure 4.9 the numerical potential for four spanwise strips is compared with the corresponding theoretical value. They are plotted to a base of normalised parametric length of each strip. The potential is normalised with respect to the free-stream velocity and ellipsoid radius. It can be seen that the comparison is very good although in the region of the tip there are small discrepancies. This was found by Lee who noted that the discrepancies were magnified when the

surface velocity is calculated but the effect was localised at the tip.

4.10.3 Circular wing

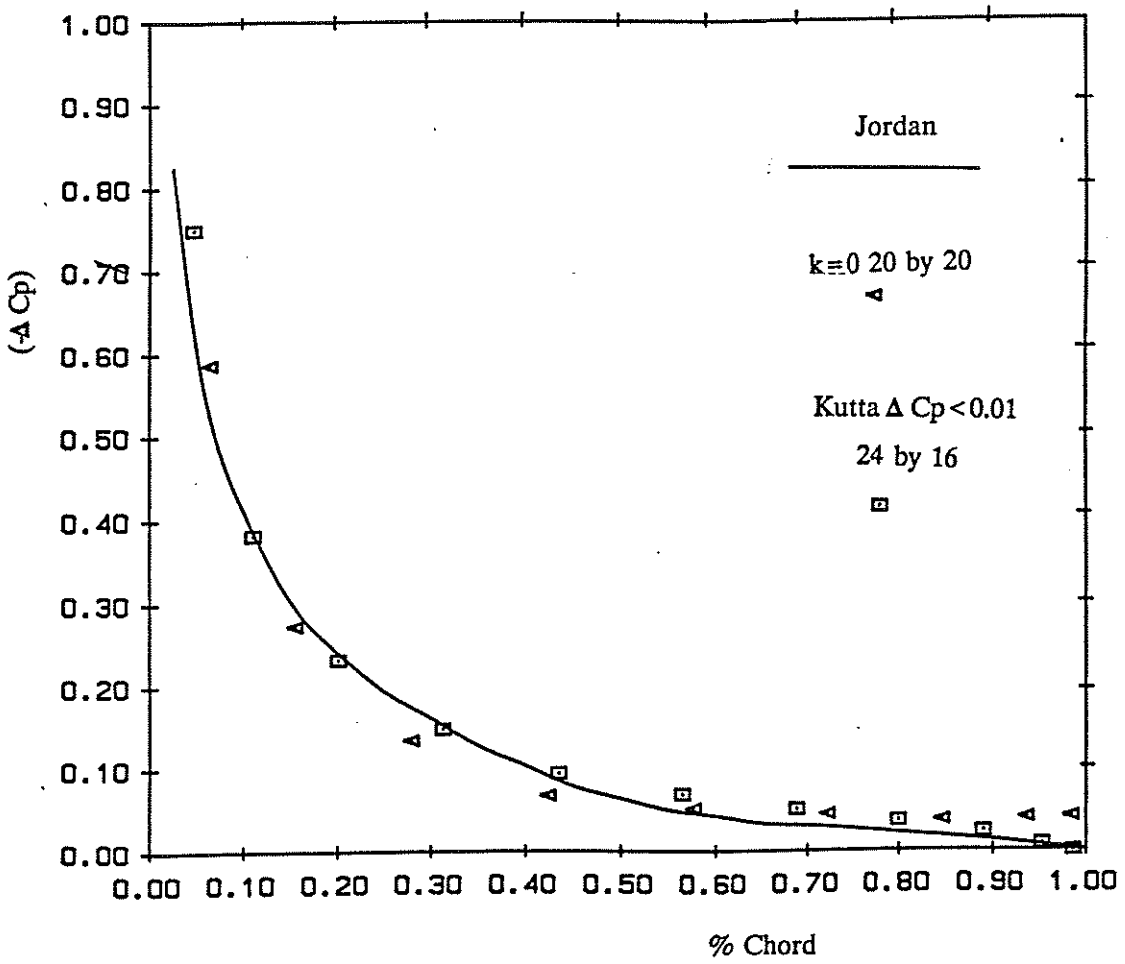


Figure 4.10 Chordwise pressure distribution for circular planform wing at 91% semi-span at 0.1 radian incidence

Jordan [4.26] gives an analytic solution to the flow over a circular planform wing of zero thickness. By comparing numerical solutions at 1% and 5% thickness/chord ratio sections for a NACA 4 digit series section with that of the analytic solution an assessment can be made on the prediction of lift for an aerofoil at incidence. As noted by Lee, this geometry represents the flow in the region of a propeller tip. The panelling arrangement used was similar to that for the ellipsoid. The limit on panel numbers restricted the accuracy obtainable (a higher number of spanwise strips is

required to achieve the converged panel density of Lee). However, as shown in Figure 4.10, there is a good comparison of chordwise pressure distribution for spanwise strip 9 for the 1% aerofoil between the analytic solution and the numerical result run with and without the iterative Kutta condition. The importance of the application of the explicit Kutta condition can be seen in removing the spurious trailing edge pressure loading.

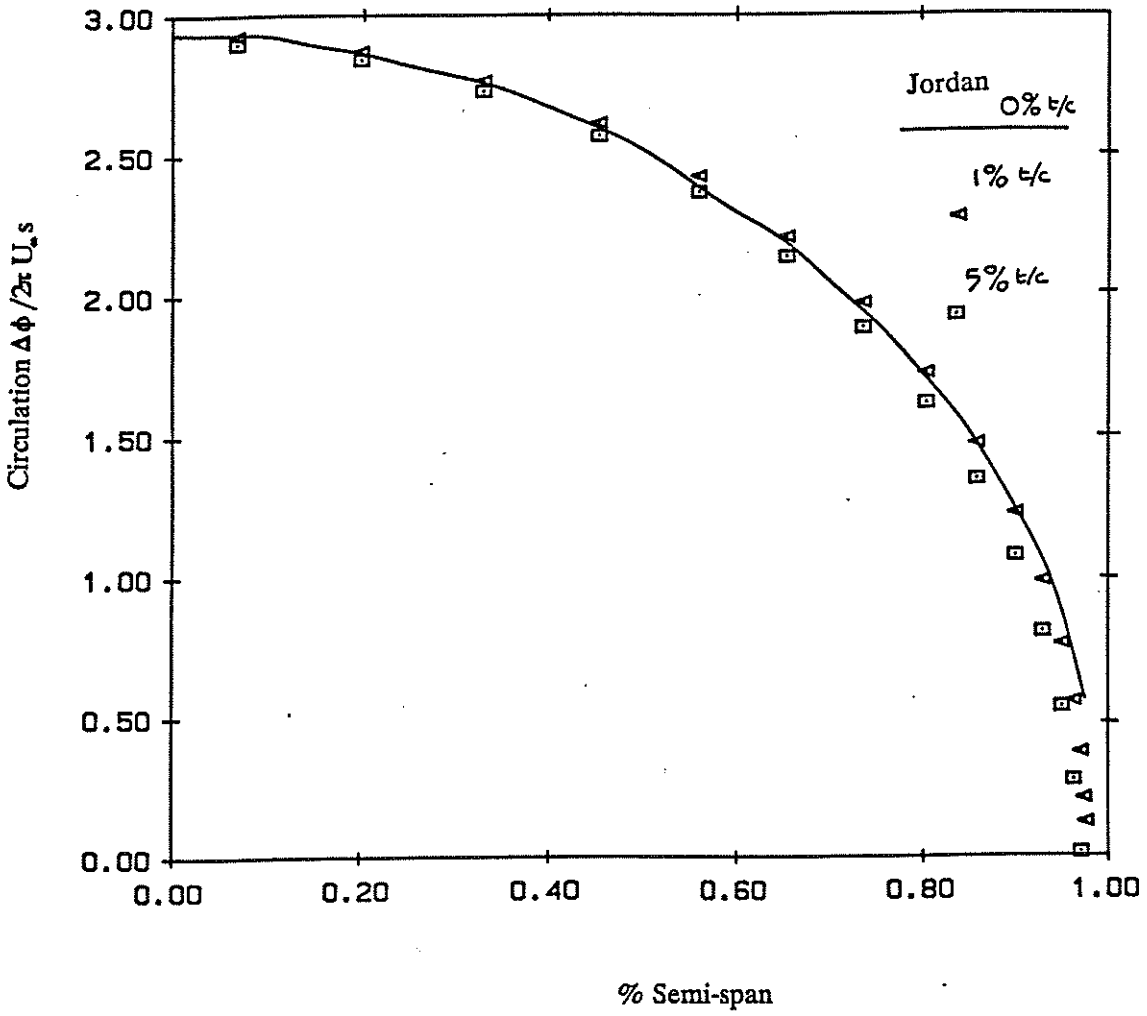


Figure 4.11 Spanwise circulation distribution for circular planform wing at 0.1 radian incidence

The effect of wing thickness in comparing the spanwise circulation distribution between the analytic (0% t/c) with the numeric solution (1% t/c, and 5% t/c) is shown in Figure 4.11. The result for the 5% and 1% distributions are close to the analytic solution with the 5% distribution slightly further away. Overall, the results for the circular wing show good correspondence with the analytic solution even with the limited number of panels. The necessity of applying the explicit Kutta condition is demonstrated for geometries with significant cross-flow components.

4.10.4 NACA 0012 unswept wing

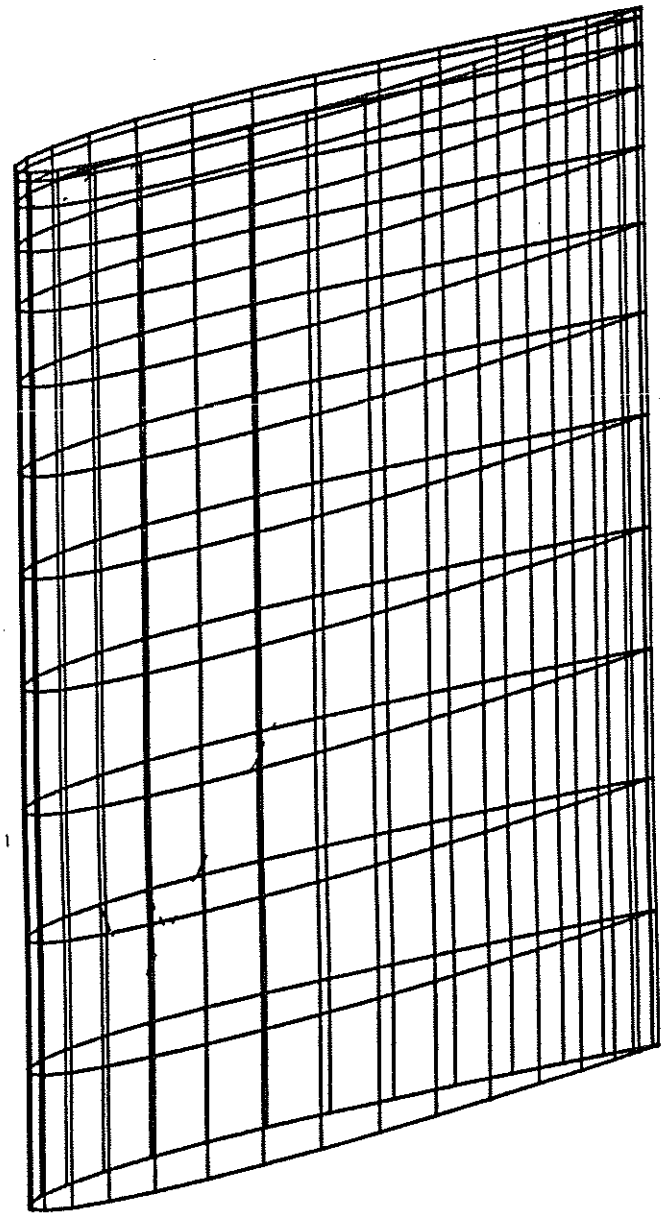


Figure 4.12 Isometric wire-frame plot of half span of NACA0012 unswept wing with AR = 5.95

To demonstrate the behaviour of the method for high aspect-ratio lifting surfaces a case was run for a NAC0012 wing with an effective aspect ratio of 5.9 at an incidence of 8°. Two panel arrangements of 50 by 8 and 33 by 12 chordwise by spanwise panels were tried. The panelling geometry of the later scheme is shown as an isometric wireframe plot in Figure 4.12. Again sinusoidal clustering of panels was used.

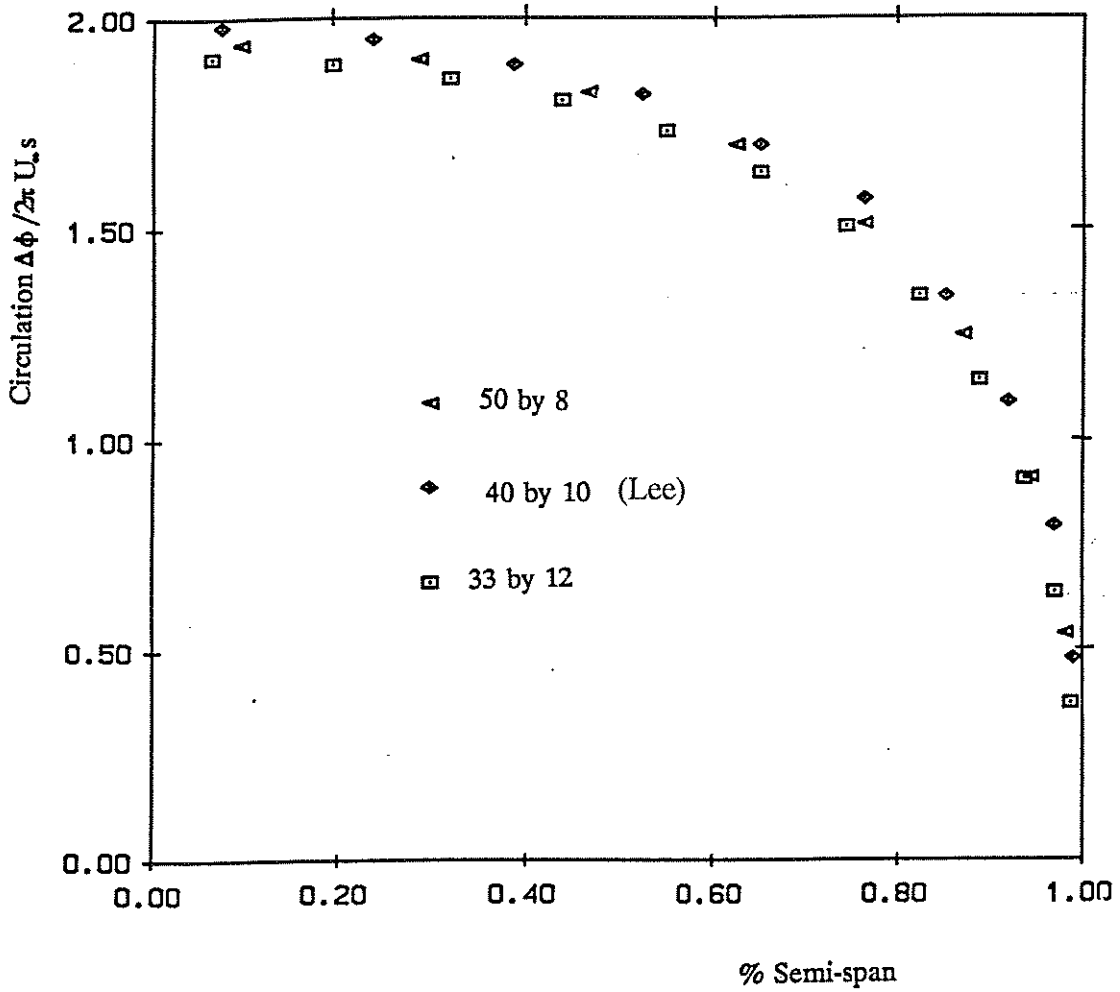


Figure 4.13 Spanwise circulation distribution for NACA0012 unswept wing at 8° incidence

In Figure 4.13 a comparison is made of the spanwise circulation distribution obtained by Lee and the two panelling arrangements tested. There is good agreement over the whole span. This agreement is also illustrated in Figure 4.14 which gives the chordwise pressure distribution at 64% semi-span. The area under the three sets of data is almost identical and close correspondence in both the leading and trailing edge

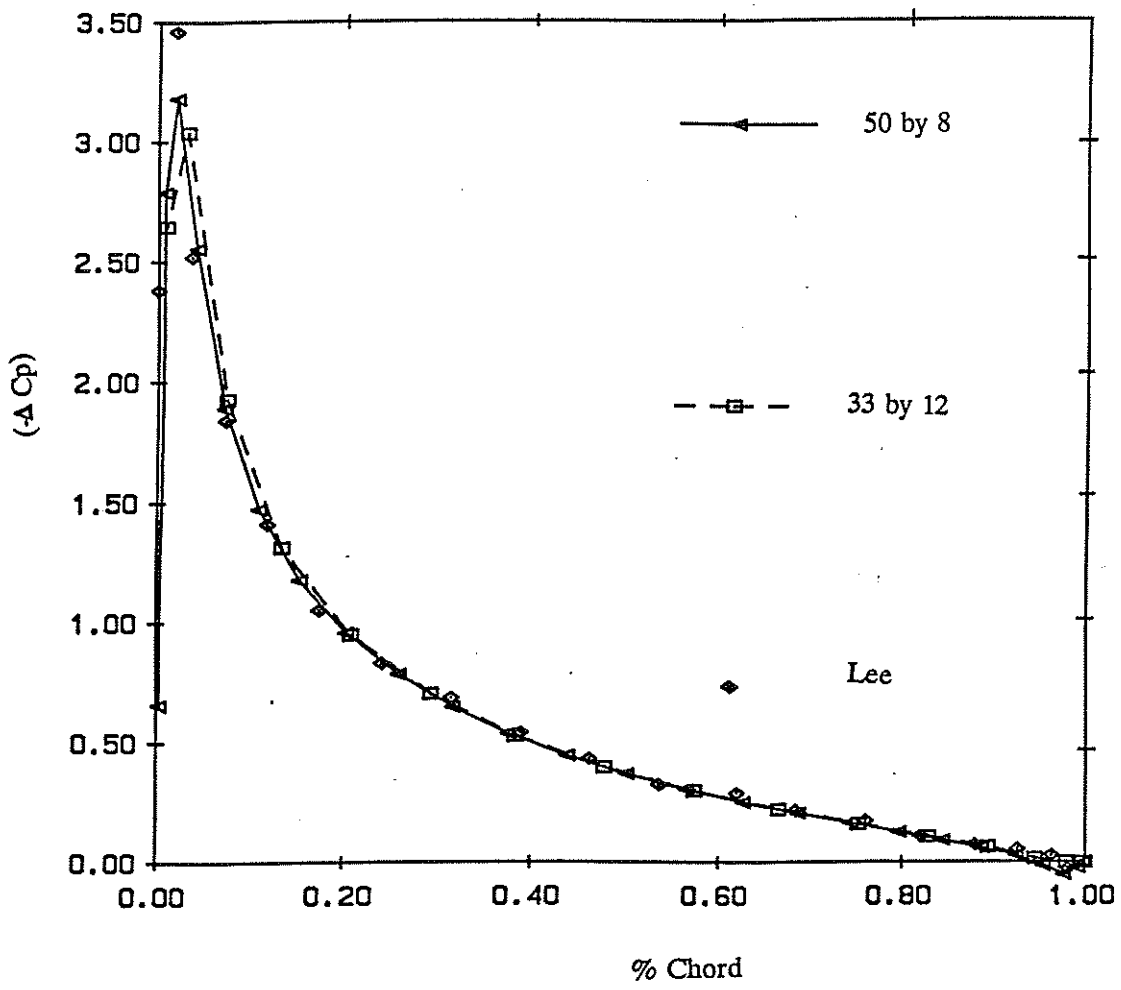


Figure 4.14 Chordwise pressure distribution for NACA0012 unswept wing at 8° incidence at 64% semi-span

regions.

A final comparison, Figure 4.15, demonstrates how the method predicts the experimentally measured spanwise local lift distribution of the wing. The lifting surface fails to pick up the slight drop in lift coefficient towards the wing centre-line (probably due to boundary layer effects) and the holding up of lift right at the tip (tip vortex effects see Molland[4.9]) but overall there is a very good correlation which indicates the suitability of the method for modelling both ship rudders and propellers.

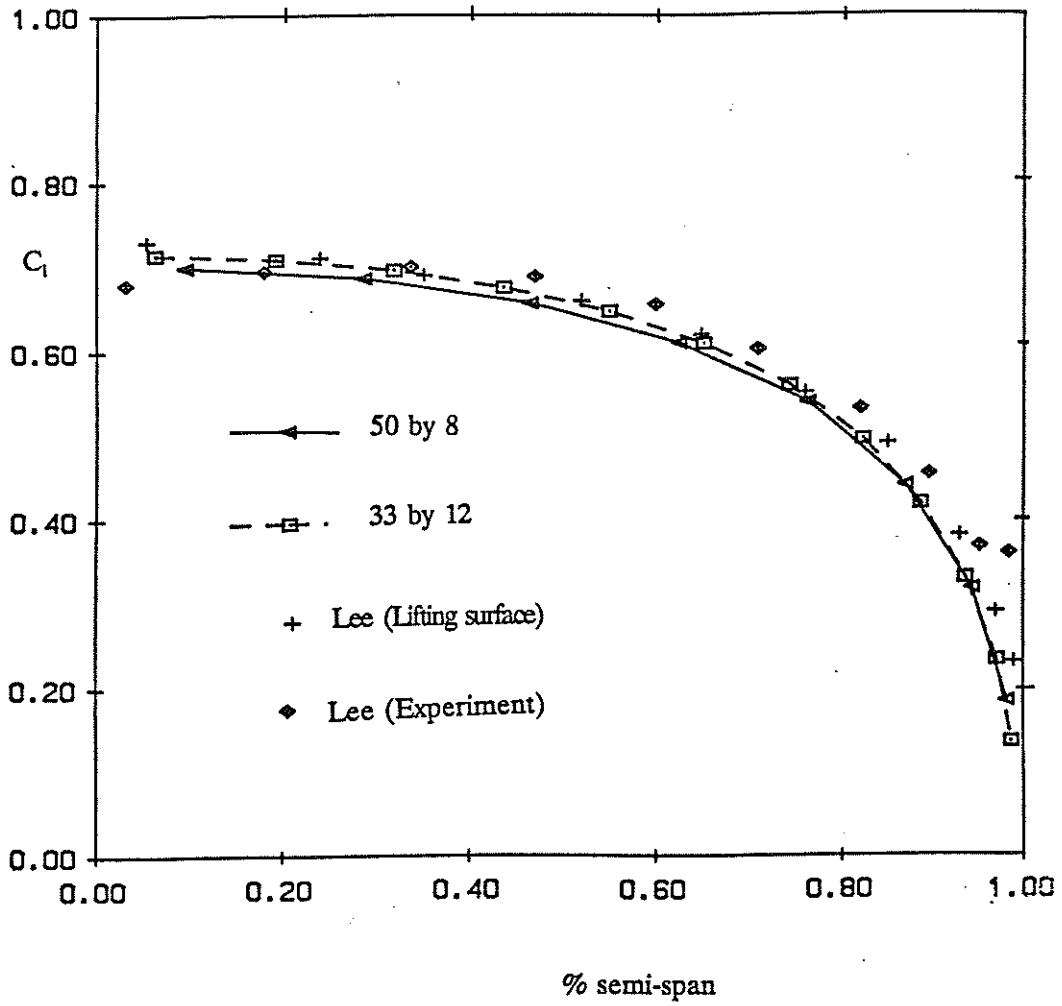


Figure 4.15 Local lift coefficient distribution for NACA0012 unswept wing at 8° incidence

4.11 Conclusion

A robust lifting surface panel scheme has been developed based on Lee's implementation of Morino's perturbation potential method. The procedure has been verified with a wide range of representative non-rotational lifting surfaces. The method is suitable for both rotational and non-rotational bodies.

The verification of the scheme numerics confirmed that the method is suitable for modelling both ship rudder geometries and ship propeller geometries.

An explicit trailing edge pressure Kutta condition was found essential to remove

spurious trailing edge pressure loading.

A method of surface definition using parametric cubic splines provides a flexible and rapid means of defining complex multi-body geometries.

Surface pressure/velocity distribution information can be obtained from the numerical solution and this allows total force and moment coefficients to be evaluated.

The formulation implemented allows three-dimensional velocity field information to be obtained. This can be used in defining the inflow velocity field necessary to implement the interaction velocity field required for modelling rudder-propeller interaction. The parallel implementation of the described theory is the subject of the next chapter.

CHAPTER FIVE

PARALLEL ALGORITHM FOR SURFACE PANEL ANALYSIS

5.1 Introduction

The implementation of the lifting surface panel method to run on an array of transputers using the developed communications harness is described in this chapter. A suite of procedures for carrying out the various stages of the analysis has been written and is referred to as the PALISUPAN (**PA**rallel **LI**fting **SU**rface **PAN**el) 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.

All the software was written in Occam2. Where necessary, important programming techniques have been included directly in the text. 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.

An interesting part of the work described in this chapter deals with techniques for the solution of simultaneous linear equations. In the context of a geometric parallelism over a square array of transputers both direct and iterative methods for solving matrices were investigated. The speed performance of these methods for different sizes of problems, numbers of transputers, and memory storage requirements were considered.

A final section gives the overall performance of the complete lifting-surface panel analysis.

5.2 Implementation of PALISUPAN on a Square Array of Transputers

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. This helps ensure an even computational load-balance between the transputers. 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.

The coarse-grained parallelism of transputers naturally lends itself to a geometric approach in computational fluid dynamics. This and the fact that it is the simplest approach to develop were the reasons for choosing geometric parallelism. It should be noted that the communications harness allows any topology of transputer network or parallelism strategy and this did not act as a restriction on the choice of method.

The use of a square array of transputers is attractive for matrix type operations which involve setting up and solving N simultaneous linear equations. As information about each panel has to be sent between all the transputers in the array the distance travelled by information (number of transputer links crossed) should be minimised and ideally a minimal diameter array should be used. The diameter of a transputer network is defined as the minimum number of transputer links which a message has to cross when travelling the shortest distance between two transputers. Work has been carried out to investigate possible topologies for minimal diameter networks. For example, Allwright[5.1] showed that a network of 41 transputers can be connected with no more than 3 links to be crossed for a communication between two transputers. At present, it is not a simple matter to make the software scaleable for changes in the number of transputers if the topology of the network changes as well. A square array of transputers which have spare links joined, as shown previously in Figure 3.1, to form a closed surface was considered a satisfactory approximation to a minimal diameter network. For this case the diameter of the network is equal to the number of

transputers along the side of the square network. It should be noted that on average the number of links most messages have to pass across will be less than this.

The choice of a square array of transputers restricts the number of transputers which can be used to 1, 4, 9, 16, and so on. However, the programming effort is significantly reduced. There will be less need for complex routeing strategies and the sub-division of panels and influence matrices can be directly mapped onto a square array of transputers.

5.3 Geometric algorithm for PALISUPAN

The data distribution and the numerical algorithm used geometric parallelism. The overall structure of the program was the same as that used for the successful implementation of the Euler solver. At the outermost level, the program consists of the harness and guest (or host) process running in parallel. The method of initialisation of the array of guest processes from the host process and of termination are identical to that described in Chapter 3.

The development of the complex lifting surface panel code requires the successful implementation of a number of separate stages. Each stage, for example the reading of the geometry input file, generating body panel co-ordinates, calculating influence coefficients and so on, has to be individually checked and verified. To aid this process it was decided to use a menu-driven structure for the lifting-surface program. Every stage is written as a separate process which only executes when chosen. The structure of the Host and Guest processes to achieve this were as follows:

```
HOST                              GUEST
SEQ                               SEQ
  ... initialise network            ... initialise network
  finito:=FALSE                     finito:=FALSE
  WHILE finito=FALSE                WHILE finito=FALSE
    SEQ                               SEQ
    ChooseMenuOption                  ReceiveMenuOption
    BroadcastOption                   ImplementOption
    ImplementOption                 ... Terminate
  ... Terminate
```

This allows each individual process or stage to be independently developed, checked, verified, and its performance on varying sized arrays of transputers to be measured. Keyboard input is used to choose from the range of options displayed. The Host process then broadcasts this option to all the Guest processes. A CASE statement is used to select the correct option of both Host and Guest processes once the option choice message has been received.

Once the program is fully developed the menu structure can be dispensed with and the program rewritten as a sequential series of processes. Figure 5.1 is a flow chart of the overall lifting-surface panel program. The various processes will be described in subsequent sections.

The scaling of a lifting-surface problem of a given total number of panels onto the available number of transputers is of interest. The program was developed to allow the performance of the program and its component processes to be measured for both:

1) variable sized square arrays of transputers;

2) different total number of body panels across fixed sized square arrays of transputers.

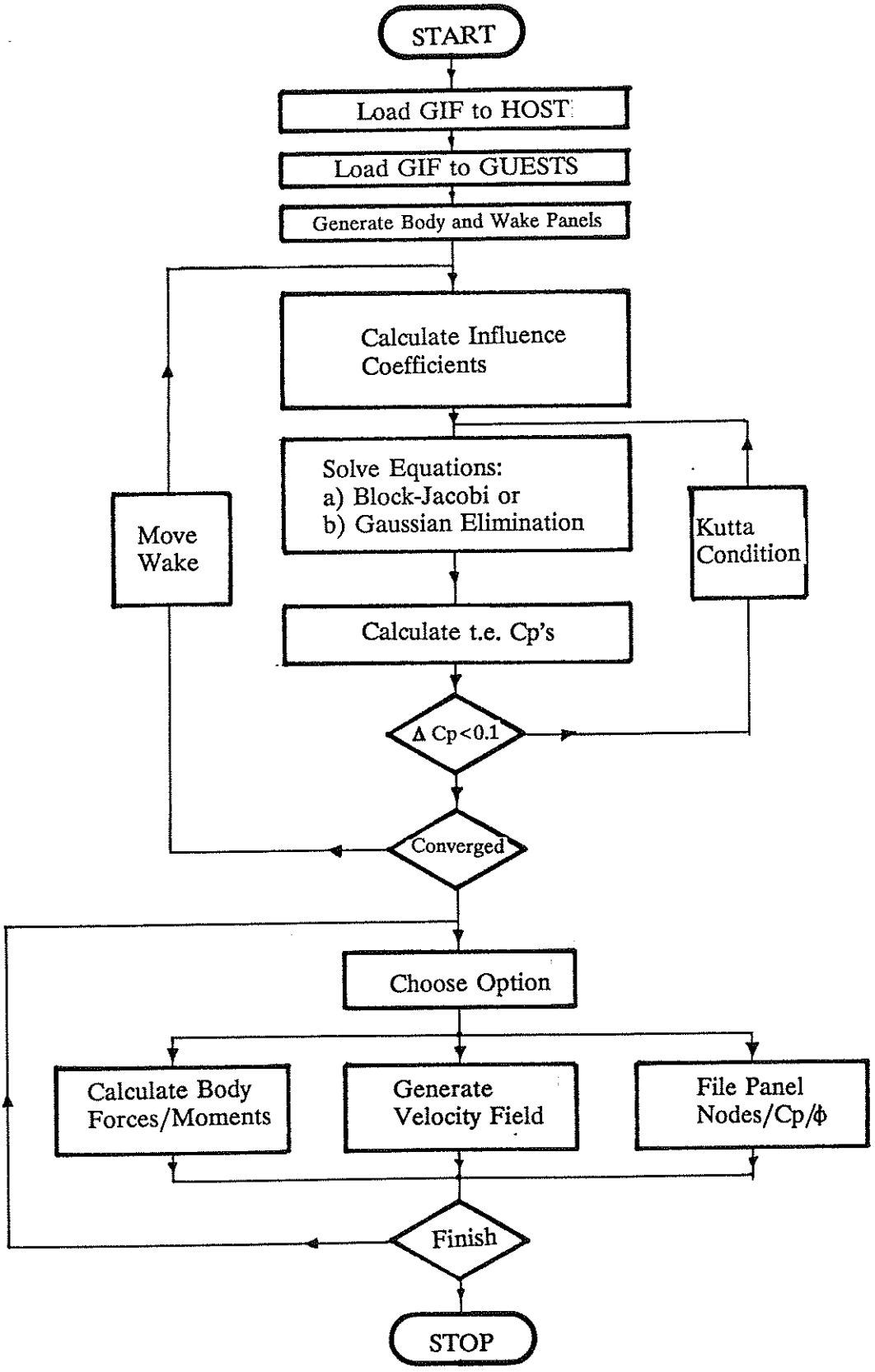


Figure 5.1 Flow chart of overall lifting surface algorithm

The total number of panels which are geometrically distributed across an array is limited by the amount of external memory assigned to each transputer. To maximise the total number of panels available the memory requirements of the component processes were kept to a minimum. In some places this resulted in extra numeric processing but it was considered that the greater the number of panels per transputer the greater the use of the code. A common restraint on panel method performance is a limit on the number of panels which can be processed at once. This block size severely limits performance as a large amount of slow data transfer operations from storage to active memory are required for processing each block. By removing the need for this type of operation a considerable time saving is achieved at the small cost of extra numeric processing.

As mentioned in Chapter 2, Occam2 TIMER channels allow the execution time of a process on a transputer to be measured in terms of an individual transputer's internal clock. Considerable amounts of detailed information can be obtained if timings are made on the individual guest processes. However, the inspection of the transputer clock involves a finite amount of time which, if too many inspections are made, can impair the code performance. It was decided that the performance of the component processes of the lifting-surface code would be measured only on the Host process. As each process starts and finishes by communication with the Host process the total execution time is easily obtained. Further detailed study of the performance of the component processes is of limited long-term interest and was therefore not considered important in this work.

5.4 Lifting Surface Definition

5.4.1 Geometric Data Distribution

The ease with which the geometry of a lifting-surface problem can be distributed across the array of transputers will determine the usefulness of the fluid dynamic code. As discussed in Chapter 4, it was decided in this work to use a scheme where each

lifting-surface body is defined in terms of series of lines of points. A parametric cubic spline procedure is then used to sub-divide the body surface into the required number of surface panels. As only a small number of points are used to initially define each body it was decided that every guest process (transputer) would be sent the complete problem geometry. From the overall geometry definition, each guest process generates its allocation of surface panels. As the parametric cubic splines used on each process are based on identical information the panel boundary points between transputers will be the same.

To ensure as even a load balance as possible and minimise delays each transputer is assigned, where possible, the same number of panels. In addition, as a considerable amount of numerical processing is required to calculate the wake strip influence coefficient, each transputer has the same number of wake strips. Delays may still occur as different schemes are used to calculate the panel-node influence coefficient in the near, mid and far field.

The method chosen to ensure an even load balance involved sub-dividing each body into streamwise panel strips with an equal number of streamwise strips assigned to each transputer. For example, this results in a two rudder problem as shown in Figure 5.2, where there are Nt streamwise panels by Ns spanwise panels for both rudders. Each transputer is assigned (Ns/T) strips of Nt panels for each rudder. This imposes a minimum limit of strips an individual body can be divided into, which is equal to the total number of transputers T (one strip per transputer). This is not considered a problem for moderate sized arrays of transputers ($T<25$).

Each surface panel is assigned an unique absolute identity number to allow the dipole, source and wake influence coefficient matrices to be correctly assembled. A unique body identity number is used to define the correct relationship between individual panels on a body which is necessary for the calculation of surface potential derivatives. By relating the panel identity number to the individual transputer number in the array, the process of setting up and solving the coefficient matrices is made

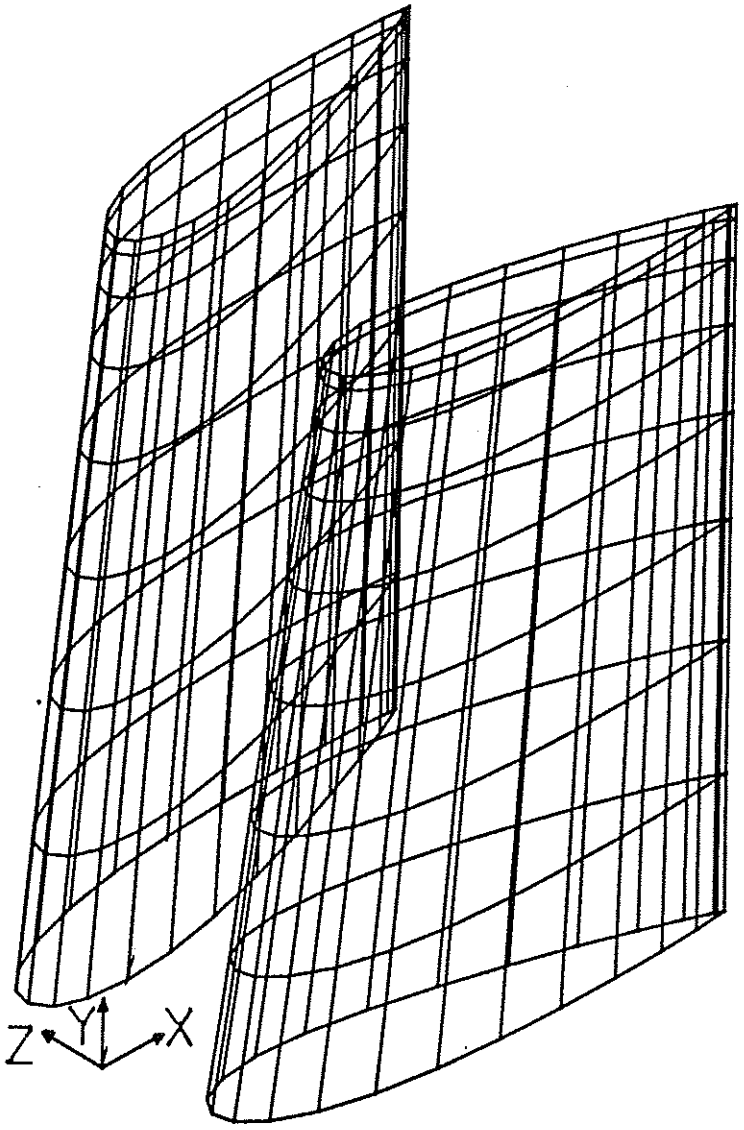


Figure 5.2 Panelling arrangement for two rudder arrangement

independent of a particular geometry. This allows multiple body problems to be easily solved.

In addition to the body panel data the velocity field and wake sheet data need to be sub-divided amongst the guest transputers. An identical method to that for the body is used for the wake sheet where the number of wake strips on each transputer are those corresponding to the body trailing-edges. As only a small amount of information is used to define a three-dimensional velocity field the complete field is

defined on every guest process. The means by which memory storage requirements are minimised are described in the next section.

5.4.2 Data Store

In conventional Fortran or Pascal programs body geometry data, such as panel vertices [x,y,z], are stored in multi-dimensional variable arrays. The maximum number of elements in each dimension is specified at compile-time. This results in the need to recompile the program every time it is wished to alter, for example the balance between the maximum number of spanwise Ns and chordwise Nt panels. Also, for multiple body problems, each body either requires its own assigned array or to be assigned as a part of an overall array. This sort of programming problem results in cumbersome detail with inherent costs in programming and debugging time. A more sensible strategy is to write an explicit indexing procedure and store all the information as a single-dimension array of real numbers. The index procedure allows the required coordinate information to be directly accessed. The maximum size of the array can be related to the memory available on a transputer and a far more flexible approach to multiple body problems becomes possible.

The geometrical panel definition is used throughout the lifting-surface panel algorithm. Therefore, all the component processes have to be able to use the information. In Fortran such information is transferred between sub-routines using common-block type definitions. Similarly in Occam2 and Pascal the information can be accessed if the individual procedures are defined within the scope of the array definition. Again, the larger the amount of array information the more cumbersome the programming detail. Fortunately in Occam2 the whole problem can be circumvented by defining "live" storage. A storage array such as that for the panel vertices is written as a process running in parallel with the numeric algorithm. Information is accessed through communication along channels in and out of the memory store process. Considerable numeric processing can be carried out within the memory store process running in parallel, thereby helping to ensure satisfactory

'hiding' of inter-transputer communications across the harness.

The basic design of the "live" memory store developed for this work uses a one-dimensional array with a simple index procedure for access, the details of which are given in Appendix E. A single input channel with a tagged protocol is used to determine what operation the store carries out. The input statement is enclosed in a WHILE loop which terminates when a 'finish' tag input occurs. A representative store process is as follows:

```
PROC Store(CHAN OF pan In,Out)
  [MaxDim]REAL32 StoreArray:
  BOOL finito:
  SEQ
    finito:=FALSE
    WHILE finito=FALSE
      SEQ
        In ? CASE
          Startup
            ... initialise
          finish
            finito:=TRUE
          etc
            ...
```

The use of such a structure for memory storage has resulted in many programming advantages. The development time of individual component processes of the lifting-surface program is greatly reduced as many repetitive programming tasks can be included within the Store procedure. For example, determining a panel's vertices and centroid can be obtained for a given absolute panel identity number. Once the tagged process within the Store has received the panel identity number it picks out the four vertices from the array and uses them to calculate the panel centroid. The Store process then outputs the four vertices and centroid.

Another advantage is that data storage requirements for the component processes can be kept to a minimum. As the communication time for information is rapid, and simply involves the passage of the address in memory where the information is stored, information can be individually processed without the need to hold duplicate copies of data.

The Store structure allows simple means of checking and debugging, through the use of tag protocols, for inspecting the contents of the store array. For example, using the GetCentroid tag on the In channel the Host process can ask all the Guest processes to send all their individual panel coordinate information back to the Host for storage. The resultant panel geometry file is then inspected to check the panelling of the lifting surface.

The basic store structure allowed different variants to be developed all using the same tagged protocol but with different functions occurring within each store. A panel store, wake store, and velocity field stores were all produced. These will be discussed in the sections after the description of the format of the geometry input file.

5.4.3 Geometry Input File

A declared aim for the lifting-surface method was the ability to define a three-dimensional velocity inflow field and multiple lifting-surface geometries in a straightforward manner. A single fixed format input file was used. The layout of the geometry input file (GIF) is given in Appendix F.

The generation of the input file itself can be either carried out through the use of an ASCII file text editor or by writing of special programs to use information from, for example, CAD systems to generate the required format.

5.4.4 Body Surface Definition

Process PanelStore was written to store the relevant panel information on an individual transputer. All the guest processes are sent a complete copy of the geometry input file.

On receipt of the information, PanelStore uses the geometry information to define the panels it has been assigned. The vertex generation is carried out using the parametric cubic spline algorithm described in Chapter 4. The relative spacing of the panel vertices is specified in the geometry input file. The following distributions are available and are illustrated in Figure 5.3 and the functions given in Appendix F.

The pivot, offset, scale, and angle transformations are carried out on the co-ordinates in PanelStore to obtain the final panel vertices in the overall cartesian co-ordinate system.

Once all the body surface panels have been defined across the array of transputers many different manipulations and inspections can be carried out. Only panel vertices are stored but, when needed, the centroid and panel unit normal vector are generated from the vertex information. The ordering of the panel vertices has to be consistent to ensure that all the body panel normals face outward from the body surface and into the exterior domain.

5.4.5 Wake Sheet Definition

A Wake Store process accepts the relevant information from the geometry input file. A fixed number of panels are distributed across the wake sheet in the same manner as the body surface panels. Once the wake sheet panels in the overall coordinate system have been generated, the location of the first (most upstream) panels are compared to their matching lifting surface trailing edge panels. If there is an offset the location of the whole wake sheet is translated so there is direct correspondence of

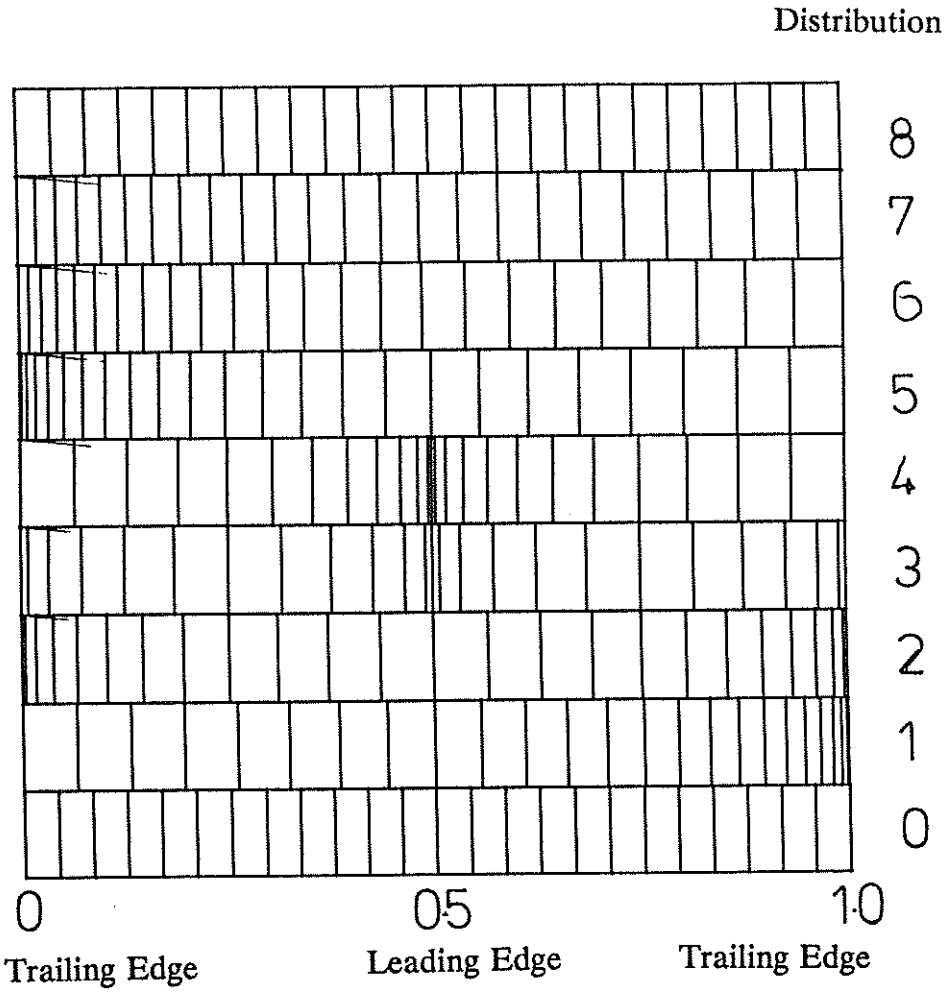


Figure 5.3 Possible panel size distributions

the wakesheet start and the body trailing edge.

5.4.6 Velocity Field

The inflow velocity field is defined on a regularly spaced three-dimensional mesh of points. It represents the velocity field in the absence of the lifting-surface geometry to be tested. A velocity store on each transputer stores the whole velocity field after input from the geometry input file. In the store the velocity is scaled by the reference velocity to give the velocity in S.I. units (metres/second). The principal function of the

store is to return, for a point anywhere within the domain of interest, the absolute inflow velocity at that point. A tri-directional linear interpolation is used to calculate the inflow velocity within the defined velocity mesh. External to the mesh the inflow velocity is set equal to the scale velocity in the freestream direction. Either a cylindrical or cubic three-dimensional grid can be used to define the velocity field. The choice is dependent on the application. For a circumferentially averaged inflow field less information is required to define the velocity field in cylindrical coordinates and this method is used for defining rudder-propeller interaction velocity fields.

5.4.7 Performance

The time for the geometry input file to be read, distributed to the array of transputers, and then the relevant store panels generated is a quick process. It is dependent on the number of bodies, number of sections for each body, number of wake sheets and the number of transputers. The lifting surface definition is a farm algorithm. The host process distributes the data packets and the individual guest processes reply on completion of the panel generation. The total time to distribute this information is proportional to the length of the geometry input file.

5.5 Calculation of Influence Coefficient

5.5.1 Introduction

The setting up of the lifting surface system of equations represented by matrix equation [4.28] requires the calculation of the source and dipole influence coefficients S_{ij}, D_{ij} between every panel (i=1 to N) and panel centroid (j=1 to N), and the wake strip influence coefficient W_{kj} between every wake strip (k=1 to NWs) and panel centroid (j=1 to N). As each guest process stores a fraction (N/T) of the total number of panels considerable communication around the transputer network is required to allow the elements of the influence coefficient matrices S_{ij}, D_{ij}, W_{kj} to be evaluated and stored.

The means by which information is transferred determines the efficiency of the setting-up process. The final results of the calculation are three matrices. The source and dipole matrices are of size N by N, where N is the total number of panels used to define the lifting surface. The wake influence matrix has dimensions NWs by N where NWs is the number of wake strips.

The boundary condition vector [**U.n**] at each panel centroid is known and hence each panel's source strength. So, rather than storing the elements of matrix $[S_{ij}]$ it is better to carry out the matrix-vector multiplication as part of the setting up process. This halves the required matrix memory storage.

To store the right hand side vector **RHS**, the dipole matrix $[D_{ij}]$ and wake strip matrix $[W_{kj}]$ three memory store variants were used. The dipole matrix is stored in a general square matrix routine which evenly distributes the matrix elements across the square array of transputers. This routine is used extensively in the solution of system of equations and is discussed in Section 5.6. Similarly, an equivalent general vector procedure is used for the vector **RHS** and a rectangular matrix for $[W_{kj}]$.

At this stage it is worth noting that, if there are N panels distributed across a network of T transputers, each transputer has information for (N/T) panels. The dipole matrix is of size N^2 and each transputer stores a sub-matrix of size (N^2/T). This indicates that as the dimension of the square array increases, for a fixed amount of memory per transputer, the maximum number of panels located on each transputer will decrease. This is due to the matrix store progressively occupying more of the transputers memory. To illustrate this Figure 5.4 plots the total number of panels distributed across the array of transputers against the individual store memory requirement for each transputer in the network. The lines plotted are for different sized transputer networks ranging from 1 to 121. The memory requirement (in bytes) for a panel, wake and matrix store is approximated by:

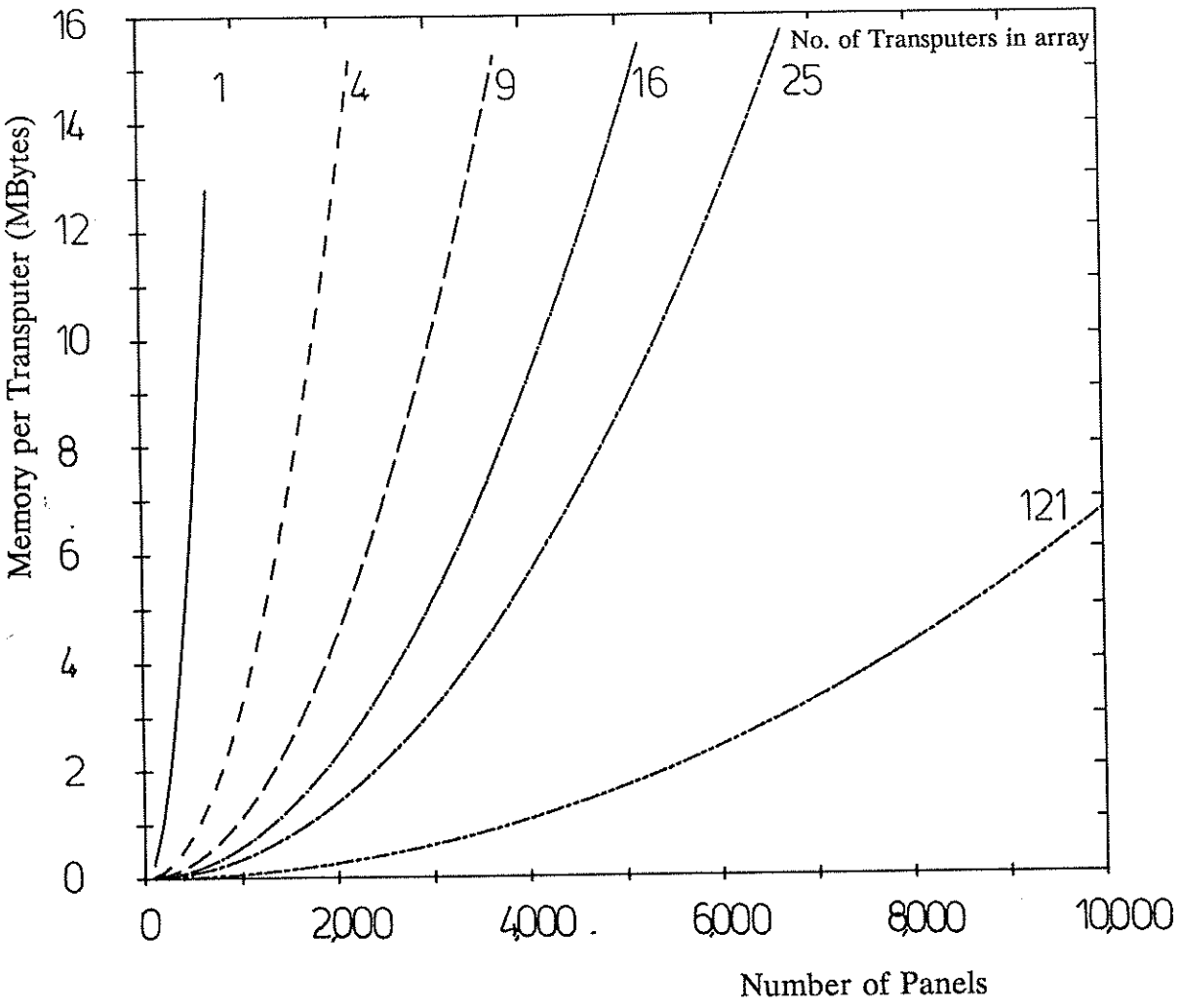


Figure 5.4 Individual transputer memory requirement for a given number of surface panels

$$M = 4 \times \left(3 \times 1.5 \times \left(\frac{N}{T} + 1 \right)^2 + 2 \times \left(\frac{N^2}{T} \right) \right) \qquad [5.1]$$

where each real number requires 4 bytes and all the memory requirements but that for the panel and dipole store arrays are ignored. This provides a good estimate of the number of panels which can be solved on a given transputer array. The different lines represent various sized arrays of transputers. For instance the Ship Science Transputer System, of dimension N=2 and 1 Mbyte of memory per transputer, could solve a 400 panel problem. A machine with 8 Mbyte per transputer could solve a 10,000 panel problem if it were of dimension 11. A surface panel problem containing this number

of panels is the order of the largest scale of lifting-surface problems currently solved on supercomputers. The Meiko Computing Surface available for use has 8MByte per transputer and with a sixteen transputer array could solve a 3800 panel problem.

The memory requirement for the dipole matrix emphasises the need to keep the memory demand of the lifting surface code to a minimum even at the expense of additional numeric calculations.

5.5.2 Strategy

The efficiency of the dipole influence matrix and **RHS** vector setting up process has to be as high as possible to ensure satisfactory operation of the lifting surface program on large dimension arrays solving large-scale panel problems. This means the strategy chosen has to keep the volume and amount of data communication to a minimum and ensure that every guest process always has information to process.

The even distribution of panels amongst the transputers helps to minimise communication. Each transputer has to calculate the self-influence of its own panels and their corresponding centroids. To calculate the influence of panels located on other transputers to its own panel centroids it is more sensible to communicate the panel centroid to the other transputers (three real numbers) rather than to receive the other transputers panel vertices (twelve real numbers).

For a particular centroid it is necessary to calculate the source and dipole influence coefficients of every panel and of every wake strip. This is ideal for a parallel algorithm. Simultaneously, the setting up process is:

(1) calculating panel influence coefficients;

(2) calculating wake strip influence coefficients;

(3) receiving panel centroids;

(4) communicating the resultant matrix and vector elements to their final store destination.

The multiplicity of parallel tasks should ensure that each guest process has minimal delays while waiting to carry out a numerical operation.

Each panel and corresponding centroid has a unique identity number related to the guest transputer on which it is stored. This allows an index address system to be used to ensure the result elements are returned to their correct destination.

As the communications harness performs better for larger messages, wherever possible the size of each message should be as close as possible to the maximum. Therefore, rather than a single centroid a packet of centroids is sent around the network. Each transputer on receiving the packet processes it with respect to its own panels and wake strip. For every panel located on a given transputer a packet of dipole and source influence coefficients are produced. The dipole packet is returned to the correct destination within the dipole matrix. The source packet elements are multiplied by the relevant source strength found from the boundary condition vector elements [**U.n**] and the result packet sent to the transputer on which part of the **RHS** vector is stored. Similarly, the packet of influence coefficients due to each wake strip is sent through to its destination in the WakePak Store.

Figure 5.5 illustrates the strategy for data flow through a network of four transputers for a packet of centroids originating from transputer 1. It should be noted that vectors are sub-divided between all transputers, each transputer having (N/T) elements and they are ordered in the same manner as the absolute panel identity numbers. The total amount of communication is related to the number of panels per transputer and the dimension of the transputer array. Increasing the dimension of a transputer array for a fixed total number of panels increases the amount of communication.

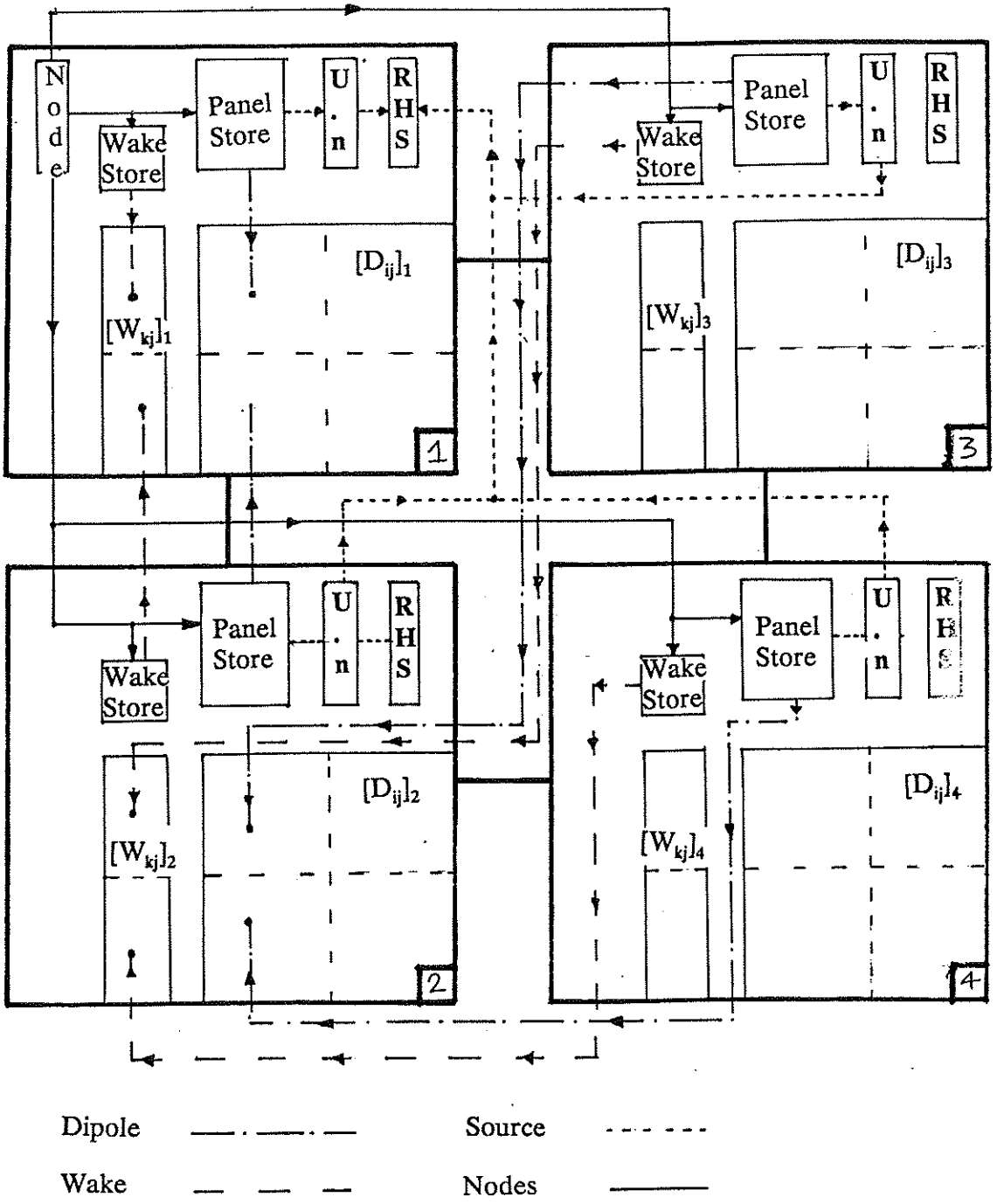


Figure 5.5 Data flow schematic for Calculate Influence coefficient process on a four transputer array

5.5.3 Structure

A number of processes are required to allow all the tasks to be carried out in parallel. Data flows in and out of the setting-up process through the harness and onto the rest of the transputer network. Also, information is passed in and out of the various live-memory stores and between the various numeric processes. As an Occam2 channel is a one-way link joining two parallel processes, the in and out channels of the harness and stores can only be connected into one of the parallel processes. Multiplexing and de-multiplexing constructs are therefore used to ensure, for example, that the centroid packets arriving at a transputer are not confused with a packet of dipole influence coefficients. Tags attached to each message packet allow this routeing requirement to be achieved.

At the outer level the setting-up process CalculateIC runs in parallel with the various live-memory stores and Harness:

```
PRI PAR
  Harness()
  --
  PAR
    Matrix32()
    PanelStore()
    WakeStore()
    SurfVect()
    WakeIC()
    CalculateIC()
```

with two channels connecting all the stores and Harness with CalculateIC. PanelStore is connected by 4 channels to allow centroids to be generated and panel influence coefficients to be processed. Figure 5.6 is a block data flow diagram showing how the various sub-processes of calculateIC are connected to each other and the stores and harness. The different types of line are used to represent the various data packets. The CalculateIC process is sub-divided in the following manner within a WHILE loop:

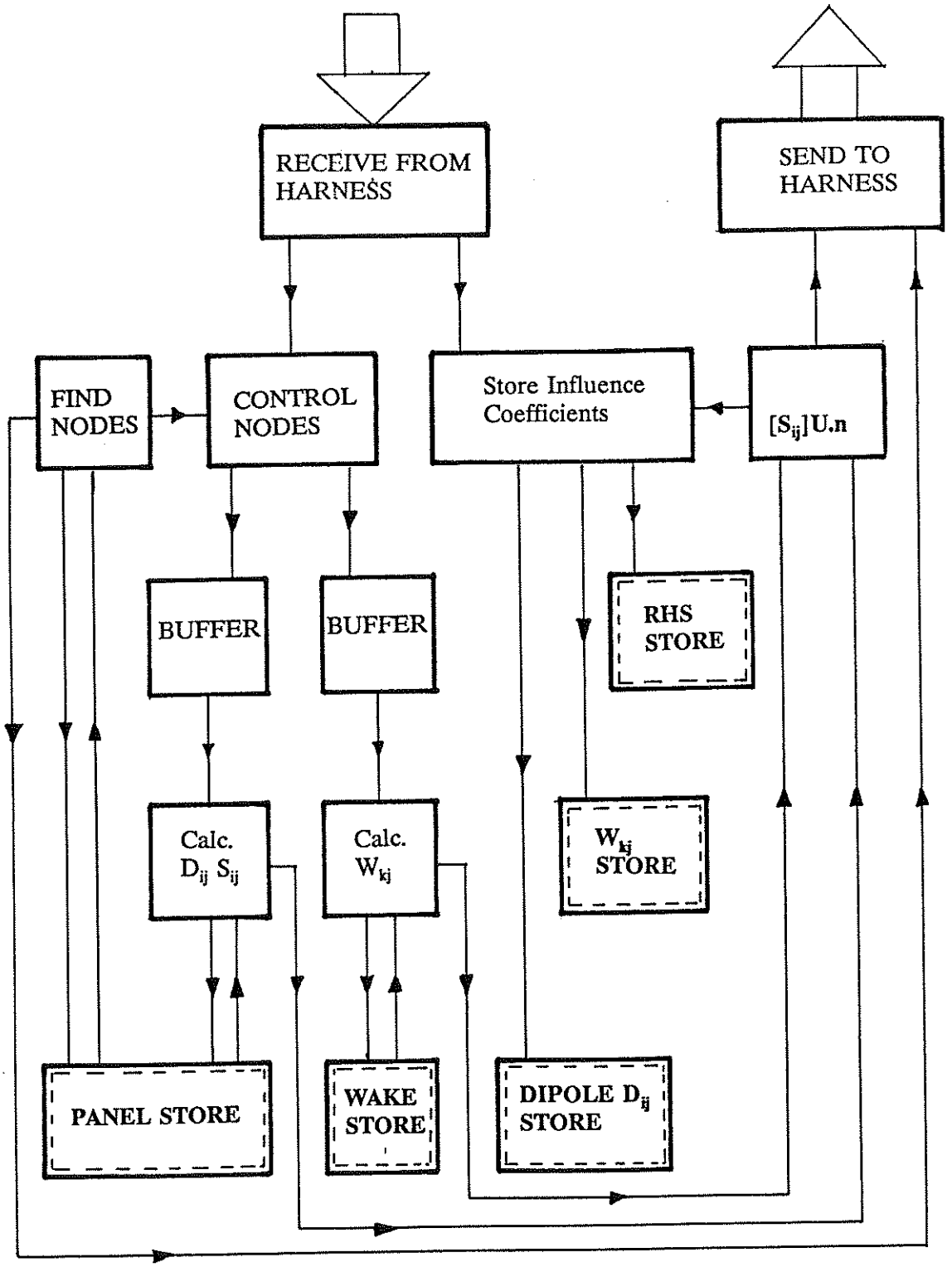


Figure 5.6 Internal structure of the calculate influence coefficient process

```
SEQ
  finito:=FALSE
  WHILE finito=FALSE
    PAR
      ... GenerateCentroids
      ... etc,etc
      --
      ... finish
```

The correct termination of CalculateIC via the finish sub-process uses the fact that a fixed number of each of the types of data packets will be sent for a given total number of panels and array dimension. Cumulative tallies are kept of the data elements arriving to be stored and similarly the data packets generated. This allows each of the sub-processes to successfully terminate and hence allows the termination of the overall process.

The various processes which control the influence coefficient element packets act as a buffer for passing the information into the live-memory stores. This clears the receive harness process to allow further input. The operation of the actual influence coefficient calculation processes are detailed in Sections 5.5.4 to 5.5.6.

5.5.4 Source and Dipole Influence Coefficients

The Newman panel influence coefficient procedure described in Chapter 4 calculates both the source and dipole influence coefficients. As input the procedure requires the test point and the panel vertices and geometric coefficients. To minimise data storage the panel geometric coefficients are recalculated every time a panel is used. By processing a package of centroids with each panel the number of times the geometric coefficients are recalculated is minimised.

For bodies with a plane of symmetry in the free-stream direction a reflection plane can be used to either halve the number of panels or double the panel density by

using the same number of panels. An example of this is flow around a finite aspect-ratio wing at incidence where flow about the mid-span is symmetrical. The location of the reflection plane is specified in the geometry input file. The influence of panels on the reflected body half on the actual panel centroid is accounted for by reflecting the panel vertices about the reflection plane and using the image panel with the centroid in the Newman panel routine. The symmetry condition means that a panel and its image will have the same value of surface potential and their influences can thus be summed.

An open-water propeller has rotational flow symmetry of order equal to the number of blades. Therefore, a multi-bladed propeller can be modelled by considering the flow around a single blade if the influence of all the image blade panels are calculated for the centroids of the panels on a single blade. Vector algebra allows all of the image panel vertices to be generated from the original panel vertices. The axis and origin of rotation are specified in the geometry input file. The reflection and rotational symmetry are also applied to the dipole panels defining the trailing wake sheet.

Within panel store on receipt of a packet of centroids for each panel the source and dipole influence coefficient are calculated for every centroid. If a reflection plane exists, using the reflected panel the additional image influence coefficients are added. Similarly if it is a rotationally symmetric body all the blade image panel influences are added. Two packets of source and dipole influence coefficients are generated for each panel. On calculating the sum of the actual, reflected and rotated influence panel for each centroid the two packets are sent out. Each centroid in a packet generates (2*N/T) influence coefficient packets.

5.5.5 Wake Strip Influence Coefficient

Each wake strip consists of a finite number of dipole panels all with the same strength. The total dipole influence coefficient at a particular centroid j is the sum of

the influence coefficients of all the panels in the strip k, that is:

$$W_{kj} = \sum_{I=1}^{N} W_{Ij} + W_{\infty\ kj} \qquad [5.2]$$

If required an expression for the additional influence of the wake sheet beyond its finite length to infinity W_∞ can be included at this stage. In the same way as for the body panels, reflected and rotated panel images are used to generate the final influence coefficient of each wake strip at a particular centroid. Again, each panel is used with all the centroids in a packet to save on unnecessary panel geometry recalculation.

5.5.6 Right-hand Side Vector

The Right-Hand-Side **RHS** vector is the matrix-vector product of the source influence coefficient matrix [Sij] and the surface boundary condition vector **U.n**. By calculating the RHS vector as part of CalculateIC it removes the need to store the complete source influence matrix. The saving in memory more than compensates for the additional programming complexity.

The solid surface boundary condition for a panel source strength σ is:

$$\sigma = - U_i \,.\, n \qquad [5.3]$$

where $\mathbf{U}_i$ is the inflow velocity at the panel centroid and **n** the panel unit surface normal. The inflow velocity is the sum of the specified velocity field $\mathbf{U}_s$ in the absence of the lifting-surface and any body rotational velocity. That is:

$$U_i = U_s + r \times \omega \qquad [5.4]$$

where **r** is the distance from the panel centroid to the origin of the axis of rotation and $\boldsymbol{\omega}$ the body rotation vector. The source strength at every panel is calculated during the lifting-surface definition process and stored in a vector store.

To obtain the final matrix-vector product each influence coefficient element has to be multiplied by the corresponding panel source strength. Each element of the RHS vector is the sum of the product of source strength and influence coefficient for a particular centroid.

$$RHS_i = \sum_{j=1}^{N} S_{ij} \cdot \sigma_j \qquad [5.5]$$

5.5.7 Performance

For a given number of surface and wake panels on a fixed array of transputers the total amount of communication remains constant. Therefore, the least efficient operation of the CalculateIC process occurs when there is a minimum of numeric processing per panel to be carried out. In Figure 5.7, which plots the number of panels distributed across four transputers against the time to calculate all the influence coefficients, lines are shown for three test cases of a rudder without reflection plane, a rudder with reflection plane and a four-bladed propeller. As the amount of numeric processing increases as the number of image panels increases the overall time increases. However, if the time to run a 200 panel problem on four transputers and one transputer is compared speed-ups of 2.96 (η_c=73.9%), 2.99 (η_c=74.8%) and 3.28 (η_c= 82%) are obtained for the three cases which demonstrates the increased efficiency with additional numeric processing per panel. The relatively small increase in efficiency is due to the additional operations to reflect/rotate panels and generate their geometry coefficients.

Overall the calculate influence coefficient process exhibits a quadratic increase in processing time with the number of panels. The knee in the curve occurs when the panel centroid is sent as two packets rather than one which results in extra processing as each panel geometry data is generated twice as many times.

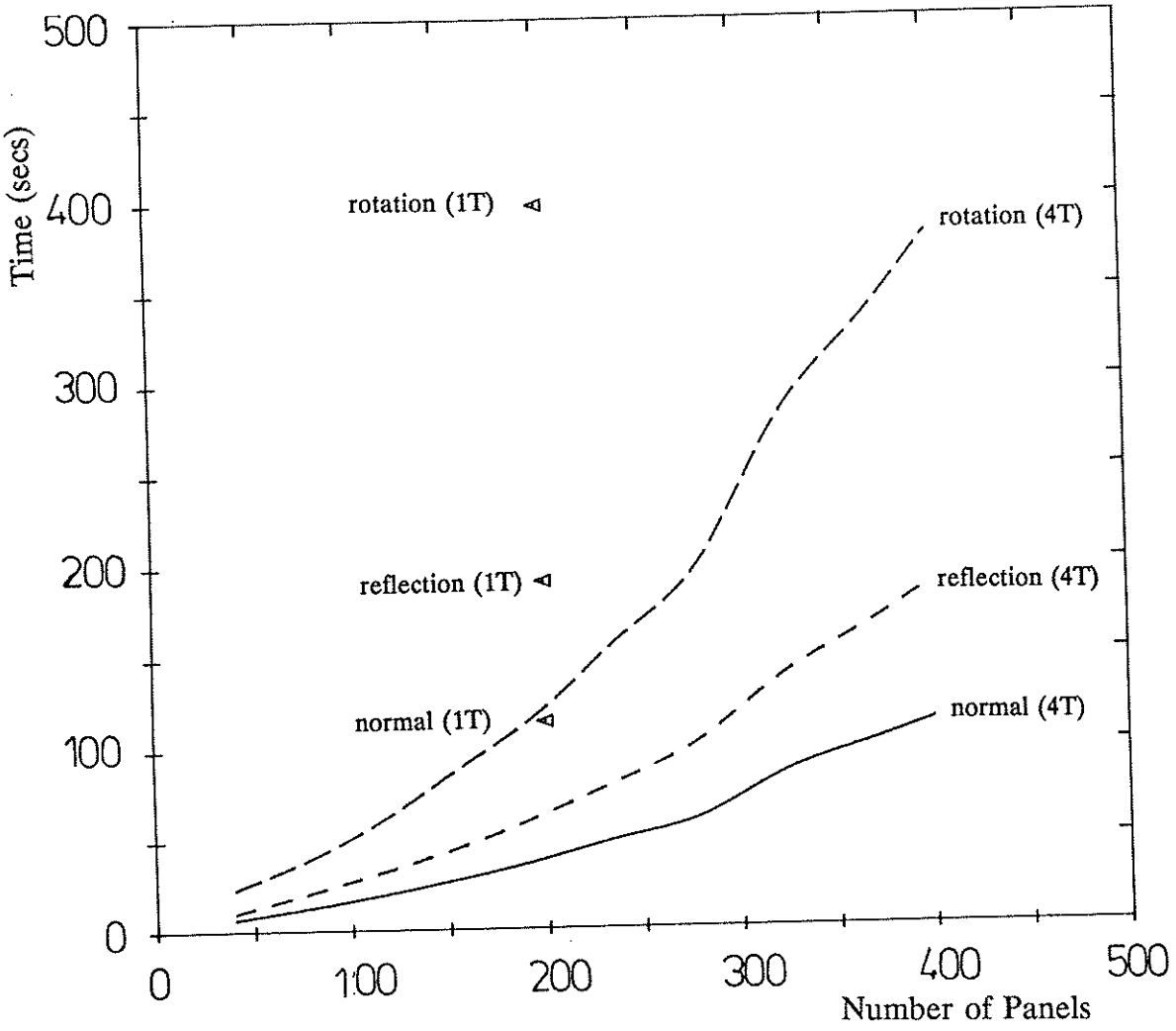


Figure 5.7 Performance of calculate influence coefficient on a four transputer array

5.6 Solution of Simultaneous Linear Equations

5.6.1 Introduction

Efficient solution of the linear system of equations generated by the Calculate influence process is essential. The basic system to be solved is:

$$[\,M\,]\,\underline{\Phi} = \underline{R} \qquad [5.6]$$

where vector **R** is known , matrix [M] is dense but diagonally dominant, and vector $\boldsymbol{\phi}$ is unknown.

There are large numbers of techniques for solving linear systems of equations on sequential computers. Also, as it is a technique of general application to many numerical problems, parallel algorithms are being developed. For instance, Rua[5.2] investigated the similar problem of solving for eigenvectors on an array of transputers using a linear chain topology. An aim of this work has been to investigate the use of transputers for solving general C.F.D. type problems. Therefore, an investigation was carried out into methods for solving systems of linear equations when the matrix data is distributed across a square array of transputers.

Several methods both iterative and direct were implemented and performance comparisons have been made. To aid in the implementation a series of basic matrix-matrix and matrix-vector processes were developed.

5.6.2 Matrix Procedures

By developing processes which use a Matrix live-memory store to carry out basic matrix operations such as addition, subtraction and multiplication more complex processes can be rapidly written as combinations of the basic operations. The live-memory store aids in reducing development time by allowing easy access to individual elements, rows and columns. Also, initial values for the matrix such as the identity or zero matrix can be carried out within the store.

Matrix addition and subtraction operations are achieved by linking three parallel processes:

PAR

Matrix32(Msize,*Ain,Aout*)

--

Matrix32(Msize,*Bin,Bout*)

--

MatrixAdd(Msize,*Ain,Aout,Bin,Bout*)

The input/output channels (*Ain,Aout,Bin,Bout*) of the two Matrix32 store processes are connected to the MatrixAdd process. This process, for every element in the sub-matrices of size Msize, obtains corresponding elements from [A] and [B] matrices, adds them, and then stores the value in Matrix [A]. No communication is required between guest transputers and all the host process need do is to start and then acknowledge the finish of all the guest processes. An identical technique is used for such things as matrix subtraction[5.7], equality[5.8], and scalar multiplication[5.9]:

$$[A] = [A] - [B] \qquad [5.7]$$

$$[A] = [B] \qquad [5.8]$$

$$[A] = k \times [B] \qquad [5.9]$$

where k is a scalar constant. For all the above operations the guest processes are independent of each other and the code efficiency will be solely affected by the start and acknowledge finish communication across the whole network. Figure 5.8 shows the time to carry out a matrix add operation against a base of overall matrix dimension Msize for a four transputer array. The process is rapid and exhibits a quadratic behaviour as the number of operations increases as a square of the matrix dimension. Figure 5.8 also shows the time to carry out the other matrix operations described later.

Matrix multiplication and obtaining the transpose of a matrix:

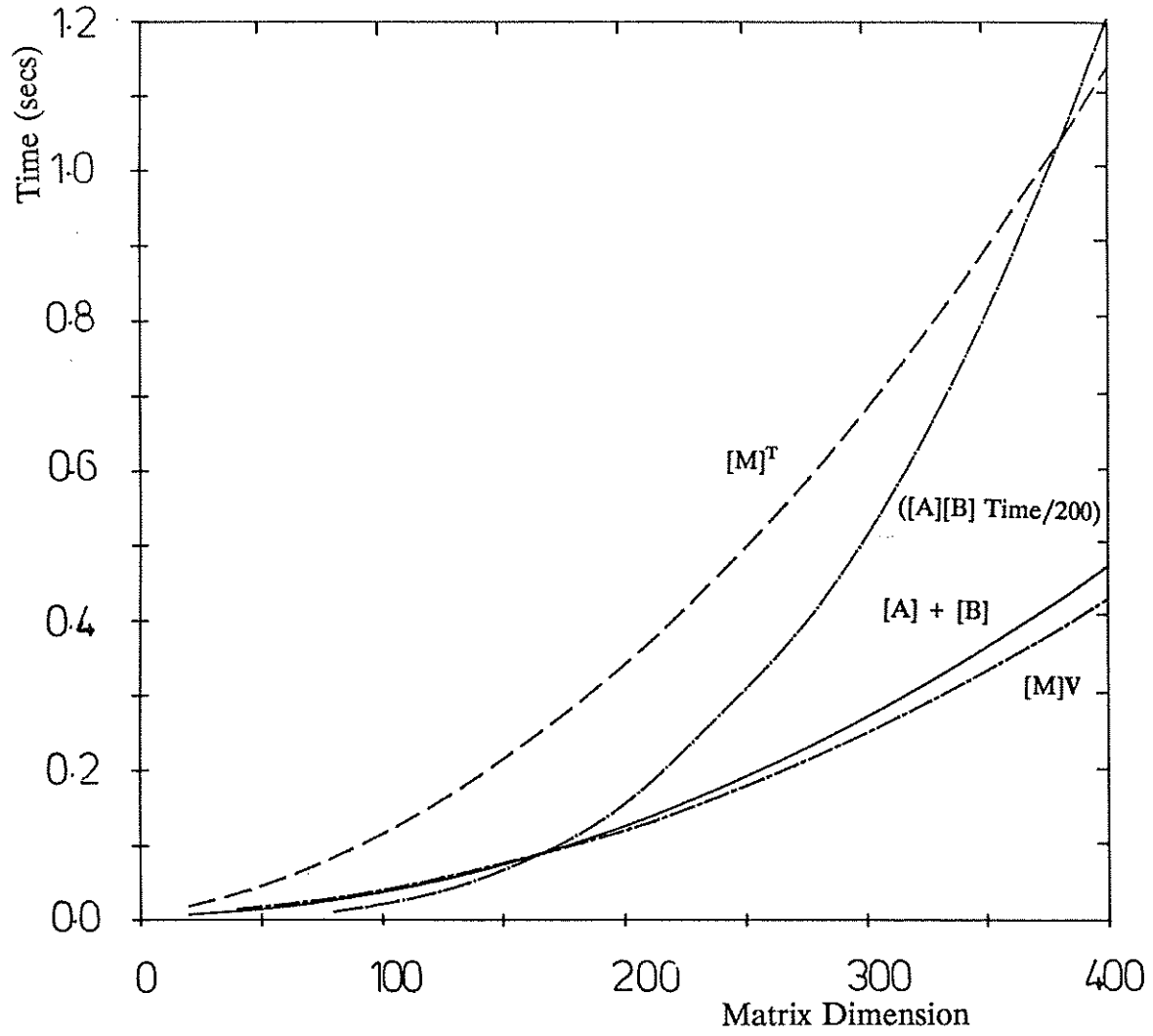


Figure 5.8 Time to carry out basic matrix operations on a four transputer array

$$[\ A\ ] = [\ B\ ]^{T} \qquad [5.10]$$

$$[\ A\ ] = [\ A\ ][\ B\ ] \qquad [5.11]$$

requires the interaction of individual elements with many others and hence inter-guest process communication is needed. To achieve maximum efficiency a minimum amount of communication is required and no guest process should ever be waiting for information to process.

The transposing of a matrix involves the swopping of rows and columns. The process is schematically shown in Figure 5.9 with matrices [A] and [B] sub-divided

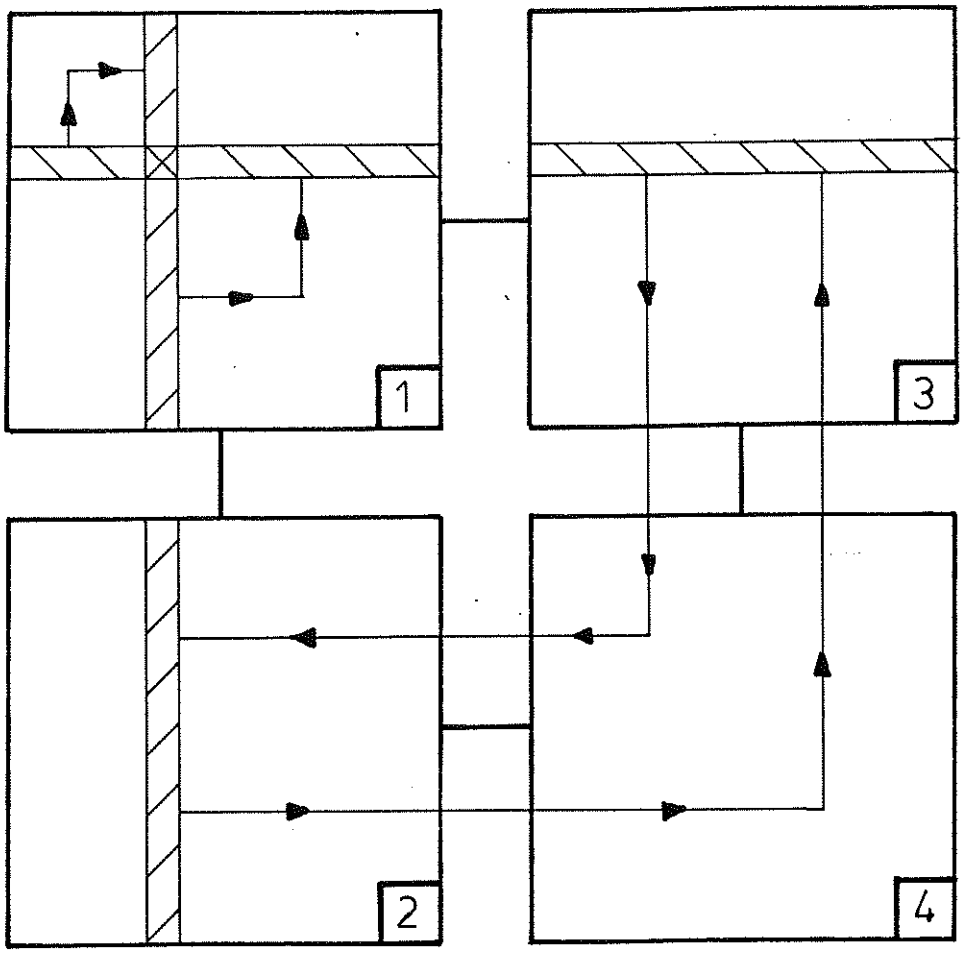


Figure 5.9 Schematic of matrix transposition on a four transputer array

across 4 transputers. For instance, row 10 of matrix [B] becomes column 10 of matrix [A]. In the example of the 4 transputer network, the diagonal transputers T1 and T3 simply take rows from [A] and put them in the corresponding column where as the off-diagonal transputers T2 and T4 swop rows and then store as columns. The parallel algorithm is written as:

PAR

... **SendRow**

... **ReceiveColumn**

... **GetRows**(*Bin,Bout*)

... **StoreColumns**(*Ain,Aout*)

where if the transputer is on the main diagonal each retrieved row is sent directly to the StoreColumn process otherwise it is passed across the harness through the SendRow process. At its destination it is received by ReceiveRow and then passed to StoreColumn. A matrix transpose simply involves communication. Figure 5.8 shows the time to carry out a transpose against a base of overall matrix size for a four transputer array, again a quadratic behaviour is seen.

Matrix-vector multiplication

$$[\,M\,]\,\underline{V} = \underline{Q} \qquad [5.12]$$

requires the summation of the multiplication of each element of a matrix row with the corresponding element of the vector. That is:

$$Q_i = \sum_{j=1}^{Msize} M_{ji}.V_j \qquad [5.13]$$

A vector is stored as a column through the transputer network, and as shown is schematically in Figure 5.10. For a 4 transputer network each guest process sends its part of the vector to all the transputers in its column in the array. This sub-row is summed with the corresponding elements in each of the rows and then the part answer vector sent back to its correct location. Each transputer carries out the same amount of communication which is proportional to the dimension of the transputer array. Figure 5.8 also gives the time to carry out matrix-vector multiplication for different sized matrices. It can be seen that the performance is similar to the matrix add operation; the number of operations being equivalent and the communication low.

Although matrix-matrix multiplication is not required as part of a solution procedure it is none the less an interesting test for a parallel computer. As every element in each matrix has to be multiplied together it requires a large amount of communication for a bare minimum of numeric processing. The approach chosen was to first transpose one matrix. Then every column in one matrix is passed horizontally through the array in a similar manner to that used in the matrix-vector multiplication and the result vectors are sent to their final destination and summed to give the result

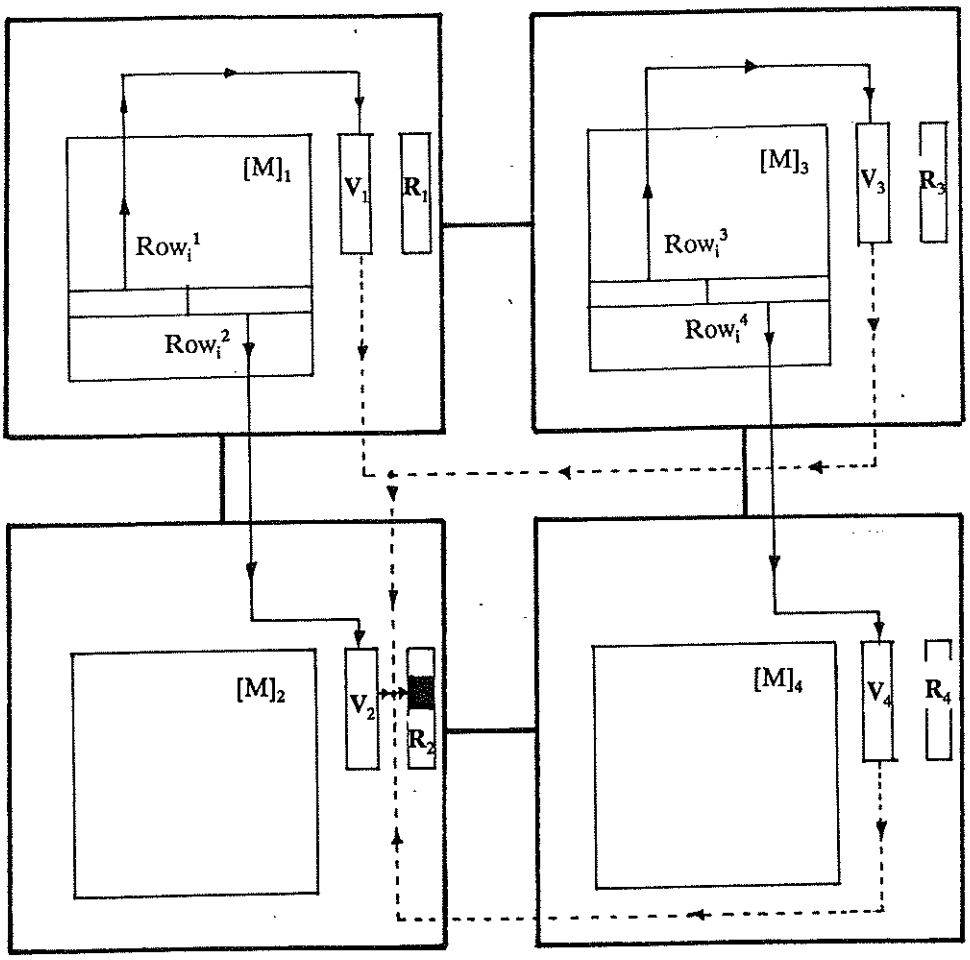


Figure 5.10 Schematic of matrix-vector multiplication on four transputers

matrix. The time to carry out this operation, as also shown in Figure 5.8 (to a scale of time(secs)/200), is about 2/3rds of the time taken to transpose a matrix and then carry out N matrix-vector multiplications. The difference is due to the reduced amount of control commands from the Host transputer. Overall, matrix-matrix multiplication exhibits a cubic behaviour.

5.6.3 Iterative versus Direct methods

The choice of technique for solving a system of linear equations revolves around whether an iterative approach can converge more rapidly to solution than a direct method takes to explicitly solve the whole system. In general a dense full matrix, such

as that generated by a lifting-surface, is more likely to favour an iterative approach. This is especially true because of the leading diagonal dominance of the system. This dominance arises from the self-influence of each dipole panel.

How the various algorithms are made to run in parallel and whether this influences the choice between iterative and direct methods needed to be investigated. A range of iterative methods were developed and their performance compared with a direct matrix inversion using Gaussian elimination.

5.6.4 Gaussian Elimination - Direct Inversion

The inverse of a matrix is defined such that the product of the matrix and its inverse gives the identity matrix

$$[\,I\,] = [\,M\,]^{-1}[\,M\,] \qquad [5.14]$$

If both sides of equation [5.6] are multiplied by $[M]^{-1}$ to give [5.15].

$$[\,M\,]^{-1}[\,M\,]\,\underline{\Phi} = [\,M\,]^{-1}\,\underline{R} \qquad [5.15]$$

Φ is obtained directly as a matrix-vector product shown in [5.16].

$$\underline{\Phi} = [\,M\,]^{-1}\,\underline{R} \qquad [5.16]$$

A linear system of equations can be solved by first creating the lower diagonal form of [M] and by then progressive back-substitution to successively determine each element of Φ. However, this process has to be repeated every time the right hand side vector **R** is changed. This happens when the iterative Kutta condition is applied. Therefore, it was decided to concentrate on generating the complete inverse and the application of the resultant Kutta condition to vector **R** then requires only a single matrix-vector multiplication.

Gaussian elimination operates by successively carrying out row operations on both the original and an identity matrix to remove elements from the original.

Comprehensive treatment of the method is given in textbooks, for example Kreysizg[5.3]. At first sight Gaussian elimination is not ideal for a parallel algorithm as the method works by progressively eliminating columns of elements.

The elements in a column are eliminated about a pivotal element. For a given pivot element a_i of row j the following two-stage process is carried out, where r is a matrix element:

```
(1)   SEQ k = 1 FOR N
        r_jk := r_jk / a_i
```

and where b_i is the element of a row in the same column as the pivot element a_i

```
(2)   SEQ j = 1 FOR N
        IF
          i=j
            SKIP
          TRUE
            SEQ k = 1 FOR N
              r_ik = r_ik - ( b_i r_jk )
```

The result of the process is that all the elements of the pivot column are set to 0 apart from the pivot element which is unity. Both stages are carried out on the identity matrix using element a_i and b_i from the original matrix. The column elimination process is repeated for all columns to generate the inverse matrix. The choice of the individual pivot element can influence the accuracy of solution especially for large matrices. Also, the number of element operations is of the order (N^3) and so the time to solution increases rapidly as the matrix size increases.

The most accurate method of choosing the pivot element is so called maximum pivoting where at each stage the maximum (absolute size) element of the original matrix is chosen and used as a pivot. However, for diagonally dominant matrices,

where the diagonals are of the same magnitude, the best choice is simply to use the main diagonal as the pivot elements. Main diagonal pivoting was used for the lifting-surface system but maximum pivoting was successfully implemented by adding a stage to obtain the maximum element on each guest pivot. This information was collected by the host process and the position of the overall maximum element broadcast back to the guests.

A sequential structure was used for the guest process as a series of N loops for pivoting on each column. For each loop the host process broadcasts the location of the pivot. Each guest then carries out steps dependent on where it is relative to the pivot.

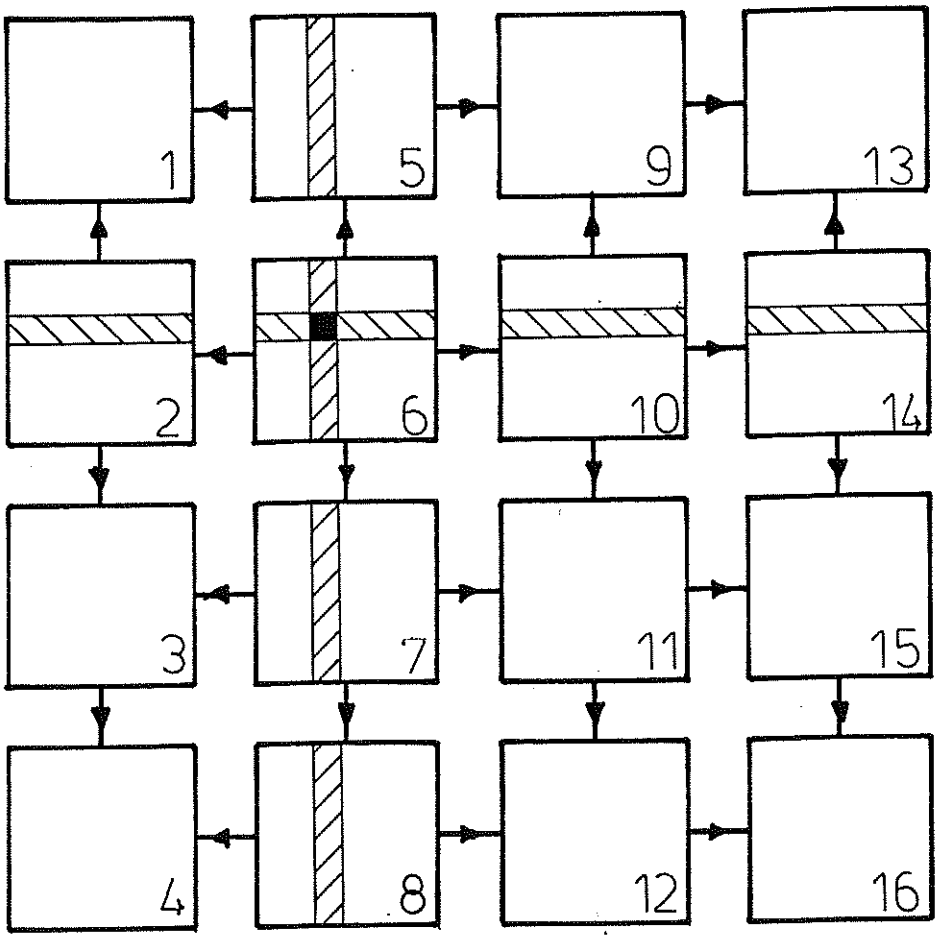


Figure 5.11 Schematic of Gaussian elimination on a sixteen transputer array

This is best illustrated in Figure 5.11 for a 16 transputer array. For a pivot on transputer T6 the steps that each transputer carries out are as follows:

T6:	Divides Pivot row by Pivot element
	Sends Pivot column left & right
	Sends Pivot row up & down
T2,T10,T14:	Receive Pivot Column
	Divide row by Pivot Element
	Send Pivot Row up and down
	Carry out row operations
T5,T7,T8:	Send Pivot Column left and right
	Receive Pivot row
	Send Pivot row
	Carry out Row Operations
T1,T3,T9,T11,	Receive Pivot Row
T12,T13,T15,T16:	Receive Pivot Column
	Carry out Row Operations

As rapidly as possible each guest process passes out information required by other processes. This minimises the delay while processes await information. The process is only shown for the original matrix but the pivot rows and columns are also passed for the eventual inverse matrix.

Two matrix stores are used for the inversion. At the completion of the process the original matrix is the identity matrix and the other is the required inverse of the original.

5.6.5 Jacobi-Iterative Scheme

An iterative scheme uses an initial guess for the unknown vector ϕ to generate a better approximation to the solution. The process is repeated until the solution has converged to a given level of accuracy. The simplest possible method for diagonally dominant matrices is to use the main diagonal elements to calculate the correction. This process is known as the Jacobi method.

The Jacobi correction $\Delta\phi$ to the original estimate of vector ϕ can be written as:

$$R^* = [\ M\ ]\ \phi^{k-1} \qquad [5.17]$$

$$\Delta\phi_{jj} = \frac{1}{M_{jj}}\left(\ R_j - R^*_j\ \right) \qquad [5.18]$$

where M_{jj} is the j^{th} leading diagonal element. The k^{th} approximation is then becomes:

$$\underline{\phi}^k = \underline{\Delta\phi}^k + \underline{\phi}^{k-1} \qquad [5.19]$$

If each guest process is sent the relevant part of the matrix main diagonal, then the Jacobi scheme can be implemented using standard matrix procedures:-

```
SEQ
  Matrix_Vector_Multiply (Mϕ,R')
  Vector_Subtract(R,R', Ans)
  Vector_Divide(Ans, Mjj, Δϕ)
  Vector_Add(Δϕ, ϕ)
```

A convergence check is carried out on correction vector $\Delta\phi$. The maximum absolute value is sent to the host process which either allows the guest process to proceed or, if all the maxima are below a cut-off criteria, finishes the calculation. The cut-off criteria used was that the absolute biggest change in an element is less than $1x10^{-5}$.

The matrix-vector multiply process is the only one which requires communication during each iteration. As a result each iteration is fast. However, as only one element is used in the updating process a large number of iterations are required. An advantage of the scheme is that all the processes are working all the time in generating the next iteration. The Gauss-Seidel scheme uses the same technique for updating, only each element update is carried out sequentially, based on using the latest values for

each element. The interdependence for element updating means that a parallel single-element Gauss-Seidel method is inefficient.

For the Jacobi scheme the initial approximation is usually taken as the zero vector. However, for the lifting-surface method with the iterative Kutta condition it is better to use the final solution for the next series of iterations as most values of potential only change by a small amount.

5.6.6 Single-Block Iterative Scheme

The drawback of the Jacobi scheme of using only one element in the updating process suggests the use of a block of information from the original matrix. A block size equal to the number of panels on each guest transputer as illustrated in Figure 5.12 can be easily accommodated.

The first stage is to transfer the leading diagonal matrices to the guest process where their self-influence panels are stored. These sub-matrices are then directly inverted using Gaussian elimination. The iterative process can then commence. It is identical to the Jacobi except that instead of the Vector Divide stage the sub-matrix inverse is used to multiply the vector difference **R-R'**. That is:

$$\underline{\Delta\Phi^{k}}_{T} = [\ M\ ]_{TT}^{-1}\left(\underline{R}_{T} - \left(\ [\ M\ ]\underline{\Phi}^{k-1}\right)_{T}\right) \qquad [5.20]$$

where T is the transputer number and $[M]_{TT}^{-1}$ the sub-matrix inverse. The implementation of the scheme is identical in all other respects to the Jacobi. A drawback to this scheme is that if the number of panels per transputer gets too large the inversion of the sub-matrix will take a long time. As in the case of the Jacobi scheme for each iteration the only communication is during the matrix-vector multiply stage. Also, once the sub-matrices have been passed to their destination, the direct inversion is an independent process.

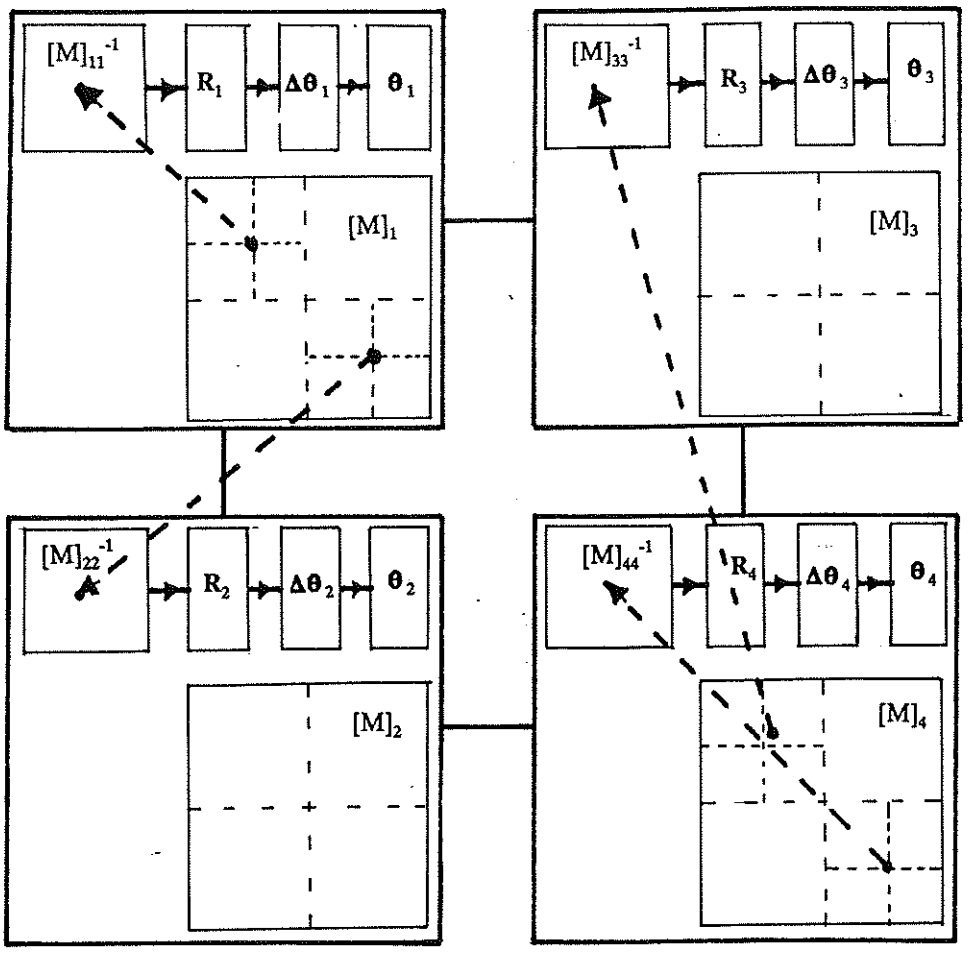


Figure 5.12 Schematic of Single block iterative scheme on four transputers

5.6.7 Multi-Block Jacobi Scheme

To restrict the maximum block size the dimension of the sub-matrices was made variable between 1 (Jacobi - single element) and that for 1 block per transputer (single-block). This is also shown schematically in Figure 5.12. The implementation is the same as for the single-block scheme except that each guest process has a number of blocks sent to it. Each block is then inverted and used in the iteration process to update its portion of $\Delta\phi$ vector. The ability to change the block size allows the iterative procedure to be tuned to obtain minimum iteration time for a given lifting-surface problem.

5.6.8 Comparative Performance

Figure 5.13 shows the time in seconds to solve a system of linear equations against the size of the system. Plots are shown for the four schemes previously described and also the time to set up the influence coefficient matrix. The data given is for a four-transputer system.

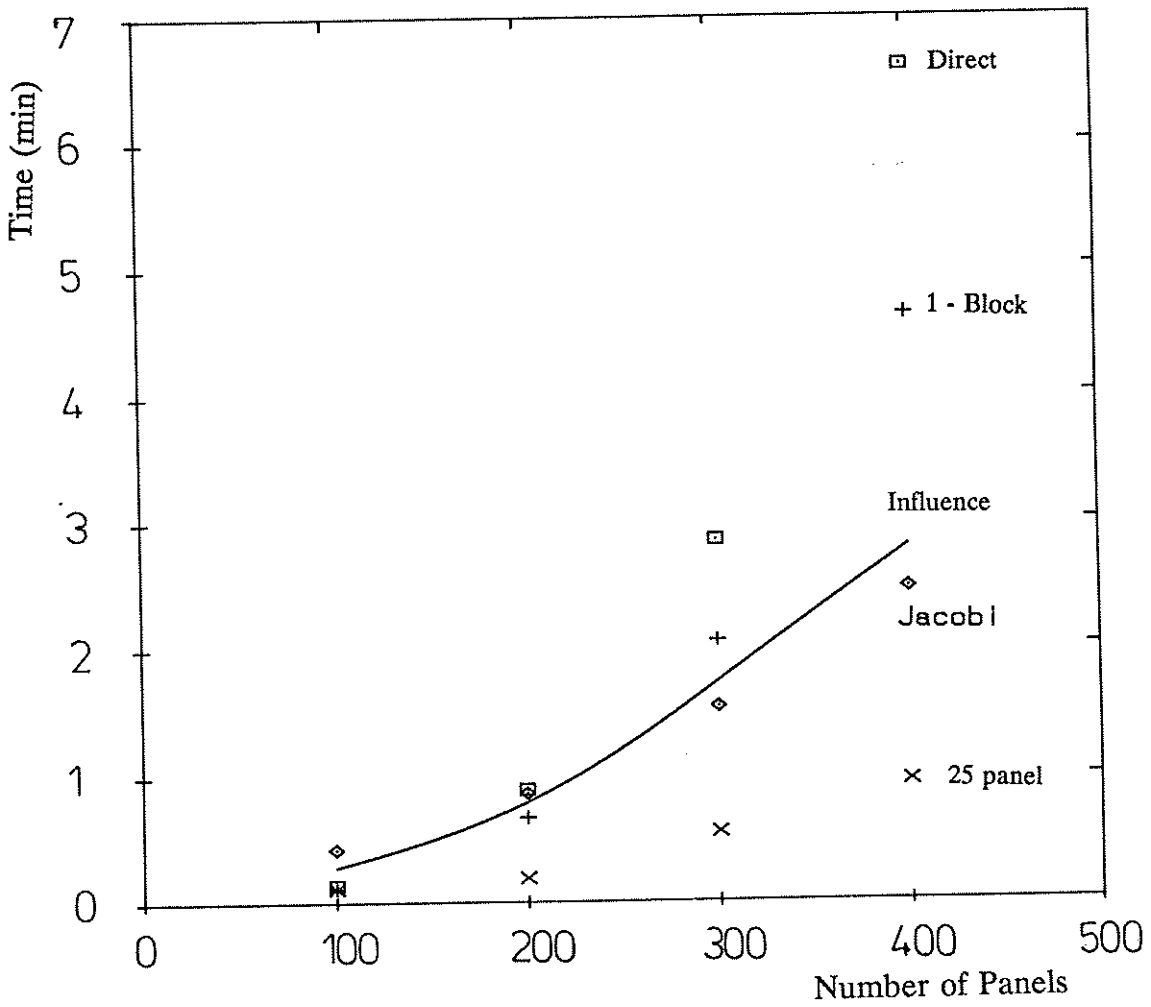


Figure 5.13 Performance of matrix solution procedures on a four transputer array

The $O(N^3)$ behaviour of the direct inversion scheme can be clearly seen. Also, exhibiting a similar trend is the single-block Jacobi scheme as the individual block size increases. The single element Jacobi performs much better than the direct scheme for matrix dimensions greater than 220. This is to be expected as the scheme converges in O(N) iterations and the time for each iteration has a fairly constant relationship to the total problem size. For the particular problem in question the Jacobi scheme takes about the same time as the influence coefficient process. The multi-block scheme used a constant block size of 25 panels and for the 400 panel problem was 7.2 times faster than the direct method and 2.7 times faster than the Jacobi method. It is worth noting that for this geometry the multi-block scheme is significantly faster than the influence coefficient set up process.

The reason for the choice of block size of 25 panels for the multi-block can be seen in Figure 5.14. This plots convergence time against block size for the 400 panel problem. A pronounced minimum is found for a block size of 25. This corresponds to the number of panels in each chordwise strip for the problem which was held constant for producing Figure 5.13. This is not entirely unexpected as the panel potential strengths will generally be dominated by other panels in the same chordwise strip. Therefore, only using the influence of these panels in updating should give the quickest convergence. Clark[5.4] also found this was the best block size for his accelerated convergence scheme. This scheme was used by Kerwin & Lee[5.5]. However, at present it has not been investigated as it contains substantial sequential processing components and requires extra memory storage.

5.7 Kutta Condition

The iterative Kutta condition was implemented in a straight forward manner. For the first iteration the Morino condition that $\Delta\phi$ was equal to the difference in potential at the trailing edge was used. After the potential vector ϕ had been solved either by a direct or iterative method, the pressure loading Δ Cp and the rate of change of pressure loading with wake sheet strength $d(\Delta Cp)/d(\Delta\phi)$ were found for all the wake

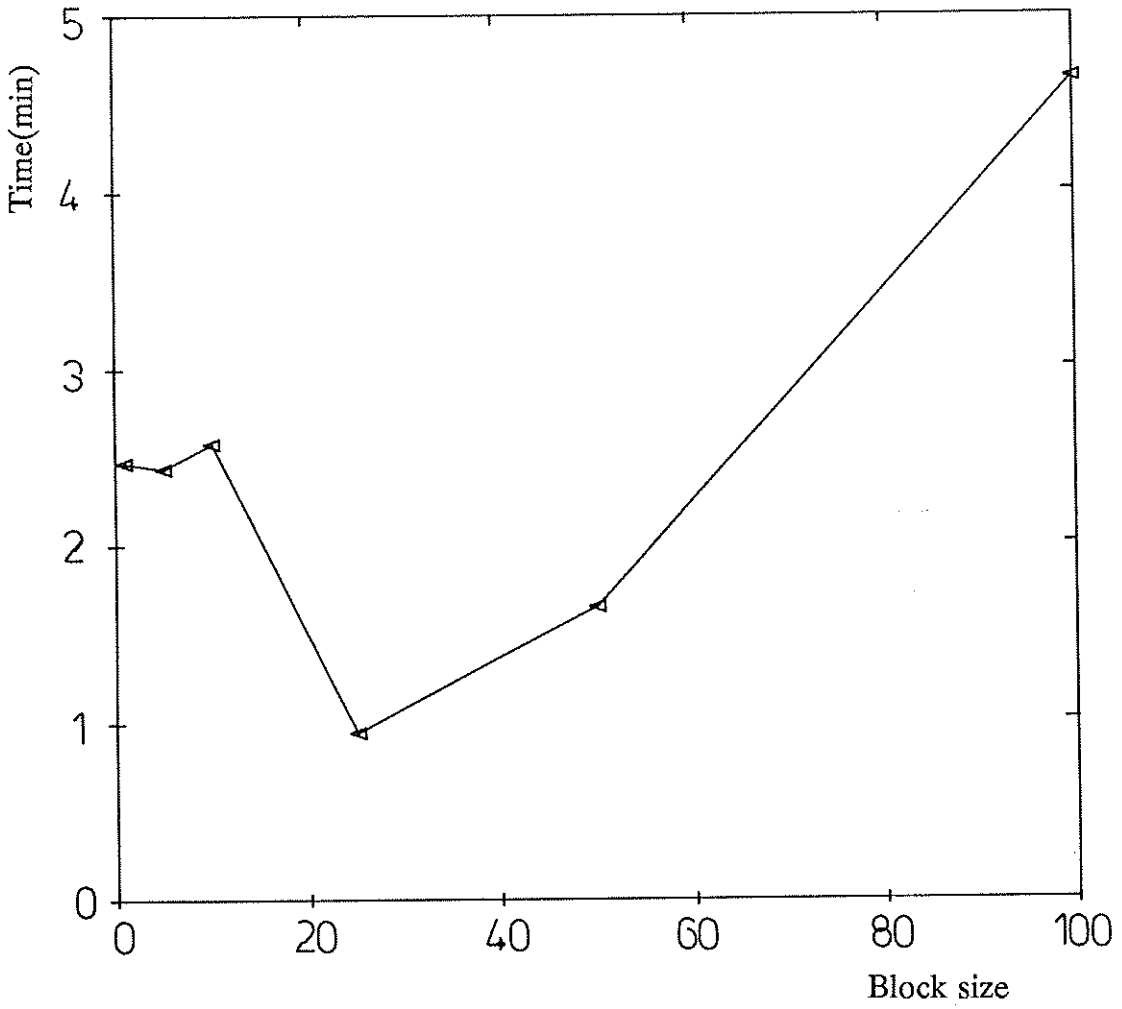


Figure 5.14 Variation in convergence time with block size for 400 panel problem

strips on each guest process. These values give a corrected wake strength $\Delta\phi'$ for each wake strip. Multiplying by the wake strip influence matrix $[W_{jk}]$ gives a correction vector to be applied to the right hand side vector:

$$\underline{R'} = \underline{R} + [W_{jk}]\ \underline{\Delta\phi'} \qquad [5.21]$$

A vector is used to store the wake strength $\Delta\phi$ as :

$$\underline{\Delta\phi} = \left(\Phi_u - \Phi\right)^k \qquad [5.22]$$

where ϕ_u and ϕ_l are the trailing-edge panels potentials. The process is repeated until Δ Cp has disappeared within a given limit. The total wake strength correction is:

$$\underline{\Delta\phi} = \sum_{k=1}^{m} \underline{\Delta\phi'}_k \qquad [5.23]$$

where m is the number of iterations to convergence.

The calculation of vector **R'** is rapid and performance figures are not given.

For the direct method of solution each Kutta condition iteration requires only a matrix-vector multiplication.

$$\underline{\Phi}^k = [\ M\ ]^{-1}\ \underline{R'}^k \qquad [5.24]$$

Whereas, for the iterative solution methods the number of iterations to convergence generally halves after each Kutta condition iteration. As noted previously, the initial approximation is taken to be the previous final solution for all but k=0.

5.8 Calculation of Velocity Field

In modelling the interaction of a ship rudder and propeller a requirement is the development of velocity field both up and downstream from a lifting surface. Differentiation of the expression for the Newman dipole and source panel influence coefficients gives the disturbance velocity at a point in space due to a unit strength panel. Once the values for the individual dipole panel strengths ϕ and wake sheet $\Delta\phi$ have been solved the total disturbance velocity at a point due to the whole lifting-surface can be found.

The total velocity U_T at a point is the vector sum of the disturbance velocity U_d and the local inflow velocity U_i.

$$\underline{U}_T = \underline{U}_i + \underline{U}_d \qquad [5.25]$$

The velocity field within a volume of space can be found by determining the total velocity on a regular mesh of points within the volume of space. This information can be used for various post-processing tasks. For example, many three-dimensional flow visualisation techniques require a regular mesh of data points. Also, the interaction velocity field can be developed from such an arrangement.

A farm algorithm controlled by the Host process is ideal for carrying out the velocity field generation process. The user specifies the x, y, and z limits of the volume of space and the required number of points Nx, Ny, and Nz in each direction within that space. A packet of points is sent to all guest processes. Each guest calculates, for every point in the packet, the disturbance velocity field due to its body surface and wake sheet panels. Also, it finds the inflow velocity field. The two packets of velocity information are returned to the Host process. The total velocity at a point is the sum of all the guest disturbance velocity field plus the inflow velocity at that point:

$$\underline{U}_T = \sum_{i=1}^{T} \underline{U}_d + \underline{U}_i \qquad [5.26]$$

where T is the number of guest transputers. Packets are sent from the Host until the velocity has been found for all points defined by the user.

As it is a farm algorithm the performance of the velocity field process is proportional to the number of panels and, the number of points, and inversely proportional to the total number of guest transputers.

5.9 Adaptive Wake

The method described in Chapter 4 was straight-forward to implement using a parallel algorithm similar to that for calculating the influence coefficients. Each wake strip is divided into a near and far region. In the near region the panels are aligned with the local flow direction. For the far wake, the panels follow the direction of the most downstream free wake panel. The first stage of the algorithm requires the generation of the centroids of the free wake panels. The total velocity is then

calculated at each centroid. Each packet of centroids is broadcast around the transputer array and the result packets returned to the originating Guest transputer. For the rotational case, the velocity is obtained in cylindrical coordinates, (V_a, V_r, and ω). The velocity is stored for each panel node by averaging the values of velocity at the surrounding four panel centroids. This summing process requires an exchange of information between transputers as identical nodes are held on different transputers. Once the information has been exchanged the remaining two processes are independent. For each node the change in position is obtained using the magnitude of the total velocity and length between stream-wise nodes to determine the timestep Δt. The increments in node position are then applied, using the regression relationships[4.33], to set the new node position. For the rotational case the increments are applied in the cylindrical coordinate system and then transformed back into cartesian system for storage in the wake store.

The performance of the wake adaption process is the same as for the velocity field generation.

5.10 Performance of Implicit Algorithms on an Array of Transputers

5.10.1 Introduction

Implicit algorithms require communication and the amount of communication traffic depends on the number of transputers used to solve a given problem. The estimation of the performance of an implicit algorithm on different sized arrays of transputers requires expressions for the time to carry out the communication transfer as well as for the numeric calculation. For an explicit algorithm, as represented by the Euler solver described in Chapter 3, the ratio of calculation to communication is constant for a given number of nodes per transputer and hence performance can be scaled from timing measurements made on a small array of transputers. However, for an implicit algorithm, how the communication time scales with array size has to be

known for an estimation process based on the performance of a small array of transputers.

As the performance of the Harness across an array of transputers is known an estimate can be made of the total communication time for any given array of transputers. In general, there will be one or more transputers in the network whose messages (to and from) have to travel over the number of links equal to the dimension of the array. This transputer will have the maximum data transmission time and determine the code efficiency.

This section firstly details the code efficiencies of the various processes which make up the lifting-surface panel method. An estimation procedure is developed using the code efficiency measurements and the performance of the communications harness so that the performance of the processes can be scaled for different sized arrays of transputers. Finally, the overall performance of the lifting-surface panel method is assessed and estimates made for its scaled performance.

5.10.2 Code Efficiency

For an implicit algorithm, code efficiency scales with the size of the transputer array. Figure 5.15 plots code efficiency against the total number of panels. These are code efficiencies for a 4 transputer array with a diameter of 2 links as compared to the performance obtained for an identical number of panels on a single guest transputer. The memory size of 1MByte per transputer on the Ship Science Transputer System only allowed performance on up to 200 panels to be assessed. Lines are shown for influence coefficient setting up, solution of linear system of equations(block Jacobi and Gaussian elimination), and velocity field calculation. Also shown are lines for the two fundamental processes: matrix transposition and matrix-vector multiplication. For all the processes η_c generally increases with the number of panels. The velocity field calculation performance is constant as would be expected for a farm algorithm. The efficiency of 91% reflects the overheads associated with generating the test nodes on

the Host transputer. Therefore, the efficiency of the wake adaption process would be expected to be higher.

The efficiency of the CalculateIC process is only 75% and reflects the inherent complexity of the process. It also suggests that potential improvements could be expected from further development of this process.

The Gaussian elimination process, for an implicit algorithm, shows a rapid increase in efficiency as the number of panels increase as does the Matrix-Vector and Block-Jacobi processes to a 70%-80% efficiency. Even the Transpose operation has a 50% efficiency for a 200 panel problem.

Overall, even for a relatively small panel problem (200) the code efficiency of the component processes of the lifting-surface panel analysis are all in the range of 70% to 90%. This efficiency would be expected to increase for larger sized panel problems.

5.10.3 Performance scaling

For a given problem with N body panels, NW wake panels, and NI images (reflection: NI=1, 4-bladed propeller: NI=3) the total number of calls to calculate the dipole/source influence of a panel at a given node will be:

$$N_{ic} = (NI + 1) \, . \, N \, . (N + NW) \qquad [5.27]$$

Each transputer in an array of T transputers carries out (N_{ic}/T) of these calls. The efficiency of the influence coefficient process on T transputers is η_{cic}. The time to solve the resulting system of equations and to carry out the Kutta condition iteration can be assumed proportional to N^2/T. These assumptions gives the total time t to solve an N panel problem on T transputers as:

$$t = \left(\frac{N_{ic}}{\eta_{cic} \, T} \right) t_{ic} + k \left(\frac{N^2}{\eta_{cbj} \, T} \right) \qquad [5.28]$$

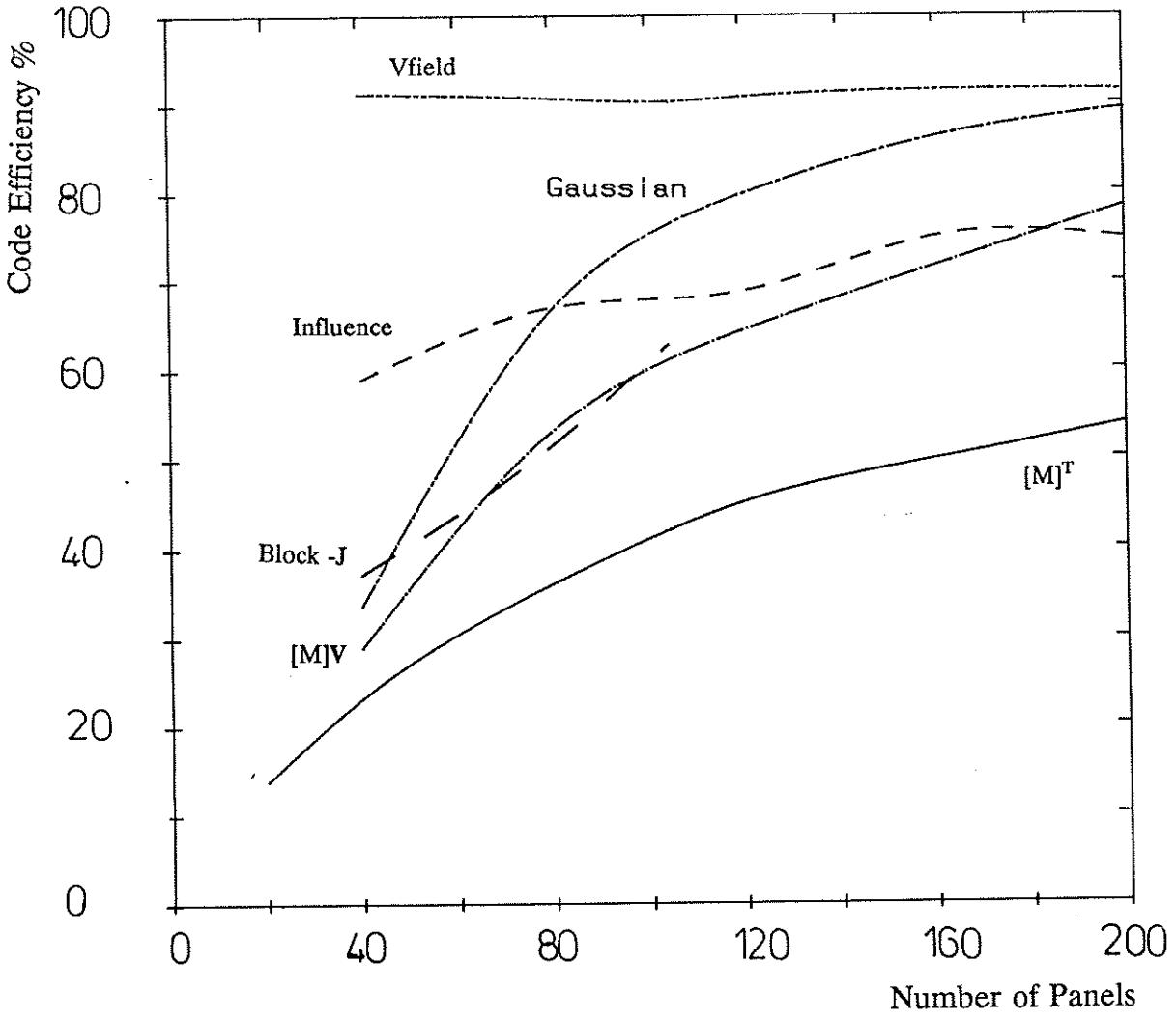


Figure 5.15 Code-efficiency of lifting-surface component processes on a four transputer array

where t_{ic} is the average time to calculate the source/dipole influence coefficient using the Newman panel procedure (for a rudder problem an average of 18-20 ticks of the transputer clock). The block-Jacobi calculation has a fairly low communication requirement and its efficiency (η_{cbj}) will only be weakly degraded by increasing the transputer array size. However, the influence coefficient efficiency η_{cic} will reduce as a function of $\sqrt{T}$, the mean message distance being $\sqrt{T}$-1 links. However η_{cic} is likely to still increase with increasing total number of panels which will generally increase with the transputer array size. This increase will offset some of the degradation due

to increased communication. If the Harness performance Equation[3.14] is used with L=√T-1, and b (no. of bytes) proportional to (N/√T) then the total communication time t_c for (N/√T) messages becomes (expressed in transputer clock ticks for different transputer array sizes):

$$\begin{aligned} t_c &= 825\ N + 2.2\ N^2 \qquad T = 4 \\ t_c &= 225\ N + 0.3\ N^2 \qquad T = 9 \\ t_c &= 91\ N + 0.08\ N^2 \qquad T = 16 \end{aligned} \qquad [5.29]$$

The above expressions indicate that, for constant N, t_c reduces as T increases. The above expressions [5.28] and [5.29] show for a 400 panel problem an almost constant overall η_c with increasing array size.

5.10.4 Lifting-Surface Panel method

The Interaction Velocity Field method for solving rudder-propeller interaction is described in Chapter 7. It is an iterative process: for each iteration the flow is solved individually for a rudder and propeller geometry. The performance of PALISUPAN is problem specific. It is dependent on the total number of panels, panel distribution, wake shape, imposed inflow velocity field, presence of a reflection plane or axi-symmetric images, variables such as rudder incidence or propeller r.p.m. and so on. As an illustration the overall performance is given for a typical rudder geometry (400 panel Rudder No. 2 at -0.4° incidence with representative propeller velocity field) and propeller geometry (400 panel 4 bladed propeller and hub).

Figure 5.16 shows the total time (in minutes) for one complete rudder-propeller interaction cycle for three advance ratio. Each total time is broken down into its four components. The propeller calculation (both solution and velocity field) dominates the overall time. The increase in time with reducing advance ratio indicates the increased three-dimensional rudder/propeller inflows in the more accelerated propeller flow (low J).

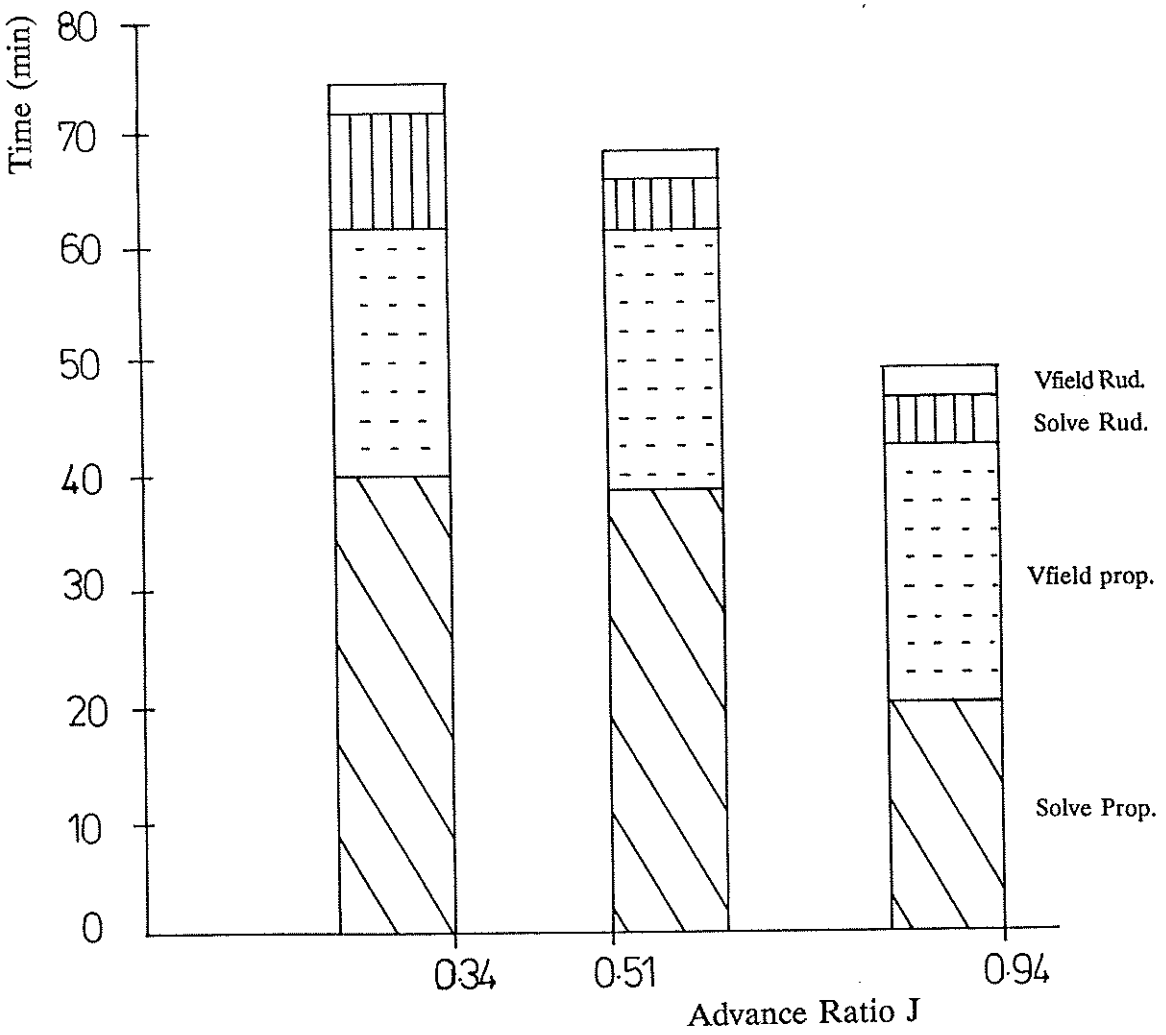


Figure 5.16 Comparison of time to carry out a single rudder-propeller iteration cycle for different advance ratio

It is worth noting the fact that although surface panel methods are considered to only have a moderate computational requirement, this work has demonstrated that there are large possible savings in time through the use of parallel processing. In this case, if the velocity field calculations are assumed to work at 90% efficiency, the propeller at 80% and the rudder calculation at 75% the overall process efficiency is 83%. This is very good performance for what should be a worst case test for parallel processing.

The cumulative tally of calls to the Newman panel process is 7.36 million which gives an average time per call of 9 clock ticks on a 4 transputer array. The time to carry out the J=0.51 iteration cycle estimated for a single transputer (with enough memory) would be 226 minutes, measured on a 4 transputer array as 68 minutes, estimated for a 9 transputer array at 30 minutes and estimated for a 16 transputer array as 17 minutes.

5.10.5 Summary

The high code efficiencies (70% to 90%) obtained for the PALISUPAN component processes indicate that:

1) The Harness communications process performs well for the large communications requirements of implicit algorithms.

2) The use of geometric data distribution and geometric algorithms allows implicit methods to perform as well as explicit algorithms and to benefit from the use of parallel processors.

From the analysis of the calculation influence coefficient performance and that of Harness it is expected that there will be little or no degradation with increasing size of transputer array as long as there are at least 25 panels located on each transputer. The expressions given should allow reasonable estimates of the performance of these algorithms to be made for different sized arrays of transputers.

The overall performance of PALISUPAN in solving rudder-propeller interaction illustrates the benefits of parallel processing. On a 16 transputer array the time per complete iteration cycle would be of the order of 20 minutes, a reasonable timescale for the investigation of detailed design information.

A large scale panel problem such as the analysis of a complete aircraft with flaps with the order of 10,000 panels could be solved using 121 transputers with 8MBytes per transputer. The estimated time to solve such a system, with the order of 200 million

calls to Newman panel, would be of the order of 60 minutes. This performance comparable is with that of existing super-computers for a fraction of investment and running costs.

The transputer used in this work is now a 5 year old design. The rapid developments in similar devices will result in even better performance and greater cost saving in the future.

5.11 Conclusion

A complex lifting-surface panel algorithm has been successfully implemented to run across variably sized square arrays of transputers. The ease with which the parallel algorithm has been developed has highlighted the use of the geometric approach for both the algorithm and data distribution. Development time was also reduced through the separation of the CFD algorithm from the Harness.

Significant memory saving and reduced programming time has resulted from the development of a live-memory datastore process for storing large arrays of information.

A large number of standard algorithms has been implemented in the lifting-surface program and the measurement of their performance can be used to provide information on how well other computational fluid algorithms would perform.

The validation of PALISUPAN is described in Chapter 7. Prior to that in Chapter 6 the development of the wind tunnel test rig for the investigation of rudder-propeller interaction, the technique for data acquisition and analysis, and verification of the experimental methods are described.

CHAPTER SIX
WIND TUNNEL INVESTIGATION
OF
SHIP RUDDER-PROPELLER INTERACTION

6.1 Introduction

The aims of the experimental programme were to: carry out a study of the geometric and flow parameters which govern rudder-propeller interaction; provide reliable data for the development of an improved model of the interaction for use in ship manoeuvring and coursekeeping simulations; and derive detailed information for the validation of PALISUPAN. This chapter details the design and installation of the wind tunnel rig and the method of data acquisition and reduction. Test procedure verification was carried out and the results are presented for these cases.

6.2 General

The overall rudder-propeller rig was designed for use in either of the two low-speed wind tunnels at the University of Southampton. The rig consists of two separate units which allow a model ship rudder and propeller to be either tested independently or as a combination. The independence of the two units allows parametric studies to be made of rudder-propeller interactions while at the same time providing a facility for testing the free-stream characteristics of both rudders and propellers.

The use of air as a working fluid rather than water eases the measurement of data and the test procedures. Propeller and rudder cavitation cannot be investigated, but as shown by Kracht[6.1] the onset of cavitation is not normally a factor in determining rudder-propeller interaction although it is important in the actual design of rudder. The use of air introduces compressibility to the flow. However, if propeller revolutions and diameter are controlled so that the blade tip Mach number is less than about 0.5 the flow generated will be similar to that in water. For example, based on a maximum revolution speed of 3000 rpm an estimate on the effect of compressibility

using a Prandtl-Glauert correction $(1-M^2)^{0.5}$ leads to a 3% increase in K_T and K_Q. As most tests were carried out with propeller revolutions at or less than 2100 rpm no compressibility corrections have been made.

6.3 Test Rig Design and Manufacture

6.3.1 Introduction

The research into rudder-propeller interaction required a test rig capable of measuring forces, moments and pressures on a rudder mounted downstream of a propeller of known loading characteristics. Full details of the design of the new rudder-propeller interaction rig and its installation are given in Turnock[6.2]. The principal features of the rig design and manufacture are given in this section.

6.3.2 Propeller Rig

For the tests, a satisfactory range of thrust loading values for a four-bladed propeller was obtained for a maximum revolution speed of 3,000 RPM and a diameter of 0.8 m. The design of the actual propeller, based on the Wageningen B4-40 series, is given in detail in references [6.3,6.4,6.5] and in Section 6.4. An adjustable pitch propeller design was used to extend the available range of propeller loadings.

The safety and integrity of the system had to be guaranteed as the high speed of revolution and mass of the propeller blade would cause significant damage if failure occurred. The supporting structure for the propeller and its associated drive system are capable of :-

- Allowing a range of propellers with diameters (D) between 700 and 1000mm to be tested.

- Alteration of the distance of propeller axis of rotation (Z) from the tunnel floor in the

range of 0.6 to 0.9 m.

- Yawing the whole rig relative to the direction of the wind tunnel flow to carry out oblique flow tests.

- Acquisition of accurate data on the performance of the propeller. A minimum requirement being the measurement of revolutions(n) , torque (Q) and thrust (T_h).

Further constraints on the design ensured that:

- The propeller drive system provided sufficient torque to enable all points within the operational envelope of the propeller to be reached.

- The propeller support structure had minimal impact on the flow-field upstream of the propeller and rudder.

- The propeller rig was compact to enable the rig to be easily manoeuvred both for setting up purposes and for the parametric studies.

- The propeller rig should be independent of the rudder rig to allow free-stream tests of propeller in both ahead and astern flows.

All the rig manufacture, apart from the propeller blades and fairing, was carried out in the joint Aeronautics/ I.S.V.R./ Ship Science workshops at Southampton University. Figure 6.1 shows an overall view of the final propeller rig. The general design philosophy was to provide a simple and robust structure which could be installed rapidly and which would provide accurate information on the propeller performance within the allowed budget.

Thrust and torque have to be measured as close as possible to the propeller. and it was decided to use an in-line strain gauge dynamometer. Figure 6.2 shows the

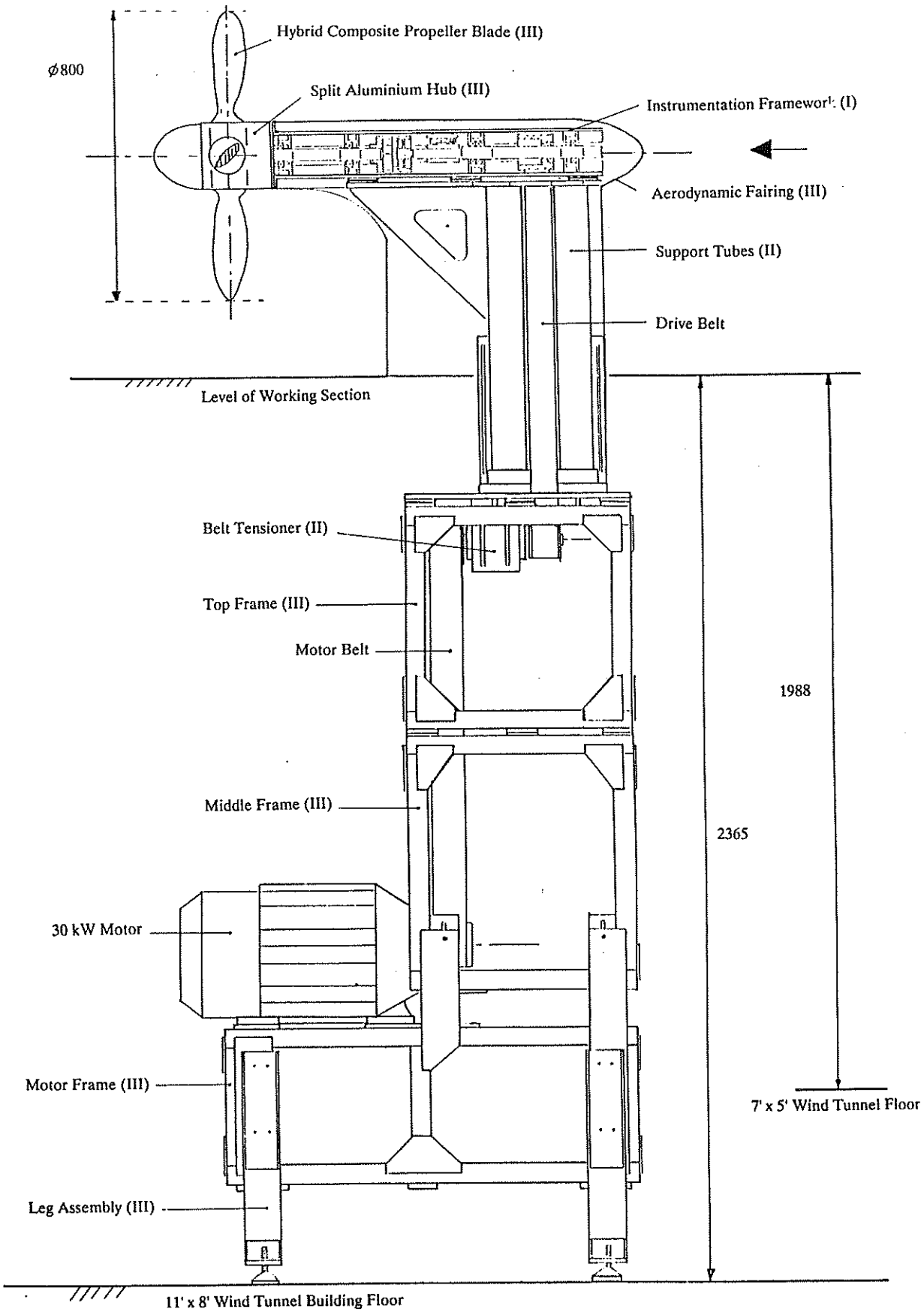


Figure 6.1 Side-view of propeller rig

All Dimensions in mm

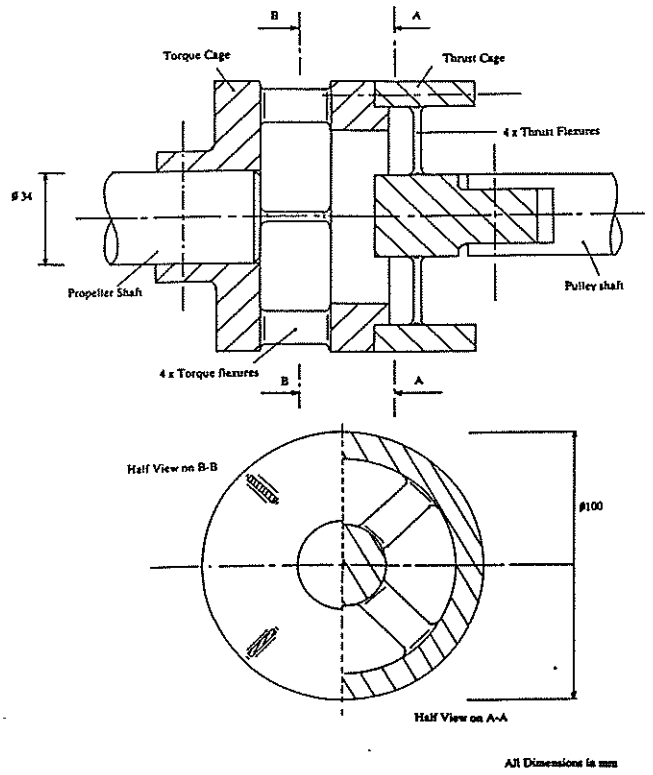


Figure 6.2 Thrust-torque dynamometer design

design of the dynamometer used. The manufacture and calibration of the dynamometer are detailed in [6.6]. The in-line-dynamometer requires 8 wires to be connected to it (4 each for the torque and thrust bridge circuits). Brushes and slip-rings are used to connect the dynamometer to the data acquisition system.

The model propeller and hub need to be rigidly supported with a minimum of disturbance to the propeller inflow. Standard propeller/hub configurations (including CP propellers) have a hub diameter to propeller diameter ratio in the range of 0.2 - 0.25. For a propeller diameter of 800mm this gave a maximum hub diameter of 200mm. As the propeller was to have an adjustable pitch a hub diameter of 180mm was used to allow for fairing of the hub into the root shape of the propeller blades.

This constraint of propeller hub diameter meant that the propeller driveshaft

bearings, associated instrumentation, structural framework and a fairing all had to be contained within an overall diameter of 180mm. The framework supporting the structure in the wind tunnel working section had to be as rigid as possible. Any vibration in the system would adversely affect the modelling of the flow produced by the propeller and the measurement of propeller forces.

The instrument framework was fabricated using 4 lengths of 40mm X 40mm angle section welded to four 1" thick blocks spaced along the length. By having the blocks of sufficient thickness a boxed structure with both lateral and cross-sectional stiffness was produced. Also, the need for access to the propeller shaft and fittings could be accommodated without detracting from the overall stiffness of the framework.

The position of the dynamometer determines the layout of the framework by dividing the power drive shaft into two halves which need supporting by pairs of bearing races. The shaft (propeller shaft) between the dynamometer and the propeller has to have a minimal effect on the forces measured by the dynamometer. The solution used was to have two roller bearing races which have a nominal zero-axial resistance and minimal rolling resistance. The bearings on the other shaft (pulley shaft) have to absorb axial thrust in both directions while locking the overall shaft and dynamometer assembly in position. The power transmission design has a toothed timing belt and pulley arrangement which requires bearings capable of taking radial loads. For the pulley shaft the arrangement used was two angular contact bearings mounted back to back. This acts as a bidirectional thrust bearing and locates the shaft. A plain ball race supports the other end of the shaft and takes half of the radial loading due to the belt tension. The four bearing units were to be each mounted in one of the frameworks blocks. Although this would entail a removal of a large amount of material from the individual blocks the strength would be maintained through the structure of the bearings and their mounts.

It was decided not to use a flexible coupling between the dynamometer and the pulley shaft as this might have an adverse affect on the performance of the

dynamometer. Instead a horizontal boring machine was used to in-line bore all four bearing blocks on a single axis.

The dynamometer is attached to both the propeller and pulley shafts by 4mm shear pins. In the event of overload the shear pins between the dynamometer and the pulley shaft will be first to fail and prevent the hub and propeller shaft pulling out from the framework.

It was found that high-power toothed timing belts provided a cheap and simple transmission system which would be able to transmit 30 kW of power, at 3,000 r.p.m. with a maximum pulley diameter within the allowable width of 140mm. Two support tubes were used for the framework as this minimises blockage and provided additional vibrational stiffness. This allowed the position of the drive belt to be moved further away from the propeller plane of rotation so that the fairing trailing edge was one propeller radius upstream.

The standard frame size of the electric motor needed to provide 30 kW at 3,000 r.p.m. has a mass of at least 250 Kg. Therefore, for reasons of static stability and to reduce the overall mass, the motor had to be placed as close as practical to the wind tunnel building floor . The high power transmitted through small diameter pulleys restricted the length of drive belt, so a lay-shaft and secondary drive belt were required between the motor and the propeller drive shaft. An advantage of this arrangement was that by the use of different lengths of secondary belts the height of the rig could be adapted for use in both of the wind tunnels.

It was decided that rather than the use of idler pulleys for tensioning the drive belts a more reliable scheme would be the physical movement of the various frame structures to directly tension the belts. This would also allow the system to be driven in forward and reverse without the need for adjustment.

To minimise the disturbance to the flow due to the instrumentation frame and

support tubes a fairing was required. For the instrumentation framework a cylindrical cowling with a diameter of 180mm equal to that of the propeller hub provides a smooth flow onto the propeller blade root sections. A nose cone on the upstream end of the cowling and another cone attached to the hub provides an overall aerodynamic shape.

A NACA 63035 [6.7] section with a chord of 550mm and maximum thickness of 148mm at 35% of the chord from the leading edge provides a fair shape over the support tubes and belt while also allowing enough space for displacement of the belt due to vibration. The aerofoil profile was merged into the cylindrical cowling and nose cone to give an overall smooth shape. The trailing edge of the aerofoil profile was stretched from an overall chord of 550mm to 750mm to enclose the angle bracket between the support tube and instrumentation frame.

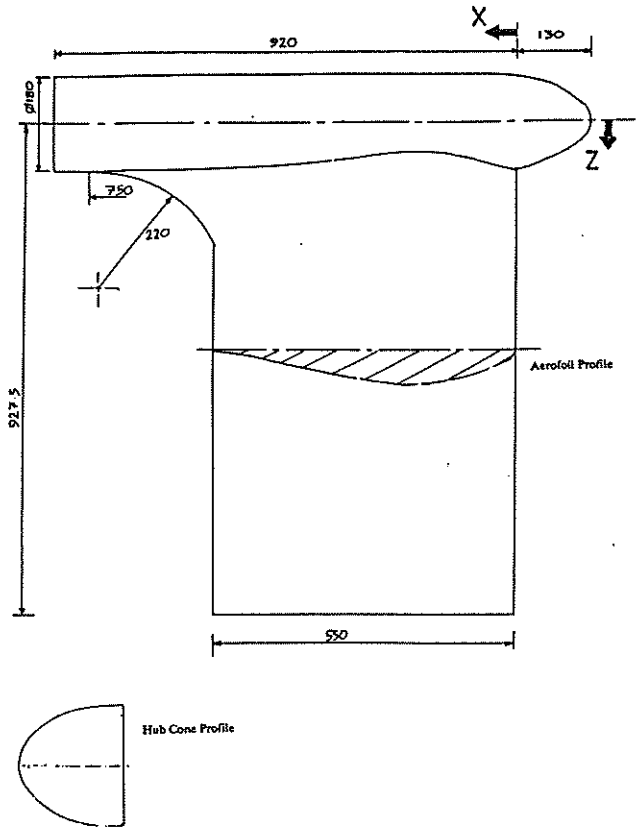


Figure 6.3 Fairing around propeller rig

Figure 6.3 shows the overall shape of the fairing in relation to the underlying structure. The cylindrical cowling and rear section of the aerofoil were manufactured from thin plywood. The nose and hub cones, and the aerofoil nose were made from Glass Reinforced Plastic (G.R.P.)

A digitally controlled variable frequency inverter was used to control the 30kw electric motor between 0 and 3,000 rpm. A remote control box was manufactured for use in the wind tunnel control area. The control functions of the inverter included a facility for forward/reverse driving of the propeller shaft, and direct control of the inverter frequency supply and hence propeller revolutions. The digital nature of the system allows a wide range of control parameters to be set. A particularly useful feature is the ability to control the ramp acceleration and deceleration times of the system which permits emergency stops to be carried out in a controlled manner.

6.3.3 Rudder Rig

The rudder rig has been modified from an existing one designed for free-stream semi-balanced skeg rudder tests conducted by Molland [6.8,6.9,6.10]. For this investigation larger rudders than the original series were to be used and the higher forces generated had to be kept within the design loading of the rig based primarily on the permitted stress levels in the five-component strain gauge dynamometer.

The rudder has to be supported clear of the wind-tunnel working section to isolate the dynamometer from tunnel vibration. The gap between the bottom of the rudder and tunnel floor has to be kept to a minimum to prevent spurious gap flow effects.

The required data are the Lift and Drag Forces acting on the rudder (and possible skeg combination) and the three Moments about the x, y and z axes to allow the centre of pressure and rudder turning torque to be determined.

The existing rig already allowed the following features.
-Testing of both all-movable and semi-balanced skeg rudders with and without taper.
-Surface pressure measurements on the rudder to allow local sectional lift coefficients to be determined and hence the spanwise lift distribution across the rudder.
-The angle of attack of the rudder can be increased until stalled flow occurs.
-The skeg and rudder can be set at different angles relative to each other.

The rig was originally for use in the 2.1m x 1.5m Wind Tunnel. The only modification required to the rig was the provision of an extension piece to compensate for the additional height beneath the tunnel working section in the 3.5m x 2.5m wind tunnel.

Figure 6.4 shows the rudder rig for both wind tunnel configurations. The five-component strain gauge dynamometer sits directly underneath the wind tunnel floor. The rudder stand is independent of the wind tunnel working section and for further support can be braced to the structure of the wind tunnel building.

6.4 Modified Wageningen B 4.40 Model Propeller Design

6.4.1 Introduction

For the investigation a propeller was to be used which had an overall diameter of 800mm and with a maximum hub-to-propeller diameter ratio of 0.25. The propeller was to have four blades to give an overall blade area ratio of 0.4. The representative propeller shape chosen was based on the Wageningen B-Series [6.11] propeller. This allowed experimental data for torque, thrust and efficiency obtained in the wind tunnel tests to be compared with previously published results. To give a variable propeller thrust loading for a given speed of revolution the blades incorporate a means of adjusting the pitch setting for both on and off design conditions.

The complex curves of a ship propeller require a highly accurate manufacturing

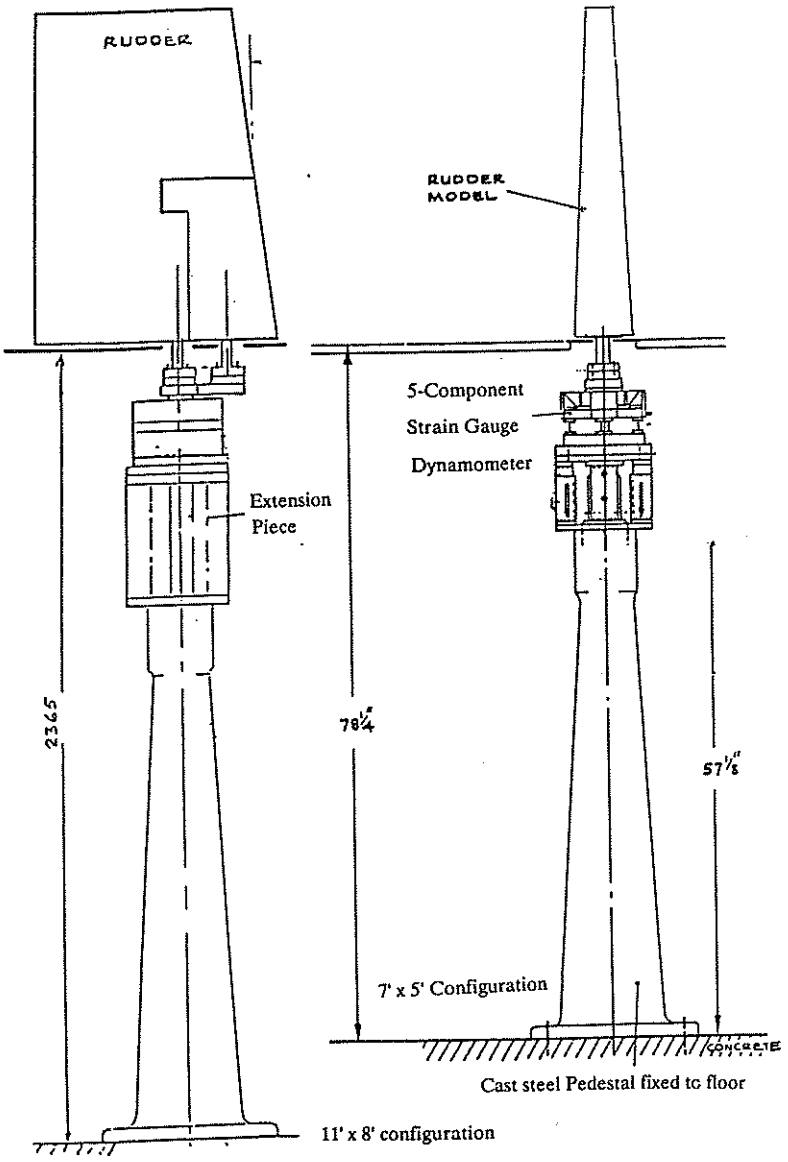


Figure 6.4 Rudder rig arrangement for both wind tunnels

technique to ensure a correct representation of the designed profile. Conventional methods of manufacture are expensive using materials such as bronze alloy and high precision casting and machining operations.

Following experience gained within the University of Southampton in the manufacture of carbon fibre composite blades for a model wind turbine rig[6.12] it was decided to use a similar route for the manufacture of the ship propeller blades. The

hybrid carbon/glass fibre composite design and manufacture of the blades themselves is detailed in Molland & Turnock[6.3,6.4,6.5]. The advantage in such a route lay primarily in the cost savings associated with the manufacturing technique. An additional benefit was that the composite blades could all be laid up using the same mould and would have an identical final blade shape. This compares with direct machining techniques of cast metal alloy blades where there are possibilities for variation in the final shape.

A local machining firm (Jason Engineering (Totton) Ltd.) had available a 3-axis Numerically Controlled (N.C.) vertical axis milling machine suitable for the manufacture of the split female mould surface. The machining of a male-plug and production of a female mould from it was considered. It was decided that it was simpler and more accurate to directly manufacture the female mould. Through the use of appropriately size standard HSS mill cutters it was possible to accurately cut both the concave and convex surfaces present on the blade mould surface.

As input data the NC machine required a prepared data file. The production of these data files was carried out using a Nimbus 386VX personal computer. The use of the computer to detail the tool cutter paths for the production of the mould required an accurate numerical description of the propeller surface. An advantage of the numerical definition was that it allowed the final propeller profile to be refined and the effects of this refinement process to be investigated.

6.4.2 Computer Aided Design of Propeller

Optimum propeller design requires that, for a given working fluid, the propeller will develop the maximum thrust for a minimum power input at a given speed of revolution. The geometry of the propeller mount and surroundings constrain the overall shape of the propeller while the blades have to be made of a material which can withstand the resultant loading on the blades.

The working fluid for the rudder-propeller interaction investigation was to be air. As a result of using the propeller in air rather than water the loading regime of the blades is changed from one dominated by hydrodynamic forces to a loading controlled by centrifugal forces. Due to this change in loading it was proposed to modify the shape of the Wageningen profile without significantly altering the aerodynamic performance of the propeller. By reducing the rake and sweep of the blade the moment about the propeller axis at the blade root due to the centripetal accelerations can be reduced. Also as the propeller was to be capable of being set at different pitches the blades were separate from the hub and so extra section thickness was required at the root to allow for blade attachment.

Conventional description of propeller geometry is based on a cylindrical coordinate system. The first stage of obtaining an accurate numerical description of the propeller surface was to transform the cylindrical coordinate into a conventional right-handed orthogonal cartesian system. The definition of the propeller in a cartesian coordinate system can then be directly used for the generation of Numerical Controlled cutting data files. It can also be used to generate the geometry for a propeller blade lifting-surface model. This has the advantage of an identical surface being used for both the actual blade and the numerical model.

6.4.3 Definition of Propeller Shape

The formal definition of a ship propeller is based on the specification of blade sections at a given radial distance from the centre of rotation relative to a straight generator line with a defined rake forward or aft of a plane perpendicular to the axis of rotation. Each section has associated with it a pitch angle ϕ defined in terms of the propeller pitch P.

$$\tan \phi = \frac{P}{2 \pi r} \qquad [6.1]$$

In general the propeller definition is tabulated in terms of six non-dimensional parameters for each blade section:

x/R - the radial distance divided by the maximum propeller radius.

P/D - the pitch ratio.

c/D - the total chord of each section.

h_d/D - the distance of the leading edge from the generator.

t/D - the maximum thickness

h_t/D - the position of the point of maximum thickness from the leading edge.

The section coordinates are given in terms of a percentage distance from the section maximum thickness based for the front (leading edge) on h_t and for the rear (trailing edge) on (c - h_t). The back and face coordinates are given as a fraction of the maximum section thickness t/D. For sections with a radiused leading edge extra information is required to define the nose.

6.4.4 Transformation to Cartesian Co-ordinate System

The transformation of the propeller definition based on pitched cylindrical sections into the right-handed cartesian coordinate system was carried out as a series of transformations. The transformation actually used is an approximation with the cylindrical sections assumed to lie on a circular arc of radius r rather than the correct helix with radius r. The analysis of the error due to this approximation is described in Appendix G. The effect of the approximation is that the propeller is always operating in the off-design condition and this is consistent for all pitch ratio settings. The process is illustrated in Figure 6.5. The first stage for each section was to define a cartesian coordinate system with an origin at zero radius with axis x1 out along the generator, axis x3 vertically upwards and axis x2 orthogonal to x1 and x3. The base of the cylindrical section lies in the origin plane formed by x1 and x2 along an arc at radius r from the origin so that the height of point P located on the propeller back or face is the thickness t. To determine the location of P in the x1-x2 plane the radial arc-length

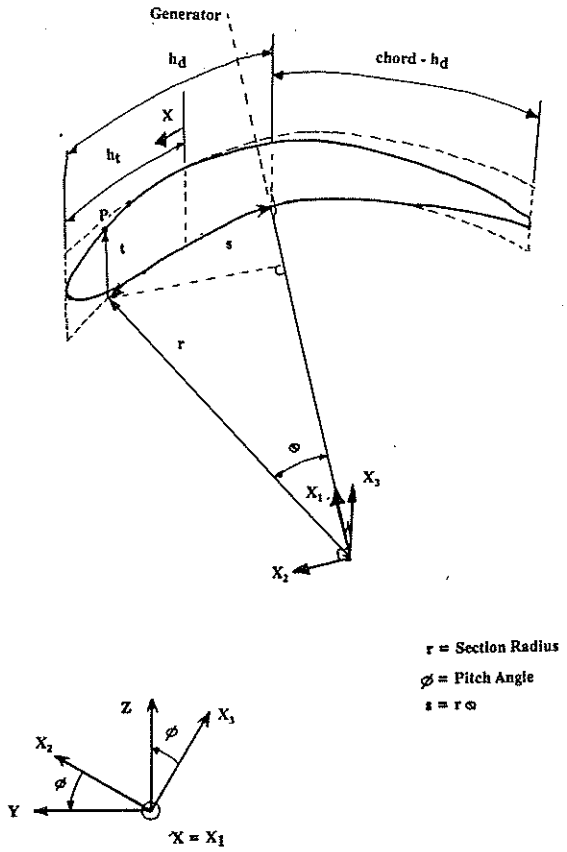


Figure 6.5 Transformation from cylindrical propeller definition to cartesian

s from the generator line is required. The value of arc-length s is defined as :

$$\begin{aligned} s &= \left(h_d - h_t \right) + 0.01\, x\, h_t \quad for\ x > 0 \\ s &= \left(h_d - h_t \right) + 0.01\, x \left(c - h_t \right) \quad for\ x < 0 \end{aligned} \qquad [6.2]$$

The arc-length s divided by the radius r defines an angle θ.

$$\theta = \frac{s}{r} \qquad [6.3]$$

This angle is used to give the value of point P in co-ordinate system x1,x2,x3 as

$$P = \begin{bmatrix} r \cos\theta \\ r \sin\theta \\ t \end{bmatrix} \qquad [6.4]$$

The final right handed coordinate system (x,y,z) has an identical origin to (x1,x2,x3) with the x axis coincident with x1. The final transformation to give (x,y,z) is found by pitching (x1,x2,x3) about x1 by an angle equal to the pitch angle ϕ.

$$\phi = \tan^{-1}\left(\frac{D\ (\ P/D\ )}{2\pi\ r} \right) \qquad [6.5]$$

The final transformation, expressed in matrix form, is:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix} . \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad [6.6]$$

Applying these transformations to the original propeller definition coordinates gives an approximate cartesian representation of the propeller blade surface.

6.4.5 Modification to Original Propeller Sections

A computer program was written which reads a standard tabulated propeller definition and then allows modifications to be made to the final propeller shape. Modifications to the overall chord length and maximum thickness of individual sections were carried out by directly editing the propeller definition input file prior to use by the program. However, the effective sweep and rake of the propeller were changed by respectively altering the distance of the leading edge from the generator and altering the position of the section baseline relative to the thickness.

The prime motive for altering the relative orientations of the sections was to reduce the moment at the blade root due to the centripetal acceleration. By determining the position of the centre of area of each section relative to the propeller generator line the necessary changes in generator position h_d and thickness t for each section could be determined and accordingly adjusted to lie on the generator.

6.4.6 Interpolation of Propeller Surface.

The Wageningen B4.40 propeller defines the back and face surfaces at 10 radii. Therefore, for intermediate locations some form of bi-directional interpolation is required to define the propeller surface. The surface definition has to be detailed enough for use in the N.C. machining of the blade or blade mould.

The procedure devised was a compromise between a straight linear interpolation and a full parametric cubic spline. There were insufficient points to define the complex curves using linear interpolation and it was judged the extra computational complexity of a parametric cubic spline was not needed. Instead, a parametric quadratic spline routine was developed. This requires as input three points only and the parametric equations to be solved are quadratic expressions which are directly evaluated. The detailed expression is given in Turnock[6.3].

The quadratic spline routine is designed to find points along a curve which have a given co-ordinate position x, y or z. For the propeller surface two interpolations have to be carried out in the x and the y direction to define the surface height z. The first stage of the propeller surface definition was to define a complete section(y,z) at a given distance x from the origin. This was the most obvious method as the N.C. production of the mould was to be carried out by cuts in y-z planes. After definition of the two-dimensional section the height z of the face or back surface could be found for any y section within the section data. A special case was the nose where the face and back surfaces join. In this region there are two values for z for a given y and therefore the procedure was inverted and z defined and used to determine y.

6.4.7 Final Propeller Design

Figure 6.6 compares the final outline of the modified Wageningen profile and the original shape. The propeller definition used to produce this profile is given in Turnock[6.3]. It was decided to alter the rake by modifying zc. Only the sweep for the two innermost sections were altered as this gave a minimal change in shape while reducing the direct stress bending moment levels. The alteration of the sweep of the

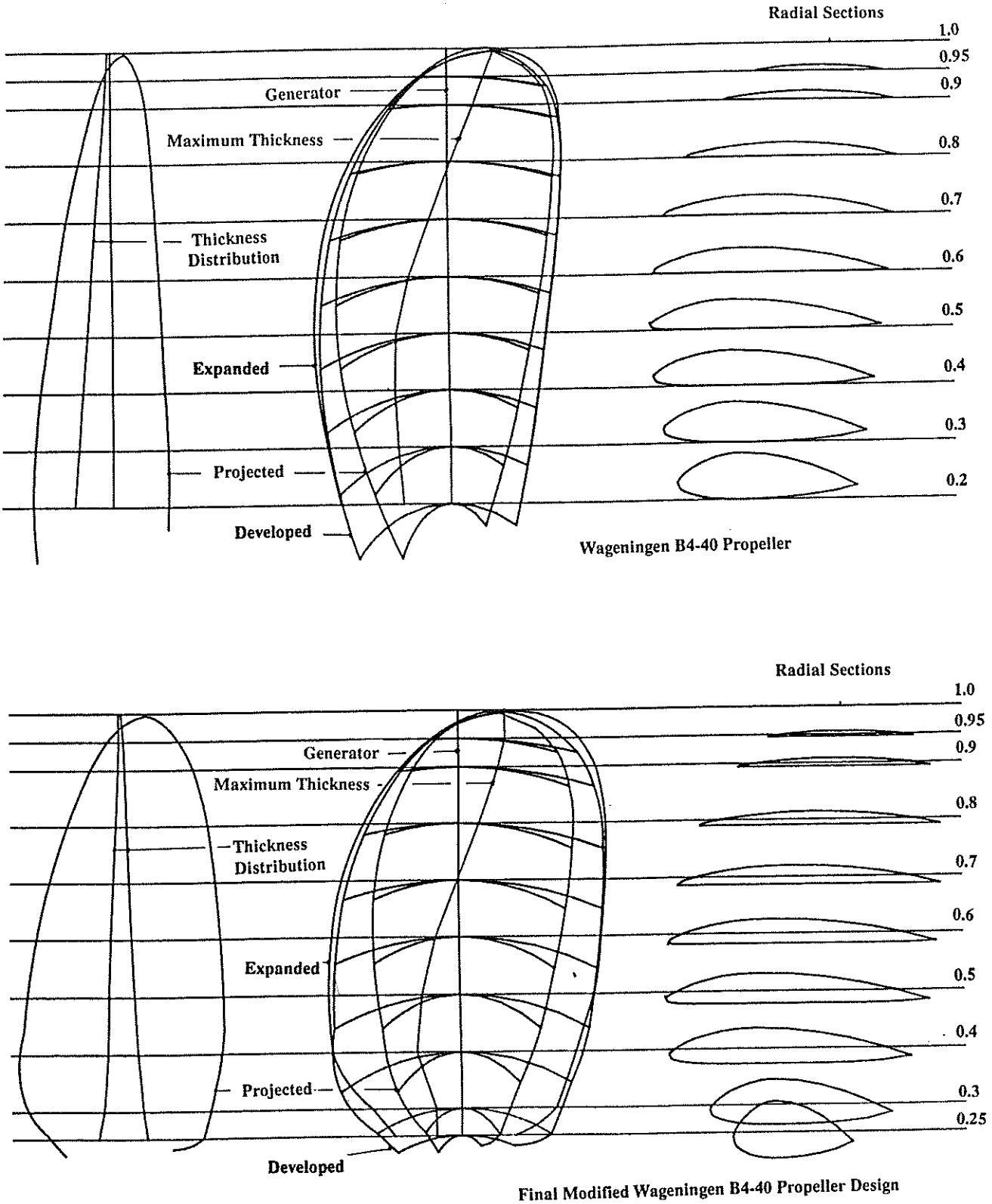


Figure 6.6 Comparison of original and final propeller profile

two innermost sections gives rise to the knee observed in Figure 6.6.

The propeller hub design prescribed a chord of 90.0 mm at the radius of 100mm (0.25 of the maximum value). At this point the blade merges into a spherical cap of radius 100mm. The spherical surface allowed the blade pitch to be varied without affecting the flow around the hub/blade region. The propeller blade was manufactured using carbon and glass fibre composite material. This meant the blade root section thickness had to be increased to allow for the spreading of the fibres into the hub attachment stem. This stem had a minimum diameter of 60mm before belling out inside the hub. The maximum direct stress levels occur in this region.

6.4.8 Propeller Hub

Following experience gained on the use of composite blades at Southampton University each composite blade has at the root, an aluminium collar which locates into a hub split along the propeller plane of rotation as shown in the photograph in Figure 6.7. The split hub is used to clamp the blades in position when they have been set to the desired pitch. For ease of machining and to reduce the rotational inertia of the system the split hub was designed in aluminium alloy.

Four shear pins take the thrust loading and a key resisting torque are used to attach the hub to the propeller shaft. Precision machining on a horizontal boring machine was required to ensure all four blades were located at the same radial position relative to the centre of the hub. During boring of the four holes and faces to locate the blade collars a thin paper gasket was used to ensure a positive clamping action when the two halves of the hub are bolted together. The recess at the rear of the hub was provided to allow space for the provision of masses required after dynamic balancing of the hub and blades. Sufficient space is available at the front of the hub for the use of a scanning pressure transducer for possible future measurement of blade surface pressures. Spherical blanks were manufactured to allow the hub to be used as a two-bladed propeller or with no blades. The blanks clamp in the same manner as the

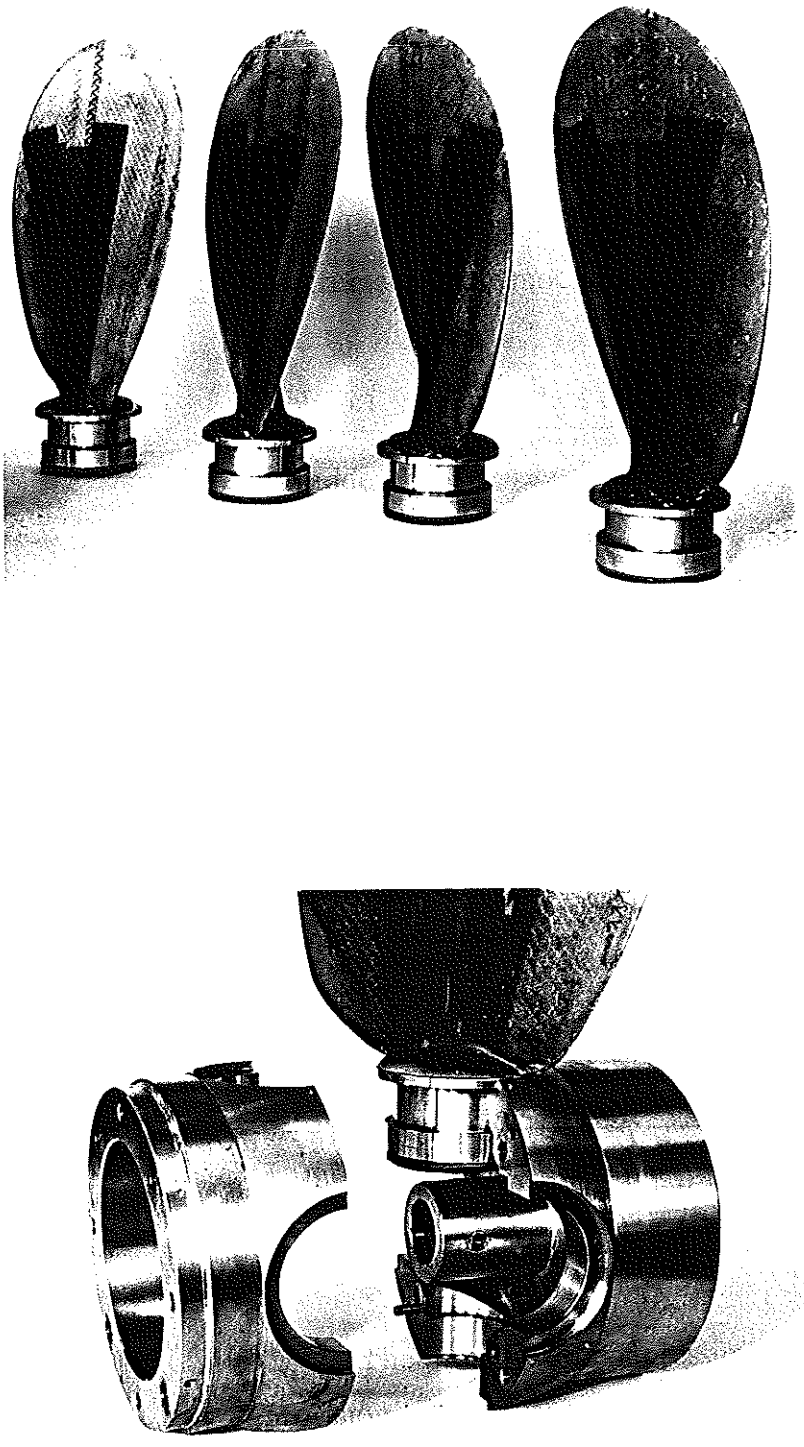


Figure 6.7 View of propeller hub and blades

blade.

6.5 All-Movable & Semi-Balanced Skeg Rudder Design

For the parametric investigation a series of six all-movable rudder models were constructed. These models allowed the effect of varying rudder span, chord, aspect-ratio

Table 6.1 Particulars of Rudder Models Manufactured and tested

Rudder No.	0	1	2	3	4	5	6
Mean chord c(mm)	667	667	667	667	667	800	556
Span S (mm)	1000	1000	1000	1200	1300	1000	1000
Aspect Ratio AR	3.0	3.0	3.0	3.6	3.9	2.5	3.6
Taper Ratio c_t/c_r	0.8	0.8	1.0	1.0	1.0	1.0	1.0
Thickness/chord t/c	0.2	0.2	0.2	0.2	0.2	0.2	0.2
Section	NACA0020 Root and tip with square tips						

Note: All-movable rudder No. 1 is semi-balanced skeg rudder No. 0 with sealed gaps.

and taper to be tested. Table 6.1 details the rudder parameters and Figure 6.8 shows the series of rudders manufactured.

The investigation of pressure distribution across the rudder surface required the provision of pressure tubes at numerous points on the rudder surface. The method used to achieve this is based on that used in free-stream tests(Molland[6.8]).

Thin grooves were routed in a spanwise direction for all the required chordwise pressure locations. A plastic tube with diameter equal to that of the groove was glued inside the groove ensuring that the tube was always below the surface of the rudder.

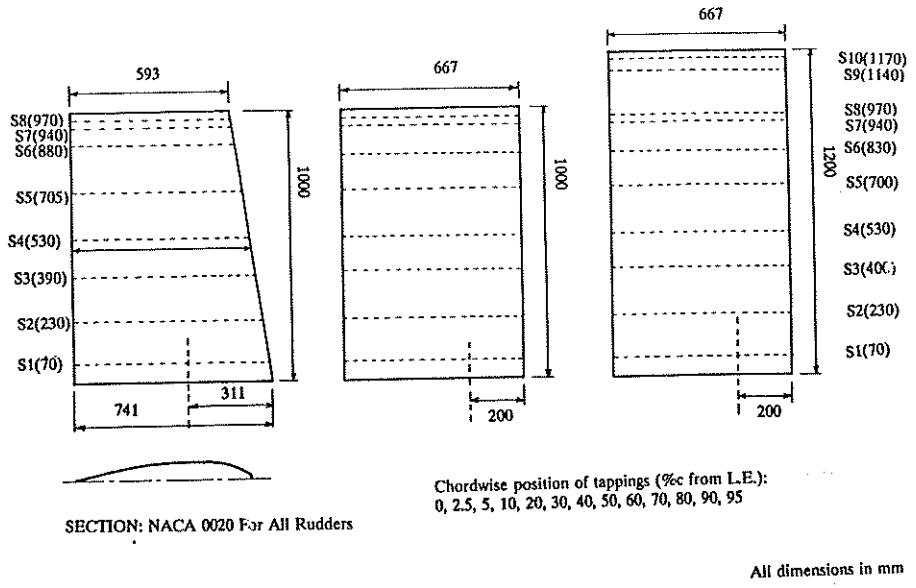


Figure 6.8 Geometry of rudder models

The surface of the rudder is then made good with filler. For all the required pressure locations a fine 1mm diameter hole is drilled through the filler into the plastic tube. One end of each tube is sealed and the other is taken out of the tunnel and connected to a port on a Scanivalve. For taking pressure measurements all holes bar one on a particular tube are sealed with clear adhesive tape.

A semi-balanced skeg rudder was manufactured (Rudder No. 0). Its dimensions are those of the original Rudder No. 2 [6.9] scaled up to have a mean chord of 667mm and total span of 1m. The rudder has a taper ratio of 0.8 and a section profile of NACA0020.

The rudder can be used either as a semi-balanced skeg rudder (No. 0) or as an all-movable rudder (No. 1). Two flanged steel tubes are used to attach the rudder and skeg to the dynamometer. The rudder was manufactured out of a solid assembly of Jelutong planks. On completion considerable interior wood was removed to keep the mass as low as possible without sacrificing structural strength.

The base geometry for the parametric study is the all-movable Rudder No. 2. This has a rectangular planform with a constant chord of 667mm and a span of 1m. Two add-on pieces allows the span of the rudder to be increased to 1.2m (No. 3) and 1.3m (No. 4). The rudder is attached to the dynamometer by a single flanged tube at 30% of chord from the leading edge. The section profile was a NACA0020 and the rudder is a scaled version of the original Rudder No. 3 [6.9] without sweep. The simple shape of the rudder allowed a lighter construction. Thin planks of mahogany were glued across 1" ply formers and then shaped to the desired profile. Four tubes were used for each chordwise pressure position so that a complete spanwise pressure distribution could be achieved in a quarter of the time of that for the skeg rudder. Two similar rudders with 1m span and chords of 800mm (No. 5) and 557mm (No. 6) were also manufactured.

6.6 Data Acquisition System

6.6.1 Description

The large number of individual data readings required the use of an automated system for data acquisition. The system used is based on that developed for measuring the performance of 2-D aerofoil sections (Turnock[6.13]). A schematic of the data acquisition system is shown in Figure 6.9.

Data is recorded from the five-component strain-gauge dynamometer which measures the forces and moments acting on the rudder. The propeller's torque and thrust are measured using another strain gauge dynamometer. The propeller revolutions are measured using an optical encoder attached to the pulley shaft which gives a voltage output proportional to RPM. Quasi-steady pressure measurements over the surface of the rudder are obtained using a compressed air stepping Scanivalve which exposes four differential pressure transducers to one of 36 input ports each. This allows pressure data to be measured from a maximum of 144 individual pressure tubes.

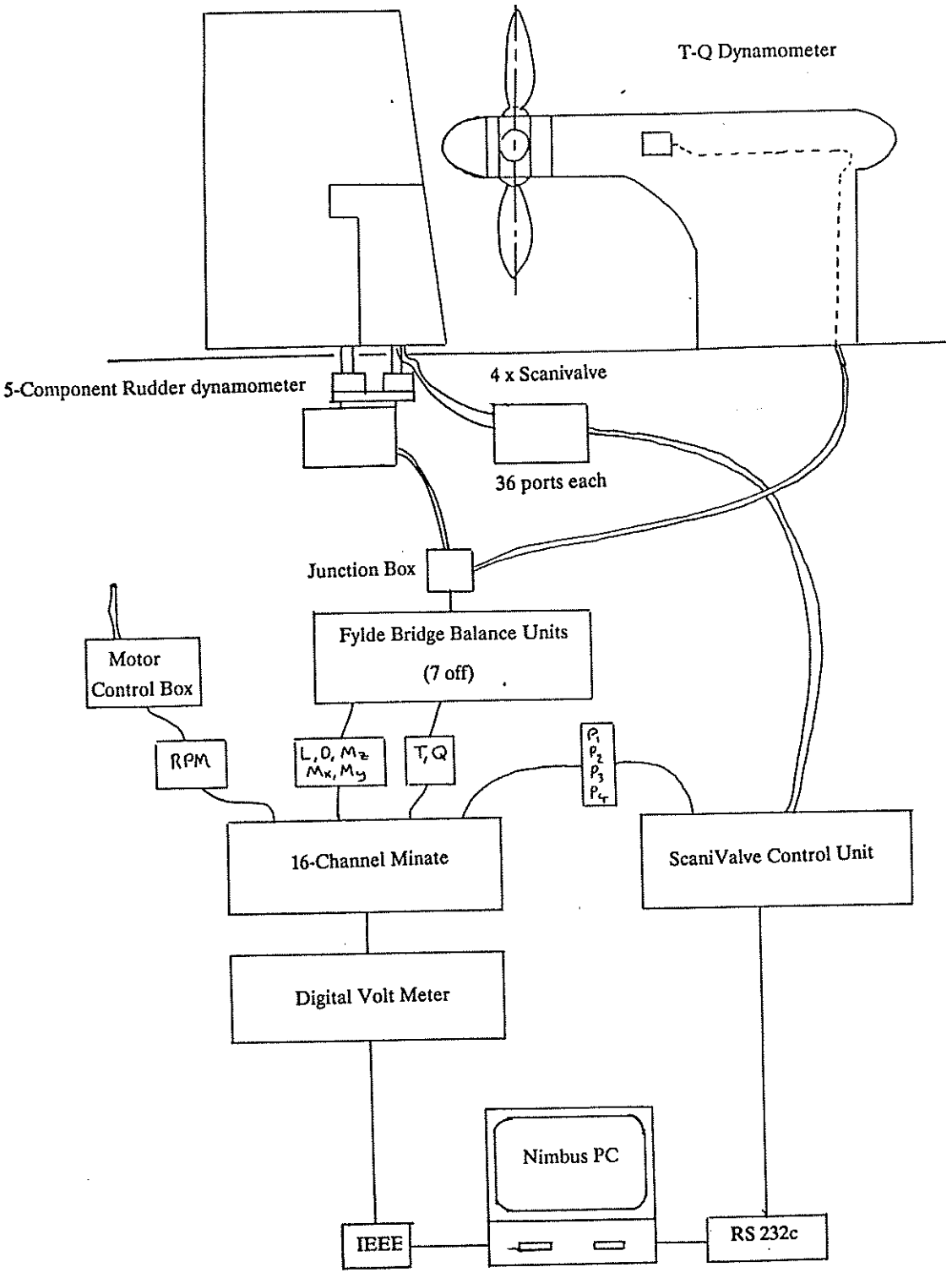


Figure 6.9 Data acquisition schematic

The measurement component of the system is an accurate digital voltmeter connected to one of 18 input channels. The voltmeter and input channels are controlled by software running on a RM Nimbus PC via an IEEE connection. The stepping of the scanivalve is also controlled via an RS232c system connected to the PC. The computer provides data storage and backup through the use of twin floppy disk drives.

6.6.2 System Hardware

The data acquisition hardware used for the Rudder and propeller rig was mainly existing equipment with well documented performance. The individual system components are described below and the overall layout for data flow is given in Figure 6.9.

I Six-Component Strain-Gauge Rudder Dynamometer

The design, construction and calibration of this dynamometer is described by Molland [6.8]. Designed originally for free-stream semi-balanced skeg rudder tests no modifications were required for use. Only five force/moment components were used. These were Lift (L), Drag (d), and the z, x and y Moments (Mz, Mx and My). The strain gauge bridge circuits used Fylde units to provide a bridge voltage supply and balance. In use, the zero (wind-off) voltages and supply voltages were measured regularly.

II Thrust-Torque Propeller Dynamometer:

This dynamometer was designed and constructed specifically for the Propeller rig. The design and calibration of this dynamometer, manufactured in the University of Southampton workshops, is detailed in Molland[6.8]. Design maximum loadings are 110 Nm torque and 700N Thrust. The rotational nature of the dynamometer means that a stationary null-loading voltage cannot be measured. Instead, a null-loading

calibration procedure has been developed. This uses the fact that propeller thrust and torque at low speed are proportional to the revolutions squared . With the wind speed at zero the dynamometer bridge voltages are measured for three different revolutions, typically 300, 600 and 1000 RPM. These voltages are plotted against the square of the revolutions and a least-squares fit applied to determine the effective null-loading voltage. This procedure is quick to carry out and was repeated regularly to check for possible drift in the estimated null-loading voltage.

III Fylde Bridge-Voltage Units.

These units were purchased by the Department for general use with strain-gauge bridge circuits. They provide an excellent stable supply voltage with negligible drift. The units provide a facility for compensating for the offset voltage. It is possible to vary the supply voltage in the range 1-15v and either to use the unit without amplification of the bridge balance voltage or with amplification in the range of 1 to 100. For these tests an accurate digital micro-voltmeter was available so no amplification was used. This ensures that potential non-linear amplification problems do not occur and the calibration of the amplification factor is not needed.

IV Minate -Solartron Digital Voltmeter (7061)

Voltages can be measured to an accuracy of 0.1 microvolt which provide good resolution for strain-gauge bridge circuits with designed outputs in the range 0-5,000 microVolts. The voltmeter is controlled via an IEEE parallel cable from any IEEE compatible device. The voltmeter has 18 separate channels for input.

V Scanivalve Controller

For these tests a gang of four 36 port rotary differential pressure transducers (Scanivalve) was used. A compressed air supply is used to provide power to step the scanivalves around the 36 input ports. The reference pressure for all four transducers

was taken from the tunnel wall static pressure in the entrance to the tunnel working section. The stepping of the scanivalve is driven by a custom built unit. An 8 bit parallel cable was used to provide the signal to either step the scanivalves one port or to return the scanivalve to the Home position.

VI RM Nimbus PC

The Nimbus PC is used in Ship Science Department as the main means of controlling data acquisition systems. The configuration of the system used for these tests consisted of a dual floppy disk RM Nimbus PC fitted with a GPIB interface board. The GPIB board provided the IEEE connection necessary to control the Minate digital voltmeter and automatic rotary switch. The COM1 port was used to control the scanivalve controller. The control software used is described in the next section.

6.6.3 Acquisition Software

The software used for acquiring data and controlling the devices connected to the Nimbus PC is based on that developed for the 2-D aerofoil tests (Turnock[6.13]). Executable programs were developed using the PROPASCAL dialect of Pascal as a wide variety of procedures were available for driving the devices connected to the PC.

To ease subsequent data analysis a flexible menu driven system was used to allow the operator to define which particular subset of data was to be recorded. A graphical screen display was used to allow monitoring of the data acquired and to produce pressure plots on the screen. A single data file was created for each data recording session which contained all the information necessary to analyse the data. This data was stored in ASCII format and allowed a readable hardcopy to be made immediately after a data test. A setup data file stores variable system parameters, calibration constants for the dynamometer and pressure transducers, and other information relevant to the individual test. This data file is read in at the beginning of each test and provides the unique test number. At the end of each test run it is

rewritten, including any changes.

6.7 Commissioning

The propeller and rudder rig was successfully commissioned in March 1990. The first trials of the propeller drive system were carried out in the George Edwards Laboratory, without the propeller and hub assembly. It was demonstrated that the belt drive system worked and could safely run up to the maximum speed of 3000 R.P.M.

Three series of two weeks of wind tunnel testing have been carried out. The first series ,in March 1990, used the semi-balanced skeg Rudder No. 0 and all-movable Rudder No. 1 and validated all the various components of the rig. The second series in August 1990 tested mainly Rudder No.'s 2 and 3 and carried out a detailed parametric study. In September 1991 a further two weeks continued the parametric study using Rudder No.'s 4 to 6.

The propeller rig has been operated, without problem, at the extremes of its operational envelope. Some belt resonance occurs at specific speeds but this imposes no major limitations on its use. Overall it provides an extremely robust and flexible means of testing a ship propeller in air.

6.8 Test Programme

6.8.1 General

Results are reported in this work from the three sessions of testing in the 3.5m x 2.5m low-speed wind tunnel. The sessions consisted of 10, 9, and 10 working days respectively. Running of the tunnel was restricted to between 9 am and 5 pm. Figure 6.10 shows a dimensioned layout of the test setup for the rudder and propeller combination.

Wind speed was set using the wind tunnel speed controller and measured using a Betz manometer. For given propeller revolutions the wind speed controller was varied as necessary to compensate for the wind speed imparted by the propeller. Exploratory tests were carried out at wind speeds of 5, 10, 15 and 20m/s. A wind speed of 10m/s was chosen for the majority of the tests being a compromise between achieving an adequate Reynolds number and enabling a satisfactory range of propeller advance ratios to be achieved. Based on a free-stream velocity of 10m/s and rudder chord of 0.667mm, the nominal Reynolds number was therefore 0.4×10^6. It should be added that velocities induced by the propeller at high-thrust loadings led to effective Reynolds number of over 1.0×10^6 over most of the rudder. English[6.14] showed that rudder sideforce was not affected by scale effects if the Reynolds number based on propeller revolutions and diameter was greater than 1.4×10^6. For these tests at 800rpm(low-thrust loading) the propeller has a Reynolds number value of 1.8×10^6

6.8.2 March 1990

The first session in the tunnel was primarily for commissioning of the rudder and propeller rig. Tests were carried out using Rudder No. 1 in both an All-Movable and Semi-Balanced skeg configuration in the ahead configuration. Only one longitudinal separation with an X/D of 0.34 was used. The propeller was set to a pitch ratio of 0.95 for the whole series of tests. A detailed survey of rudder surface pressure measurements were carried out on Rudder No 1 in its All-Movable configuration. These were conducted at a windspeed of 10m/s and propeller revolutions of 800, 1450, and 2100 rpm. These correspond to nominal J values of 0.94, 0.51, and 0.35 and nominal K_T/J^2 values of 0.05, 0.88, and 2.30. The results obtained for these advance ratio are representative of the propeller's performance over its whole range of ahead operating conditions. Rudder incidence was varied in the range of -30° to 30°. Rudder force and propeller torque and thrust measurements were taken for a wider range of wind speeds (5m/s - 20m/s) and rudder incidence.

6.8.3 August 1990

A detailed parametric investigation of rudder longitudinal separation and rudder aspect ratio was carried out in the second wind tunnel session. All tests were carried out in the ahead condition. At a constant wind speed of 10m/s and for propeller rpm of 800, 1450, and 2100 all-movable rudder's No.2 and No.3 were tested at three longitudinal separations corresponding to X/D = 0.3, 0.39, and 0.53. Surface pressure and rudder force measurements were made at a nominal rudder incidence of -30°, 20°, -10°, 0°, 10°, 20°, and 30°. Rudder forces were additionally measured at rudder incidences between -35° and 40°. Propeller torque and thrust measurements were also recorded for all test points.

In addition to the parametric study, a free stream rudder investigation and open-water propeller tests were carried out. In each case, the redundant part of the rig was completely removed from within the tunnel working section.

6.8.4 September 1991

A series of open-water propeller tests were carried out with 4 blades at pitch ratio settings of 0.69, 0.95, and 1.34, and 2 blades at a pitch ratio of 0.95. With the propeller removed a rudder free-stream test was conducted to measure possible flow mis-alignment. This was found to be +0.4° for all rudders tested.

Rudder No. 2 was tested in the base position with the propeller axis 600mm above the tunnel floor, zero lateral separation, and a longitudinal X/D=0.39. The four-bladed propeller at a pitch ratio of 0.95 was used. This corresponds to the mid-position used in the previous tests and allowed comparison of results.

For the parametric study Table 6.2 lists the location of propeller axis, propeller pitch, rudder pivot and the rudder used for all three test sessions. A constant windspeed of 10m/s and three propeller revolutions of 800, 1470, and 2150 rpm were

Table 6.2 Particulars of rudder-propeller arrangements tested

Rudder No.	0	1	**2**	3	4	5	6
X_1/C	0.35	0.35	**0.3** 0.54	0.3	0.3	0.3	0.3
X/D	0.34	0.34	0.30 **0.39** 0.54	0.30 0.39 0.54	0.39	0.39	0.39
Z/D	0.75	0.75	**0.75** 1.125	0.75	1.125	0.75	0.75
Y/D	0.0	0.0	-0.25 **0.0** +0.25	0.0	0.0	0.0	0.0
P/D	0.95	0.95	0.69 **0.95** 1.34	0.95	0.95	0.95	0.95

used. These respectively correspond to propeller advance ratios of 0.94, 0.51, and 0.35. Rudder and propeller force measurements were made at 5° increments of rudder incidence between -35.4° and +39.6°. Surface pressure surveys were carried out at 10° increments between -30.4° and +29.6° at the mid advance ratio of 0.51. In addition for the two lateral movements surface pressure surveys were obtained for advance ratios of 0.94 and 0.35. A change of rudder pivot was achieved by mounting the rudder stock on the rudder dynamometer skeg fixture. This gave an additional 160mm of pivot on Rudder No. 2.

The wind tunnel has recently been equipped with a system for carrying out non-intrusive flow field measurements using Laser Doppler Anenomometry (L.D.A.). As part of the commissioning process this was used to obtain time histories of velocity of two-components of velocity. Only a limited number of positions were investigated at an advance ratio of 0.51 and rudder incidence of -0.4°.

6.9 Data Reduction

6.9.1 Introduction

The software controlling the data acquisition generated a data file for each individual test. Data was stored as ASCII text which allowed a direct transcription of all data recorded to be obtained. The data recorded were the unprocessed voltages read from the voltmeter. The hard copy allowed 'by hand' verification checks to be made of the software used to analyse the test data files. The format of the test data file is described in Molland and Turnock[6.15]. The data file format was designed to ease the subsequent analysis.

A suite of specific programs were written to manage the large amount of test data acquired from the two wind tunnel sessions. This allows effective interrogation of the database of test results to be carried out. Data reduction has been divided into the three distinct tasks of rudder forces, rudder surface pressures, and propeller thrust and torque. The data reduction of the three categories is detailed in the following sections.

6.9.2 Rudder Forces

The five components of the rudder dynamometer are reduced using the method described in detail in Molland[6.8]. For each force component, the supply voltage Vs and no-load voltage Vo are recorded for each test. With these, the measured bridge balance voltage V_m is offset and then scaled to give a voltage V proportional to the applied force at the original calibrated supply voltage Vc:

$$V = (V_m - Vo) \times (\frac{Vs}{Vc}) \qquad [6.7]$$

$$\begin{bmatrix} L \\ D \\ M_z \\ M_x \\ M_y \end{bmatrix} = \begin{bmatrix} 0.338035 & -1.188x10^{-03} & -3.952x10^{-04} & -1.484x10^{-03} & -3.2605x10^{-03} \\ 2.12x10^{-05} & 0.252512 & -6.659x10^{-04} & 1.6823x10^{-03} & -9.901x10^{-04} \\ -1.82x10^{-04} & 2.845x10^{-04} & 3.9367x10^{-02} & 1.344x10^{-04} & 3.2x10^{-04} \\ 7.002x10^{-04} & -2.37x10^{-05} & -3.4036x10^{-03} & 6.3492x10^{-02} & 2.049x10^{-04} \\ -3.3x10^{-06} & 6.333x10^{-04} & 1.47x10^{-05} & -3.008x10^{-04} & 4.8468x10^{-02} \end{bmatrix} \begin{bmatrix} V_L \\ V_D \\ V_{Mz} \\ V_{Mx} \\ V_{My} \end{bmatrix}$$

[6.8]

Where V_L, V_D, V_{Mz}, V_{Mx} and V_{My} are the scaled and offset voltages in microvolts for the five dynamometer components.

The results presented in this thesis are only for the all-movable rudders tested. Therefore, the forces measured are in the direction of the wind-tunnel axes. The appropriate sign conventions are illustrated in Figure 6.10. The non-dimensional coefficient form of the forces are obtained as shown:

$$C_L = \frac{L}{\frac{1}{2}\rho\, Uo^2 A} \qquad [6.9]$$

$$C_D = \frac{d}{\frac{1}{2}\rho\, Uo^2 A} \qquad [6.10]$$

$$C_{Mz} = \frac{M_z}{\frac{1}{2}\rho\, Uo^2 A\ c} \qquad [6.11]$$

$$C_{Mx} = \frac{M_x}{\frac{1}{2}\rho\, Uo^2 A\ S} \qquad [6.12]$$

$$C_{My} = \frac{M_y}{\frac{1}{2}\rho\, Uo^2 A\ S} \qquad [6.13]$$

where ρ is the tunnel air density (kg/m3), Uo the free stream velocity (m/s), S the rudder span, c the mean rudder chord, and A the total rudder area (A = S .c). The position of the centre of pressure on the rudder in the spanwise and chordwise directions are obtained as follows:

$$CP_c = \left(\frac{M_z}{L} + V \right) \times \frac{100}{c} \qquad [6.14]$$

$$CP_s = \left(\frac{M_x}{L} - W \right) \times \frac{100}{S} \qquad [6.15]$$

where V is the distance of the dynamometer rudder stock from the leading edge and W the distance from the dynamometer measurement centre to the rudder root.

No corrections were made to the values of non-dimensional coefficients thus determined as effects such as tunnel blockage for the 3.5m x 2.5m working section were found to have a negligible effect for the rudder size and propeller diameter tested.

6.9.3 Surface Pressure

The no-wind pressure transducer output voltage V_{Po} was measured at the

beginning of each test. The reference pressure connected to all four transducers was the static line of the main wind tunnel pitot-static probe. This line was also connected to one input port for each transducer to give a 'real' zero pressure difference value V_{Po}. The pitot line was connected to another input port to give a voltage V_{Dp} proportional to the total tunnel dynamic pressure.

For each pressure port measurement V_P the non-dimensional pressure coefficient C_p is obtained directly as:

$$C_p = \frac{V_P - V_{Po}}{V_{Dp}} \qquad [6.16]$$

This expression does not require an explicit value for the calibration constant of the particular pressure transducer. All that is assumed is a linear response.

As the location of each pressure measurement on the rudder surface is known, integration of C_p around the chord of the rudder for a constant span allows the local non-dimensional Normal Force coefficient C_N to be calculated. This integration was carried out using a quadratic numerical procedure similar to Simpson's rule but with variable spacing.

6.9.4 Propeller Forces

The static calibration carried out on the torque-thrust dynamometer gave a linear response to loading of both thrust and torque with negligible interactions (Molland and Turnock[6.6]). The calibration for both channels was carried out with a supply voltage of 7 volts. The slope constant (C_T) for the thrust channel was 0.0903 N/μv and for the torque channel (C_Q) of 0.01284 Nm/μv. The propeller is separated from the dynamometer by a short shaft supported on two axially unconstrained roller bearings. When the propeller is rotating these give minimal torque and thrust losses. However, when stationary, frictional resistance in these bearings is enough to produce

spurious no-load voltages. To overcome this problem, the previously described, dynamic dynamometer zeroing procedure was developed.

For each individual torque and thrust measurement both the balance output voltages, V_Q and V_T (μ v), as well as the balance supply voltage, V_{Qs} and V_{Ts} (v), were recorded. This gives the measured thrust T_h (N) and torque Q (Nm) as:

$$T_h = (V_T - V_{To}) \times (\frac{V_{Ts}}{7.0}) \times C_T \qquad [6.17]$$

$$Q = (V_Q - V_{Qo}) \times (\frac{V_{Qs}}{7.0}) \times C_Q \qquad [6.18]$$

where V_{To} and V_{Qo} are the relevant dynamic zero voltages, the 7.0 is the value of calibration supply voltage in volts and C_T, C_Q are the calibration slopes.

Using the relevant windspeed V (m/s) and n (revs/sec) the advance ratio J is calculated as:

$$J = \frac{V}{n\,D} \qquad [6.19]$$

where D is propeller diameter. The non-dimensional thrust coefficient(K_T) and torque coefficient(K_Q) are given by

$$K_T = \frac{T}{\rho\, n^2\, D^4} \qquad [6.20]$$

$$K_Q = \frac{Q}{\rho\, n^2\, D^5} \qquad [6.21]$$

where ρ is the air density. The propeller efficiency η is:

$$\eta = \left(\frac{J}{2\pi}\right) \times \left(\frac{K_T}{K_Q}\right) \qquad [6.22]$$

6.10 Verification of Test Rig

6.10.1 Presentation of Data

The notation of rudder incidence and coefficients is given in Figure 6.10, noting that the propeller rotates in an anti-clockwise direction when viewed from aft. All rudder force coefficients have been non-dimensionalised in terms of the free-stream velocity. The graphical presentation of rudder force information show the lift and drag force coefficients, centre of pressure chordwise (as a percentage of mean chord from the mid-span leading edge) and centre of pressure spanwise (as a percentage of span from root) versus rudder angle of attack for different propeller advance ratios.

The chordwise integration of surface pressure information allows the variation in local normal force coefficient to be plotted against percentage rudder span from the root. It is assumed that the local C_N at the root is equal to that at the next inboard section and that at the tip C_N is zero.

The data are presented in terms of advance ratio J values rather than conventionally as thrust loading (K_T/J^2). This is to eliminate ambiguity between the use of the propeller free-stream (open-water) thrust coefficient and the actual measured thrust coefficient in the presence of the rudder. The latter is generally larger than that in the free-stream for the same free-stream J value.

6.10.2 Rudder in Free-stream

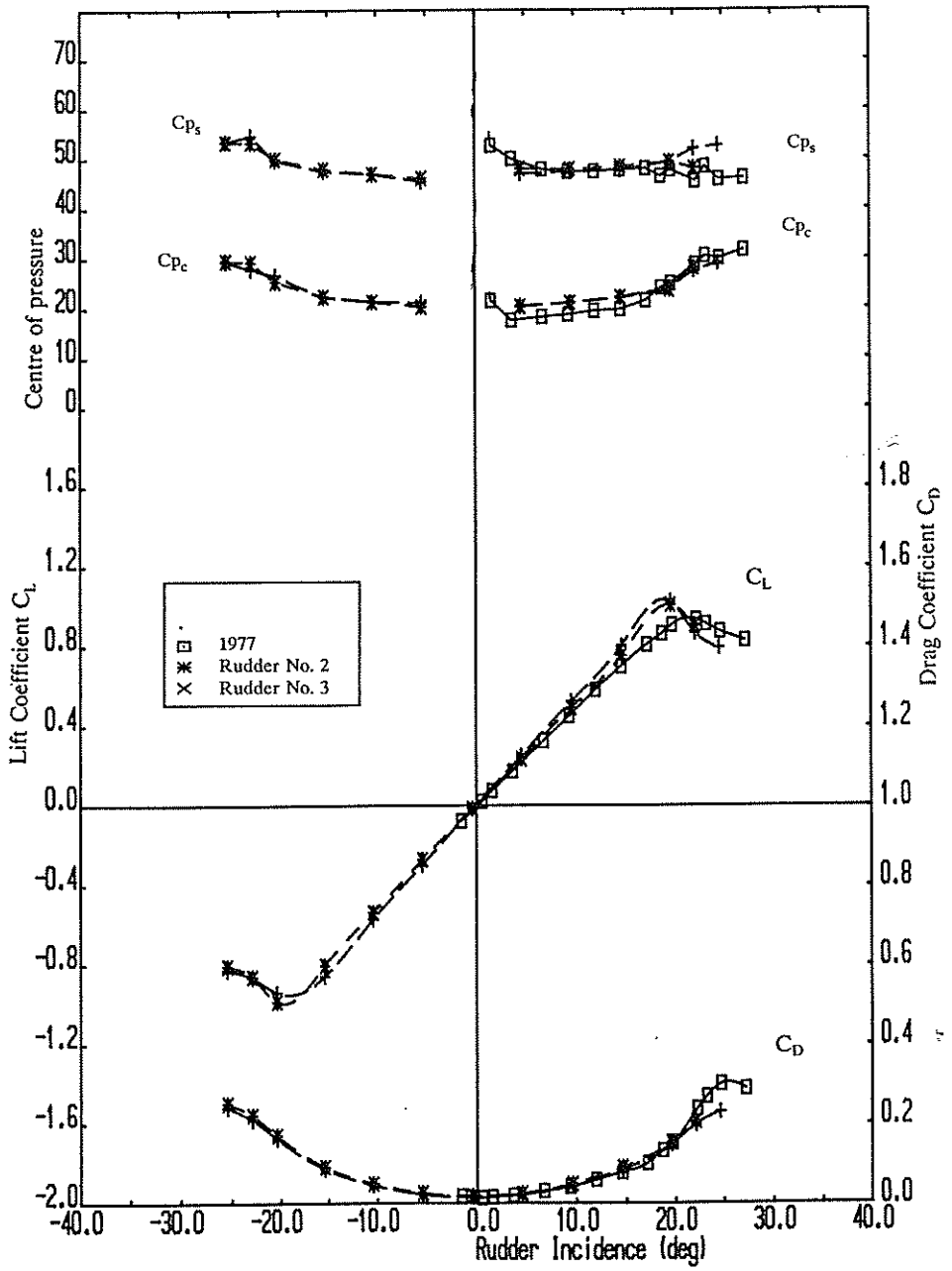


Figure 6.11 Lift, Drag and Centre of Pressure Characteristics in free-stream

Figure 6.11 presents free stream results for Rudders No.'s 2 and 3 together with those obtained for the similar although swept rudder tested in the same rig and reported in Goodrich & Molland [6.16].

The results are very similar verifying the set up and data acquisition procedures for the current tests. Also, the small increase in C_L with increase in aspect ratio between Rudders No. 2 and No. 3 is as expected from low-aspect ratio wing theory.

6.10.3 Propeller in Free-stream

Table 6.3 Details of model propeller

Number of Blades	2 or 4
Diameter (mm)	800
Blade area ratio	0.40
Boss Diameter (max -mm)	200.0
Mean Pitch Ratio	0.69, 0.95, 1.34
Rake (deg)	0°
Blade Root Thickness Ratio	0.050
Section shape	Based on Wageningen B Series
Blade Outline shape	Based on Wageningen with reduced skew

Dimensions and particulars of the model propeller are given in Table 6.3. For most tests the propeller pitch ratio was set at a mean value of 0.95 and the propeller free-stream characteristics for this pitch are shown in Figure 6.12. Full details of all the free-stream performance characteristics, including the results for different pitch settings are given in Molland and Turnock[6.17].

Departures from the Wageningen B4.40 characteristics were relatively small and accountable; these amounted to K_T values within 2% of the Wageningen results and K_Q values 8%-10% higher than the Wageningen results. The deviations in K_T are estimated to be due to the combined effect of the increase in boss diameter ratio from 0.167 to 0.25 (-1% K_T) and compressibility effects (0% to +3% K_T). The increase in K_Q is estimated due to the cumulative effect of increase in boss diameter ratio (+1% K_Q),thickness ratio from 0.045 to 0.050 (+1% K_Q), compressibility effects(0% to +3% K_Q) and Reynolds number effects (+10% to +3% K_Q).

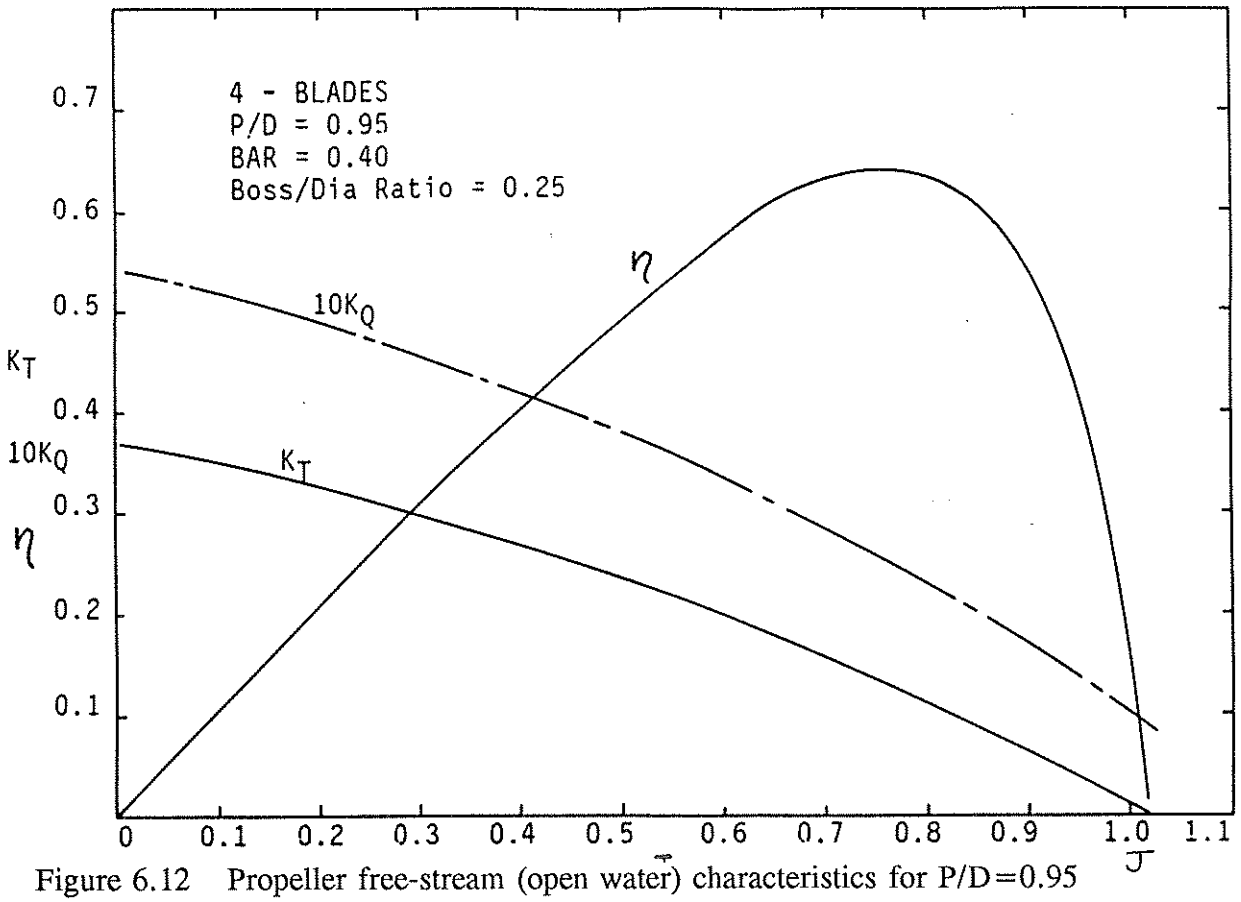


Figure 6.12 Propeller free-stream (open water) characteristics for P/D=0.95

6.10.4 Influence of propeller on rudder performance

Figure 6.13 shows the influence of propeller thrust loading on Rudder No. 2 for the mid-longitudinal separation position (base condition of Table 6.2).

In order to investigate scale effect a number of exploratory tests were carried out at windspeeds of 5, 10, 15 and 20 m/s. It was found that while there was some influence at 5m/s, comparison of the results at 10m/s and 20m/s showed that changes in the principal rudder performance characteristics, such as $dC_L/d\alpha$, C_D, CP_c and CP_s, were not significant. This is illustrated in Figure 6.13 where the results for 20m/s have been superimposed on the results for 10m/s at a J value of 0.94. A windspeed of 10m/s was subsequently chosen for the majority of tests.

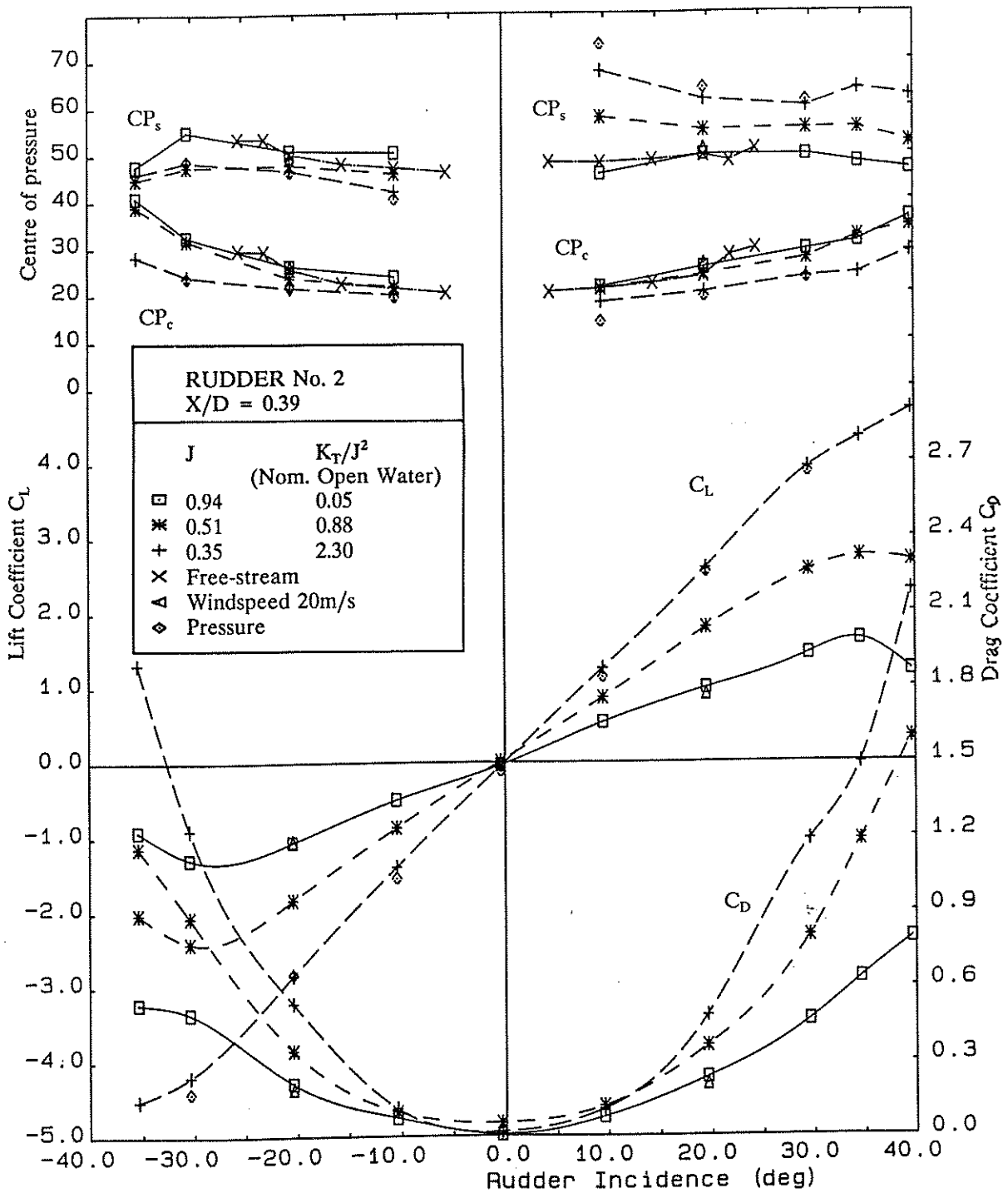


Figure 6.13 Lift, Drag and Centre of Pressure Characteristics: Influence of propeller thrust loading

It can be seen that with decrease in J (increase in thrust loading) the lift curve slope, $dC_L/d\alpha$, increases. These increases are found to be less than those predicted by simple momentum theory.

There is a significant delay in stall angle compared with the results obtained in a free stream where stall occurs at approximately 20° (e.g. Fig. 6.11). The stall angle is no longer the same at positive and negative incidence, stall occurring later for positive incidence. This would appear to be due to the rotational nature of the rudder inflow. It is also noted that stall angle increases with increasing thrust loading.

The rudder drag coefficient (C_D) for all three advance ratios is similar between -10° and +10°. As rudder incidence is increased the drag component due to lift increases rapidly for the lower advance ratios (higher thrust loadings).

Centre of pressure chordwise, CP_c, generally moves forward with increasing thrust loading. CP_c in the presence of the propeller tends to be at or slightly further forward than the free-stream case, particularly at the higher thrust loading.

Centre of pressure spanwise, CP_s, moves toward the rudder tip with increase in thrust loading at positive rudder angles, whilst for negative rudder angles, CP_s moves toward the root with increase in thrust loading.

Figures 6.14a to 6.14d show the spanwise distributions of load for Rudder No. 2 for the free-stream and advance ratios of 0.94, 0.51, and 0.51. Spanwise integration of these distributions lead to values close to the total force measured directly by the dynamometer, as seen in Figure 6.13. It follows that these diagrams provide an accurate picture of the spanwise distribution of load over the rudder.

At high J, low propeller thrust loading (Fig. 6.14b), the spanwise distribution shows only small differences from that expected for a rudder in a free-stream (Fig 6.14a). The development of a tip vortex is observed (e.g. as discussed in Ref. [6.9])

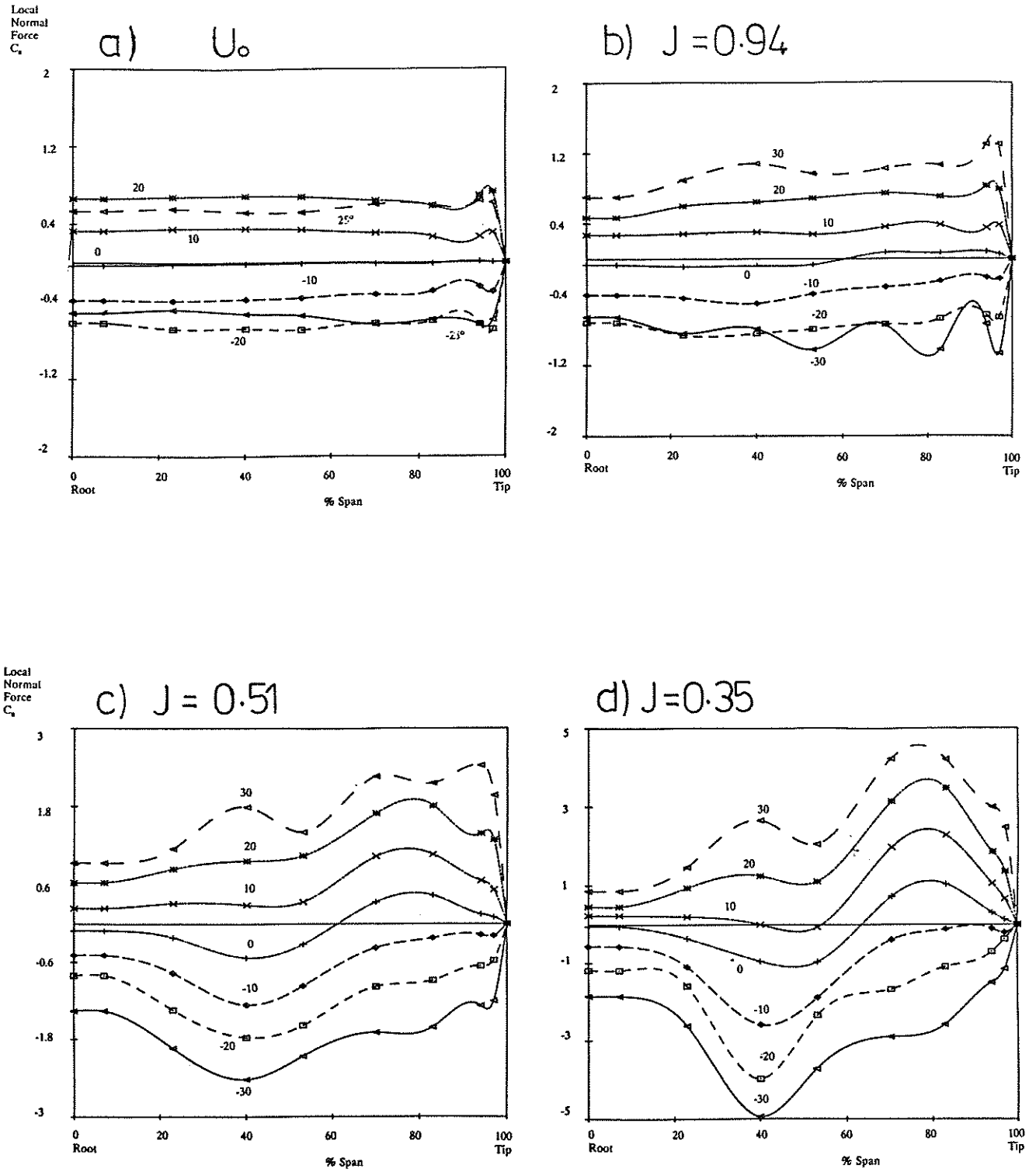


Figure 6.14 Spanwise load distributions Rudder No.2, X/D=0.39

which becomes prominent at higher angles of attack. At zero rudder incidence a flow angle offset is observed between the tunnel floor and the propeller race (<0.2m span).

Figures 6.14c and 6.14d illustrate the effect of increased propeller loading as J is reduced. These figures clearly illustrate the asymmetric nature of the load distribution which results from the angular flow change induced by the propeller. It is also noted that at high J values the relative magnitude of loading outside the propeller slipstream (<0.2m span) is of the same order as that within the slipstream (0.2 to 1m span) whereas at the low J value the loading within the propeller slipstream dominates the loading of the rudder.

6.10.5 Influence of rudder on propeller

Figure 6.15 plots the variation in propeller thrust loading (K_T/J^2) against a base of rudder incidence for the three advance ratios tested. The propeller free-stream characteristic value is shown for each condition. It is seen that at low J the operating condition of the propeller is more sensitive to the presence of the rudder than at high J. Overall the blocking effect due to the rudder's presence is more significant than the diversion of flow due to rudder incidence. The increase in propeller thrust coefficient reflects the slowing down of the flow passing through the propeller caused by the rudder's presence.

6.10.6 Discussion

The results obtained are consistent with the work of others reviewed in Chapter 1. The advantage of carrying out the tests in the wind tunnel with low blockage and with large models to allow the easy measurement of quasi-steady surface pressure has been demonstrated.

In particular, the spanwise load distributions indicate the effect of both axial and rotational rudder inflow velocities induced by the propeller. They also illustrate the

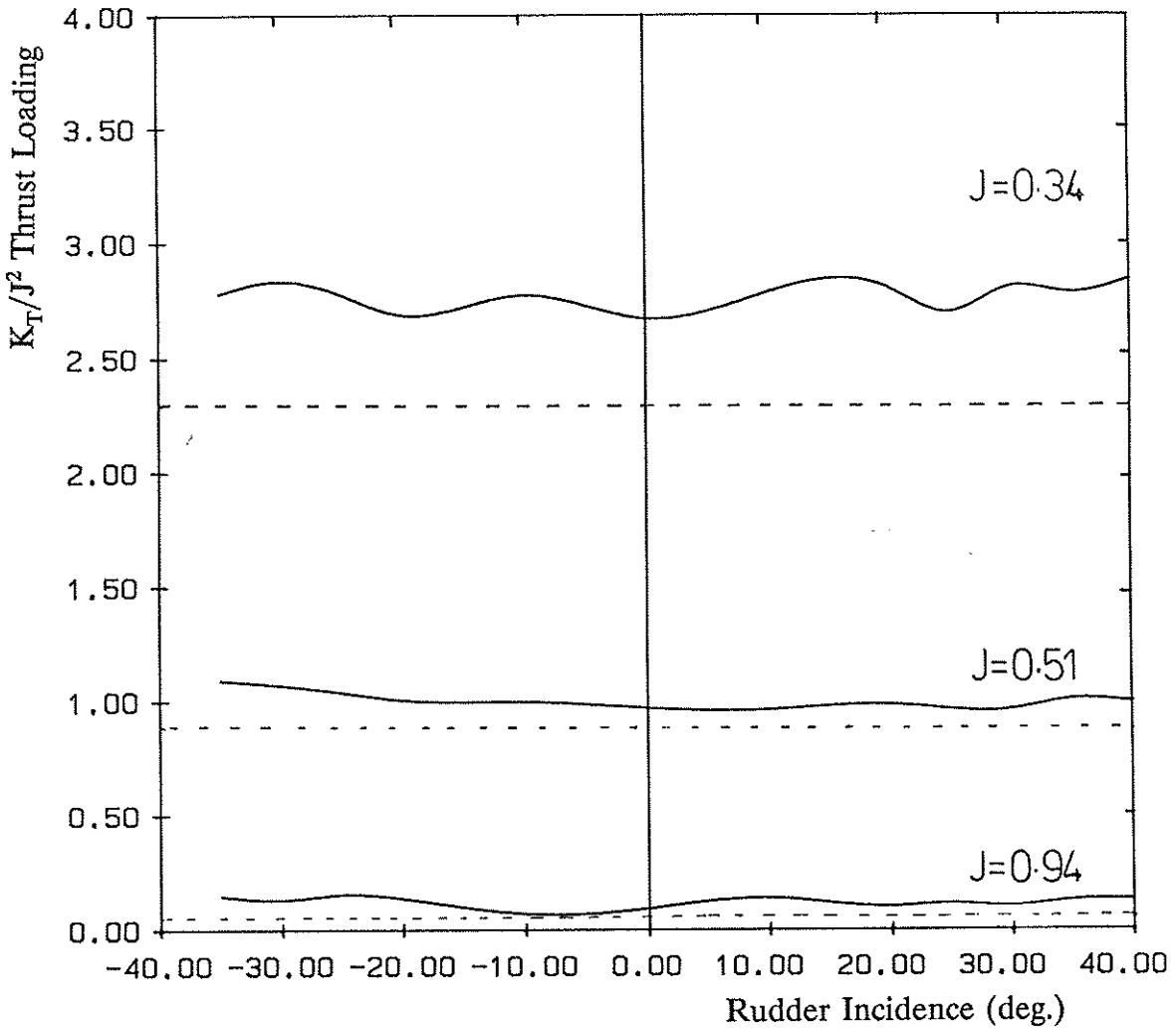


Figure 6.15 Variation of propeller thrust with incidence

presence of the rudder tip vortex identified in earlier free-stream rudder tests. The strength of the tip vortex is increased due to the accelerated flow induced by the propeller.

Only the results for a single rudder-propeller arrangement have been presented so far. The complete set of data from the three wind tunnel sessions are given in Molland and Turnock [6.15,6.18]. The validation of the lifting surface panel method using the experimental data is described in Chapter 7. In Chapter 8 the results from the parametric study using both experimental and theoretical information are given.

6.11 Conclusion

A rig has been developed which allows the influence of a propeller upstream of a rudder to be accurately investigated. An automated data acquisition system allows forces, moments and surface pressures on the rudder to be measured and provides information about the thrust and torque of the propeller. The rig can be separated to carry out free-stream tests on both rudders and propellers.

The rig allows parametric studies to be made of rudder-propeller interactions which provides a basis for developing theoretical and empirical formulations for use in ship manoeuvring calculations.

The results obtained from the rig have been verified against previously published data. In particular, it was shown that:

There is an increase in rudder lift curve slope with increase in propeller thrust loading which follows reasonably well documented trends. These increases are less than those predicted by simple momentum theory (discussed in greater detail in Chapter 8) indicating that momentum theory alone does not adequately model the complex physics of the flow.

There is a significant increase in stall angle, compared with free-stream, when the rudder is working aft of the propeller. This is also documented elsewhere, but reinforces the need to appreciate that very large rudder angles can be usefully employed in the lower speed/high thrust loading manoeuvring situation.

There is some movement in the centre of pressure with changes in propeller thrust loading. Chordwise centre of pressure, CP_c, moves forward with increasing thrust loading and in most cases is forward of the free-stream CP_c. Spanwise centre of pressure, CP_s, moves towards the tip for one direction of rudder incidence whilst moving towards the root for opposite incidence.

CHAPTER SEVEN
VALIDATION OF A LIFTING SURFACE PANEL METHOD FOR MODELLING RUDDER-PROPELLER INTERACTION

7.1 Introduction

The validation of a method for modelling the interaction between a rudder and propeller is the subject of this chapter. A brief description is given of the process of successful validation of a CFD algorithm. The parallel algorithms detailed in Chapter 5 are the basis of the lifting surface analysis carried out. Experimental results obtained from the wind tunnel test programme are used to validate firstly a lifting-surface model of a free-stream all-movable rudder (Rudder No. 2), secondly to model the open-water performance of the four-bladed propeller (modified Wageningen B4.40), and finally a representative rudder-propeller geometry. The representative geometry chosen was the mid-longitudinal separation of Rudder no. 2 (X/D=0.39) and propeller, with no lateral separation (Y/D=0.0), and the maximum height of the propeller tip coincident with the rudder tip Z/D=0.75.

The development of the interaction velocity field method is described. An aim of the validation exercise was to discover the minimum number of panels necessary to obtain reliable rudder and propeller force characteristics. Also, the minimum number of panels to adequately define the interaction velocity field was found. The minimisation of computational effort allowed the parametric studies, described in the following chapter, to be carried out within a reasonable time-scale.

7.2 Validation of Numerical Algorithms

A mathematical model of a physical process generally involves a degree of approximation. In using such a model it is necessary to appreciate the confidence with which the model can be used. In the same way in which there is always error involved in the acquisition of experimental data, numerical modelling gives rise to uncertainty in the obtained answer. The process of validation attempts to quantify these

uncertainties.

The precise definition as to what constitutes a validated CFD code is the subject of debate at the present time. The process of code validation can be seen as a series of stages.

1) Verification of the code implementation against the underlying mathematical formulation. This is to ensure the code is free of error due to mistakes in expressing the mathematics in the particular computer language used. Ideally the comparison should be made against an analytic solution although often the comparison can only be made with other numerical codes.

2) Investigation of the independence of the solution from numerical parameters. The most common form of dependence is on the density of the grid of points at which the governing equations are solved. Normally, the number of grid points (or panels) is increased until the solution does not change. For iterative techniques which use a convergence criterion the dependence of solution on its value has also to be investigated.

3) Comparison of numerical and experimental data. This is the most tendentious area of code validation. As the majority of fluid dynamic codes are an approximation to the actual physics of the flow there will be differences between the experimental and numerical results. Experimental data should have a specified accuracy. This should then allow the difference between experiment and theory to be quantified. However, in many codes some degree of empiricism is used to adjust the numerical model to fit specific experimental data. The extent to which such an empirically adjusted model can be said to be valid for cases run at different conditions requires careful consideration.

In this work the aim has been to limit the amount of empiricism used in the modelling. This should then allow more confidence to be placed in the results obtained outside the conditions used for comparison. However, there will be differences

between the numerical model and the experimental results which arise from the approximation of the Navier-Stokes equations by an inviscid, irrotational and incompressible formulation using discrete panels.

The verification of the lifting surface panel method against both analytic and other lifting surface solutions was described in Chapter 4. In this Chapter a validation process is carried out for the free-stream modelling of the rudder and propeller and then for their mutual interaction. A limited number of examples of solution convergence against panelling density are given. The convergence behaviour of panel methods having been well documented. Detailed comparison is made with the experimental data for a limited series of cases. In these it is shown that the method identifies the correct physical trends and thus can be used with confidence for interpolation between the values of experimental parameters tested and for a limited degree of extrapolation.

7.3 Free-stream Rudder Performance

7.3.1 Introduction

A ship rudder, for the purpose of analysis, can be considered to be a low aspect-ratio wing. Difficulties arise when lifting-line theory is applied because of large cross-flow velocity components developed, especially at high incidence. Molland[7.1] developed empirical corrections to allow the lifting-line modelling of both semi-balanced skeg and all-movable rudders. To avoid such empiricism in modelling rudder-propeller interaction a lifting-surface panel method has been implemented. Various rudder test cases were run using the developed program and the results compared with the wind tunnel tests for a free-stream rudder. Comparisons were made for Rudder No. 2 which has a 1m span, chord of 0.667m and a constant NACA0020 section ($t/c=0.2$).

7.3.2 Lifting Surface Model of Rudder

The numerical definition of Rudder No. 2 specifies three identical NACA0020 sections, with ordinates taken from Abbott & Van Doenhoff[7.2], at heights of 0.0m, 0.5m, and 1.0m above the wind tunnel floor. The wake sheet trails downstream in the inflow direction. The flexible geometry definition allows both the number of span-wise and chord-wise panels (Nt, Ns), and the local panel density to be varied. Figure 7.1 shows three panelling arrangements for Rudder No. 2 each with a total of 400 panels: (a) shows 50 chordwise and 8 span-wise panels of uniform size; (b) shows 25 chordwise and 16 span-wise uniform panels; and (c) 25 by 16 panels with sinusoidal clustering at the leading and trailing edge and rudder tip. The rudder model uses a reflection plane at the rudder root to account for the flow symmetry due to the wind tunnel floor. This neglects the wind tunnel boundary layer and approximates to the flow symmetry which occurs on a ship due to the presence of the hull.

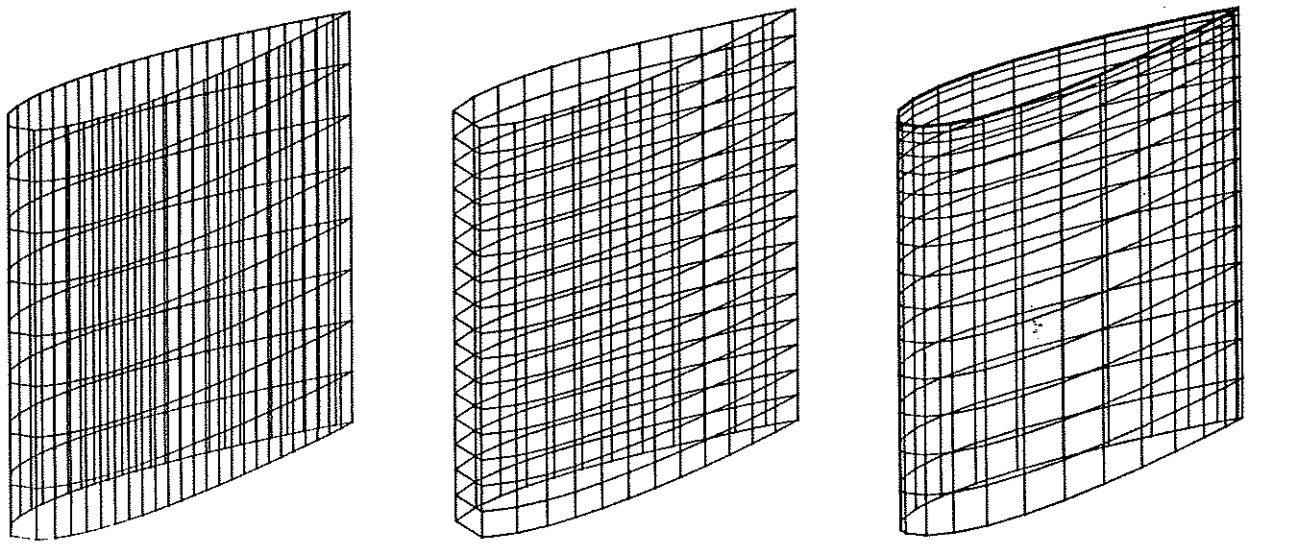


Figure 7.1 Three panelling arrangements for Rudder No. 2

Different rudder incidences are modelled by applying a rotation about the z-axis with the pivot point set at 200mm from the leading edge. This corresponds to the location of the rudder stock on Rudder No. 2.

Changing the scale vector S allows the experimental Rudder No.'s 3,4,5 and 6 to be generated from the same basic configuration.

7.3.3 Comparison of Model and experiment

For a rudder incidence of 9.6°, sensitivity studies were carried out into the effect of panel distribution, local panel density, and total number of panels. The maximum number of panels used was 400, which corresponds to the maximum available on the 4 transputer Ship Science system (SSTS) upon which the flow solver was developed. Figure 7.2 shows the lift C_L and drag C_D variation against the total number of panels for two cases

1) a constant number of spanwise panels, Ns=8

2) a constant number of chordwise panels, Nt=25.

These figures were obtained with sinusoidal panel clustering at the leading edge, trailing edge and tip. The straight line represents the wind tunnel experimental value. It can be seen that there is only a limited convergence with increasing number of spanwise strips. However, increasing the number of panels per spanwise strip causes a gradual convergence towards the experimental value. This is due to the increased panel resolution around the area of rapidly changing curvature at the leading edge. The drag is sensitive to small changes in panel geometry which accounts for the fluctuation about the experimental value. Four hundred panels is not sufficient to converge the value of lift or drag, but it was considered to be good enough for assessing the manoeuvring performance and for finding parametric trends. It should be noted that the lifting-surface panel method only models potential flow. Therefore, because of the absence of a viscous boundary layer the lift will be overpredicted and drag underpredicted.

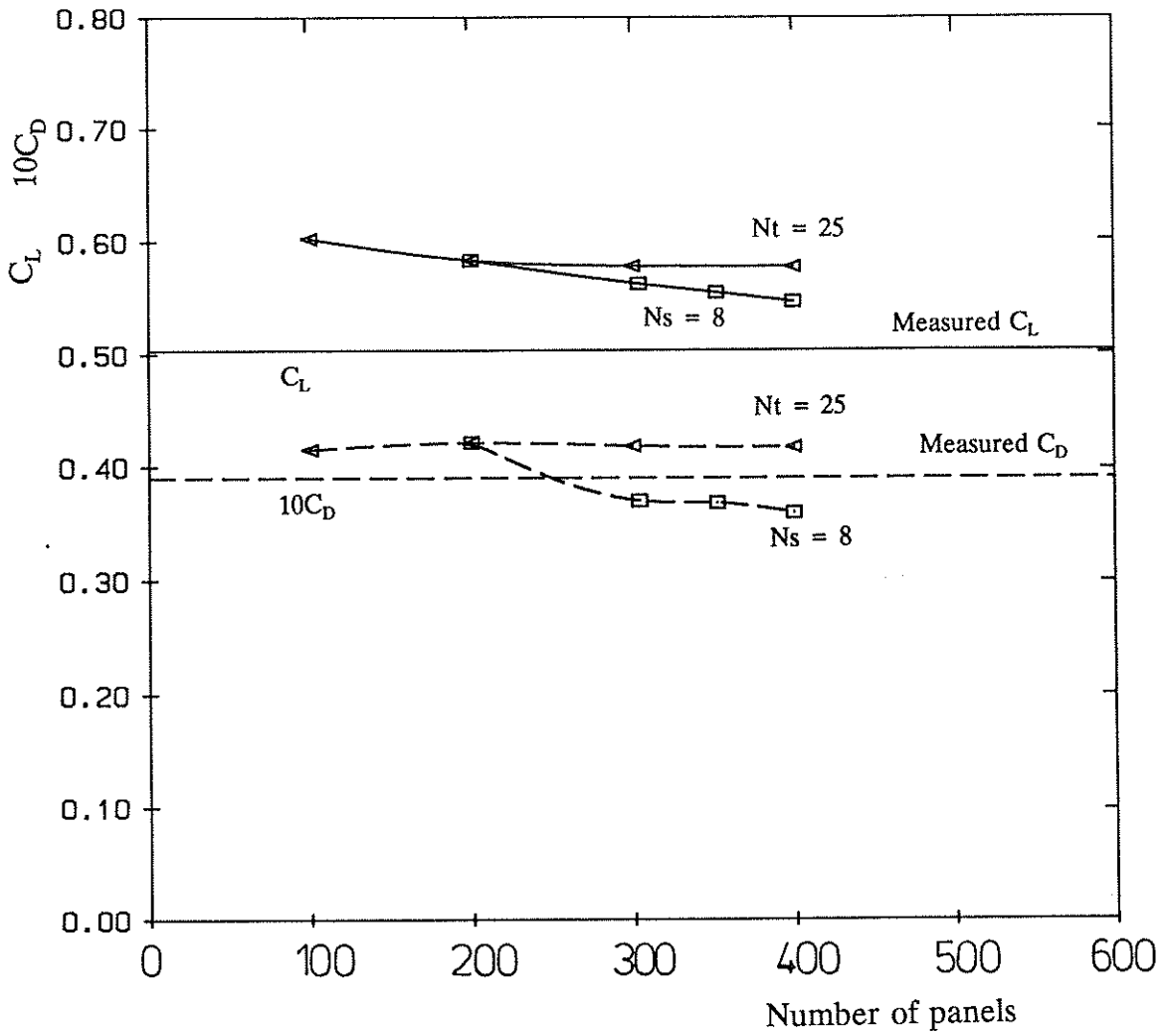


Figure 7.2 Variation of predicted rudder lift and drag with the total number of surface panels

For the investigation into the sensitivity of the final solution to total number of body panels the wake length was held constant at a value of 5 rudder chords. Each wake strip was equally divided into 50 panels. This value was chosen because, as shown in Figure 7.3 (Nt=25, Ns = 16), after a wake length of about 4 rudder chords the value of lift and drag converged. A constant panel density of 10 per rudder chord was used. For the cases described a convergence criterion was applied that the maximum non-dimensional trailing edge pressure load was less than 0.01.

$$\Delta Cp_{t.e.} < 0.01 \qquad [7.1]$$

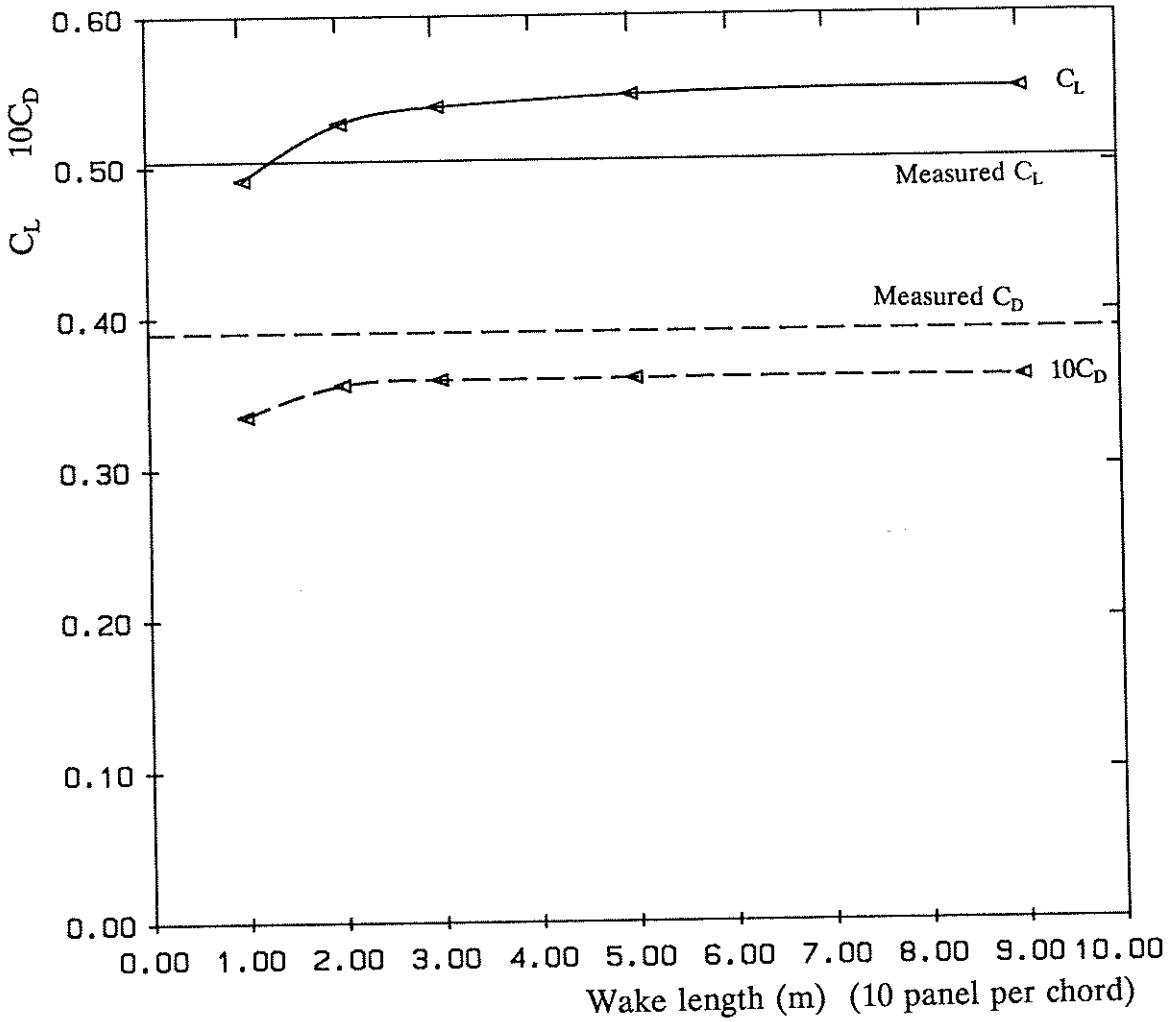


Figure 7.3 Variation in rudder lift and drag with increase in wake length

Figure 7.4 compares the surface pressure values for this numerical test case with experimental values. The numerical test had 50 chordwise by 8 spanwise panels with sinusoidal clustering. A linear interpolation was applied to the theoretical values so that the spanwise location of theory and experiment corresponded. The lifting-surface values closely follow the form of the experimental points although the absolute value of pressure is always overpredicted by the lifting-surface. At the spanwise section closest to the rudder root the lifting surface values are significantly larger. It is thought likely that this is due to the boundary layer flow along the wind tunnel floor which reduces the inflow velocity near the wall leading to reduced pressures and reducing the

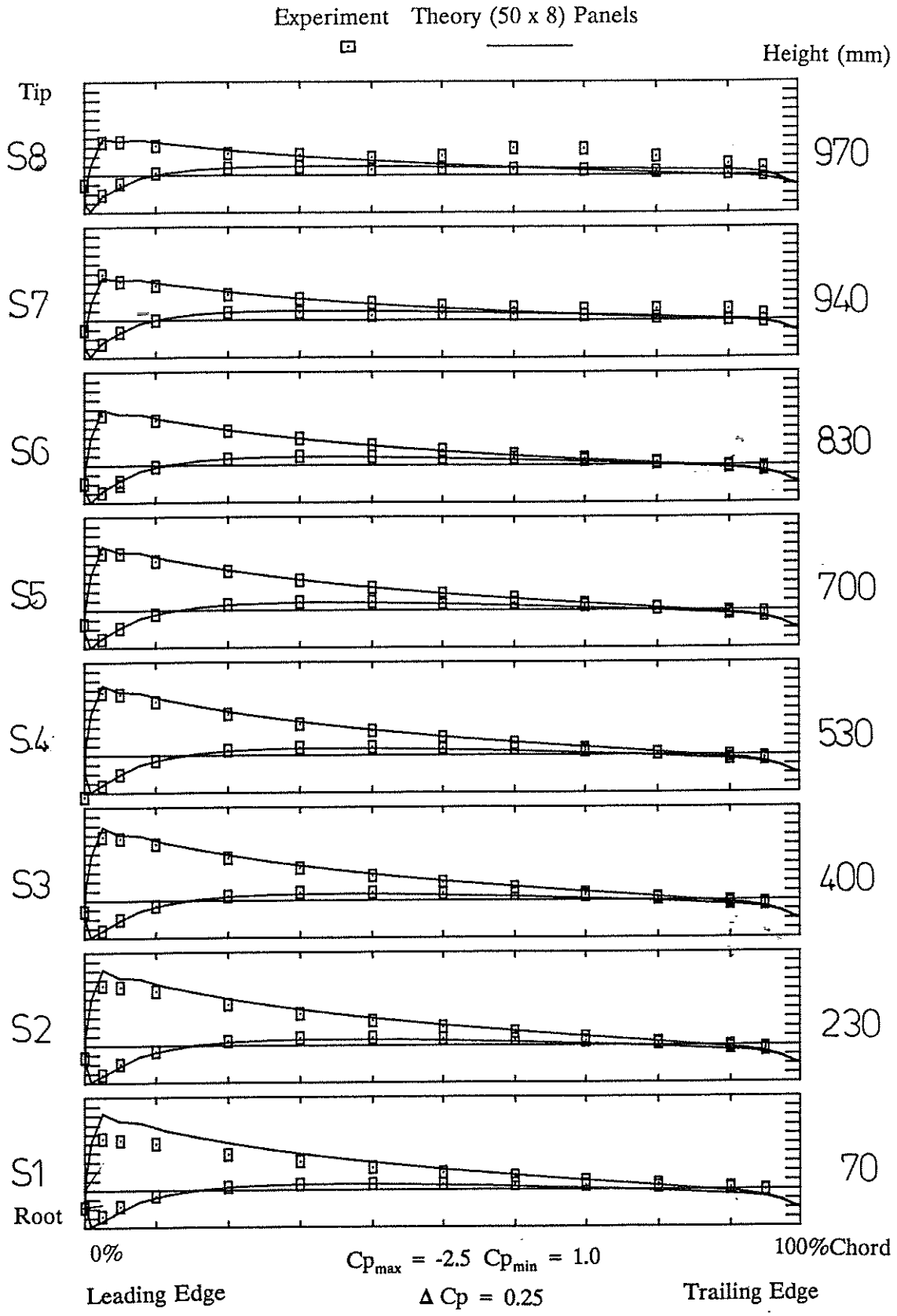


Figure 7.4 Comparison of experimental and numerical chordwise pressure distributions at 8 spanwise positions for rudder no. 2 at +9.6°

effective aspect ratio of the rudder. Also, at the spans nearest the rudder tip there is a divergence between experiment and lifting-surface towards the trailing edge of the suction side. This is probably due to the presence of a tip vortex (Molland[7.1]). The presence of the trailing vortex is indicated by the relatively high negative Cp values on the rearward portion of the suction side of the rudder. The effect is not modelled by the lifting-surface panel method. Molland obtained good prediction of the tip vortex using an empirical relationship for the tip vortex strength.

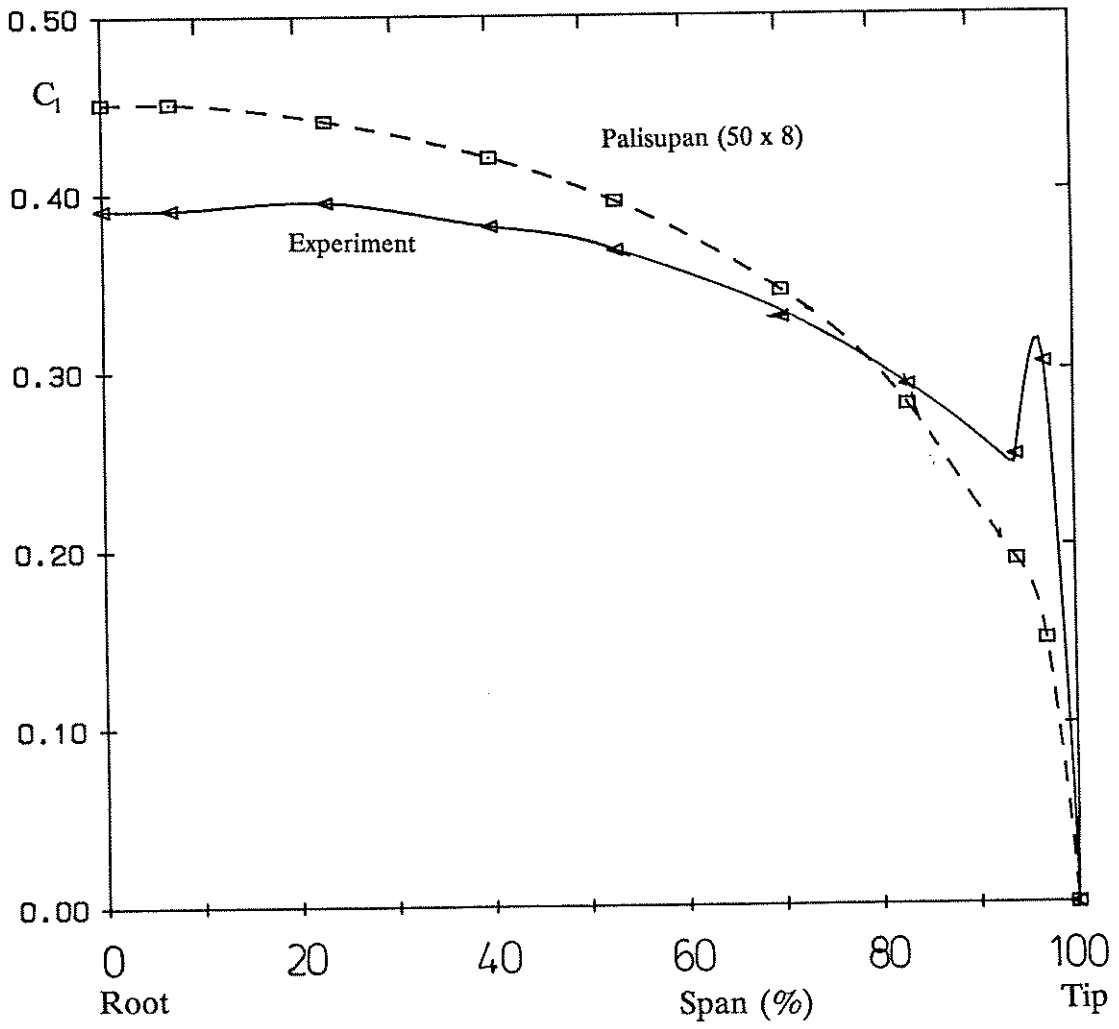


Figure 7.5 Comparison of experiment and theory for the variation in spanwise distribution of local lift coefficient C_l for rudder no.2 at 9.6°

Chordwise integration of the surface pressure allows the local sectional lift coefficient C_l to be evaluated. For a rudder incidence of +9.6°, Figure 7.5 compares the spanwise distribution of local lift for the experimental and lifting-surface result. Overall a good comparison is obtained. As noted in the surface pressure comparison the lifting surface overpredicts the pressure difference and hence lift. The difference is due to a combination of viscous effects and the restriction of surface panels to 400. The development of the viscous boundary layer alters the effective shape of the aerofoil and hence surface pressure distribution thus decreasing lift and increasing drag. The effect of the shedding of a vortex from the square edged rudder tip is clearly seen in the experimental result. It induces a characteristic hump to the load distribution close to the tip. For the purposes of rudder-propeller interaction modelling this effect has been neglected. Although it is of significance, especially at high incidence, it was considered unlikely to influence the overall trends.

Finally, Figure 7.6 compares C_L, C_D, CP_s, and CP_c for a range of rudder incidence between -25° and 25° up to and including stall. As would be expected for an inviscid method the gradual onset of stall and subsequent reduction in lift at higher incidence is not predicted. However, the overall comparison of lift and drag characteristics is good. The numerical lift curve slope at 0° is 10% greater than the experimental value. The chordwise position of centre of pressure is fairly well predicted but because of the overprediction of lift by the lifting-surface method and the physical presence of a tip vortex the numerical prediction of the spanwise centre of pressure is about 5% of span nearer the root. As mentioned previously, the difference is mainly due to neglecting the effect of the boundary layer on the pressure distribution (viscous pressure drag), tip vortex effects, the effective aspect ratio being less than twice the geometric aspect ratio, and the flat plate assumption for skin friction drag.

7.3.4 Velocity field ahead of rudder

The flow upstream of a rudder is affected by its presence and a knowledge of this upstream influence is necessary for determining the influence of the rudder on the

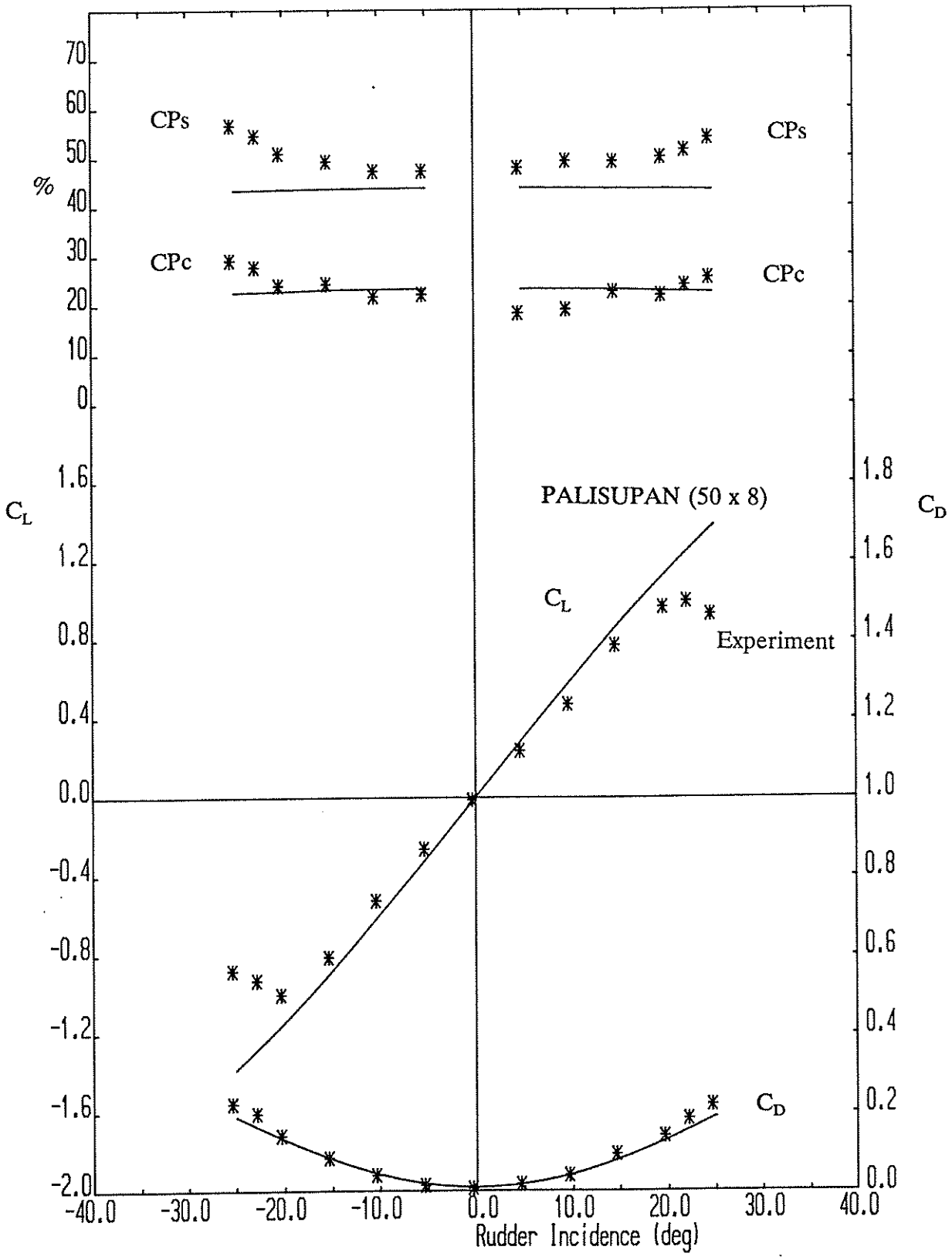


Figure 7.6 Comparison of lifting-surface and experimental lift, drag and centre of pressure characteristics for rudder no. 2 in free-stream

propeller. Using the solution of the lifting-surface the velocity field upstream of the rudder can be obtained by the summation of all the individual velocity influences due to the body source/dipole panels and the wake dipole panels. This process defines an interaction velocity at any point in the field around the rudder. The total velocity at any point is the sum of this interaction velocity and the inflow velocity. To illustrate this, Figure 7.7 shows a velocity vector plot at a span of 0.5m around Rudder No. 2 at an incidence of 9.6°. The characteristic upwash upstream of the rudder and flow acceleration around the rudder upper surface can be seen. The flow within the rudder was found to be at the free-stream value which indicates that the internal potential field is constant and equal to zero as required by the perturbation potential theory. The velocity field was generated by finding the velocity on a regular mesh of points. Close to the body surface there are localised effects due to the modelling of the surface as constant strength quadrilateral panels.

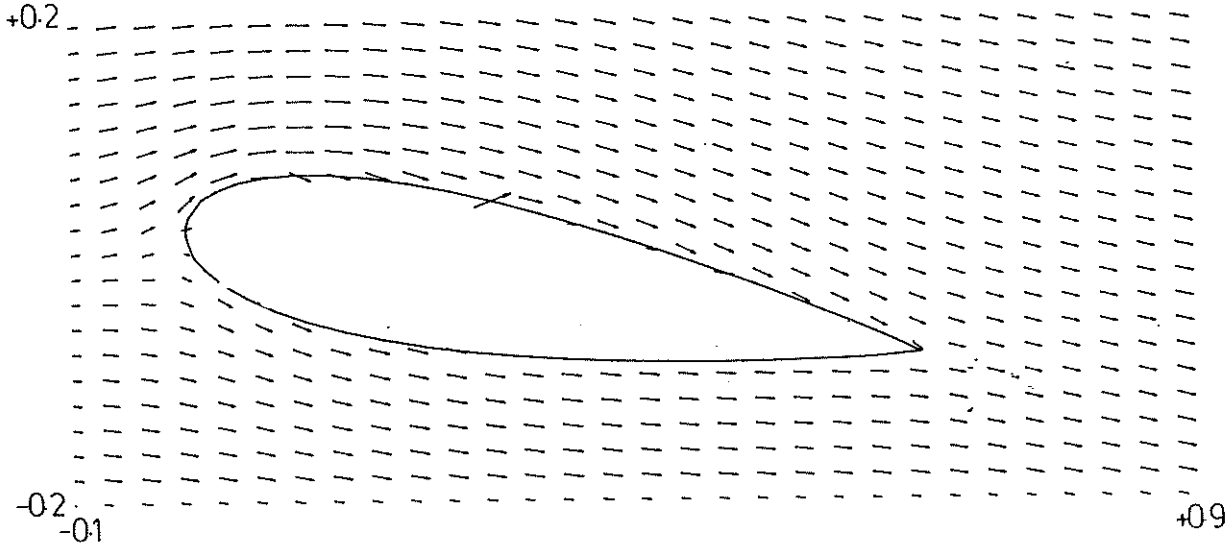


Figure 7.7 Velocity field at mid-span of rudder no. 2 at incidence of 9.6°

To account for the interaction of the rudder at the propeller plane the interaction velocity is found on a circular disk of points. The location of the circular disk corresponds to the actual location of the propeller relative to the rudder. Circumferentially averaging the velocity of points at a given radius generates the spatial

average inflow to the propeller. It was found that a minimum of 10 radial and 20 circumferential points gave adequate resolution of an average velocity field. As an illustration of this process for modelling the interaction Figure 7.8 shows the radial

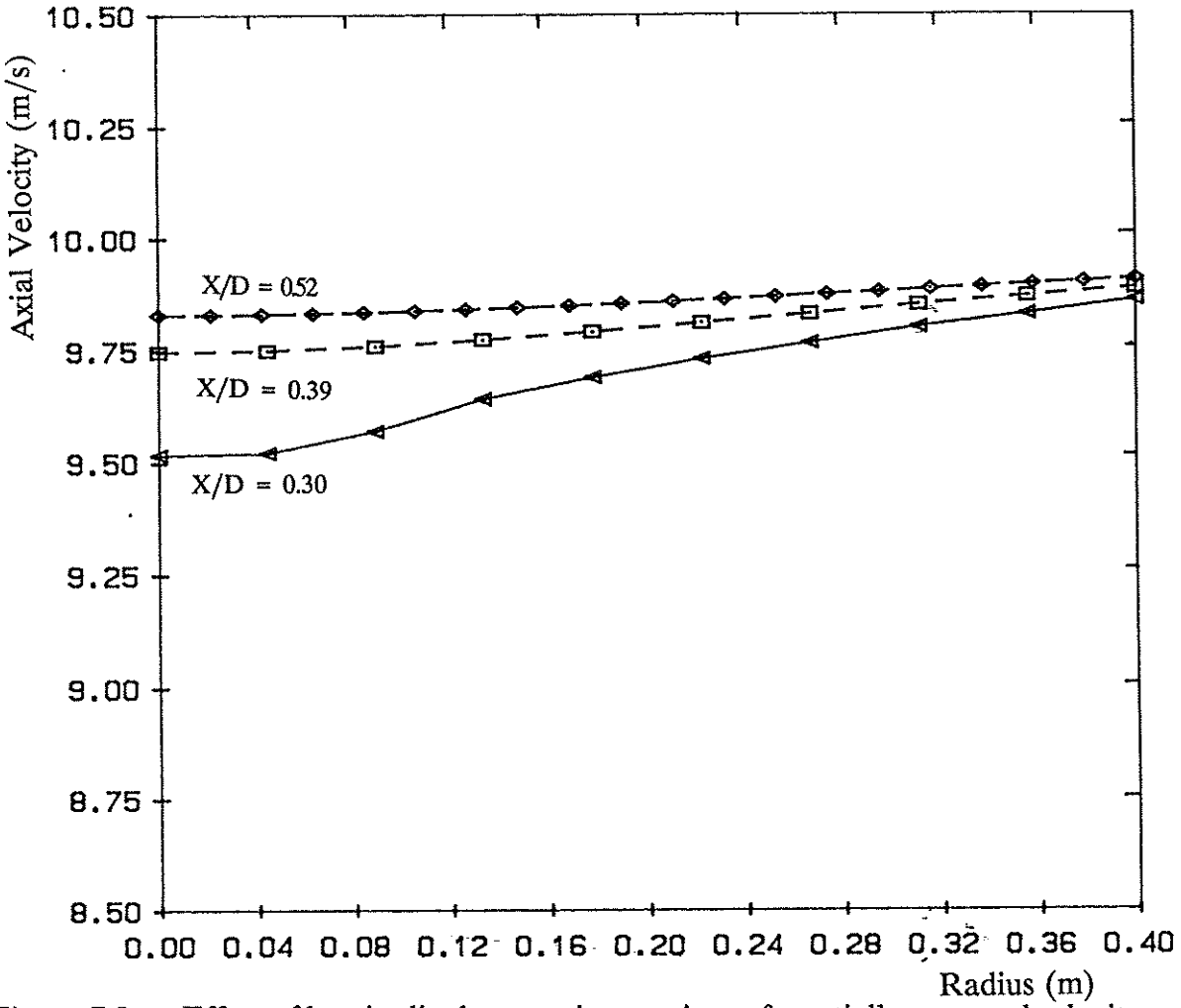


Figure 7.8 Effect of longitudinal separation on circumferentially averaged velocity upstream of Rudder No. 2

variation in average axial velocity for three rudder-propeller longitudinal separations corresponding to X/D= 0.30, 0.39, and 0.52. The greatest drop off in axial velocity occurs at the propeller hub (r<100mm). As expected, the smaller the separation the greater the velocity defect. The variation in the average radial and circumferential components of velocity are small Vr<0.25m/s and Vw<0.05m/s for a free-stream velocity of 10m/s and can be neglected when modelling the interaction. It should be noted that these components arise from the asymmetric location of propeller axis (Z=0.6m) relative to the flow over the rudder.

7.3.5 Summary

The comparison of experimental and lifting-surface results is satisfactory for rudder incidence up to stall. The method does not predict tip loading due to the shed vortex or the effect of a viscous boundary layer and wake. Bearing the above restrictions in mind and the limitation on the number of panels available it is considered that the prediction of lift-curve slope within 10% of the experimental value is acceptable for the purposes of investigating rudder-propeller interaction. The results can be improved through the use of empirical corrections for more detailed study of rudder performance but would then restrict the use of the method.

The circumferentially averaged interaction velocity field for use as a propeller inflow can be adequately generated from 200 velocity field points.

7.4 Open-water performance of propeller

7.4.1 Introduction

An important part of the developed interaction velocity field method is the accuracy with which the velocity field downstream of the propeller can be modelled. In order to obtain this velocity field it is first necessary to validate the open-water performance of the lifting-surface code against the free-stream tests of the experimental propeller. The propeller and hub geometry used for the lifting-surface model exactly correspond to those used in the experimental investigation. However, the influence of the ground board (reflection plane) cannot be included as this would result in a spatially varying propeller inflow. Modelling this requires a time accurate model of all four propeller blades. This is feasible with the current formulation but would require large computational effort due to the need for extra surface panels and modelling at different angular positions. For similar reasons the propeller support structure present in the wind tunnel tests is ignored. A quasi-steady model which assumes an identical

flow across all four blades has therefore been used.

7.4.2 Lifting surface model of propeller

The whole four-blade propeller and boss geometry is rotationally symmetric. Therefore, only one blade and a quarter segment of the boss need be modelled. The overall geometry is split into two components a non-lifting body (cylindrical boss) and lifting-surface (blade).

The numerical definition of the modified Wageningen B4.40 blade that was used to NC machine the model propeller blade was also used to generate sections at different radii out along the propeller blade for creating the surface panel data. The use of the same base propeller data ensures an identical geometry between lifting-surface model and experiment. The method developed also permits any standard table of propeller offsets to be used to generate a numerical model. The propeller pitch can be set at this stage by altering the angle of rotation of the blade about its generator. Figure 7.9 shows three projections and an isometric view of the model propeller blade section definition. The trailing wake sheet is specified as lying on a helical surface following a given pitch for each section.

Maitre[7.3] found, especially at high propeller loading, that modelling of the boss was necessary to obtain the correct blade load distribution. For this reason the final numerical model includes the boss. The design profile of the boss is identical to that used in the wind tunnel model except the upstream cylindrical section is shortened to increase the panelling density around the blade root. At the blade root-boss intersection to prevent numerical problems a correct matching of the geometry is required. The quarter segment of the body was set on a helical surface with the geometric pitch of the blade root. To obtain a smooth spline through the boss-blade interface region it was found necessary to use the elliptic grid generator described in Appendix C. Figure 7.10 shows the elliptically refined grid before fitting around a helical surface. In all, six curves were used to define the boss. The final surface

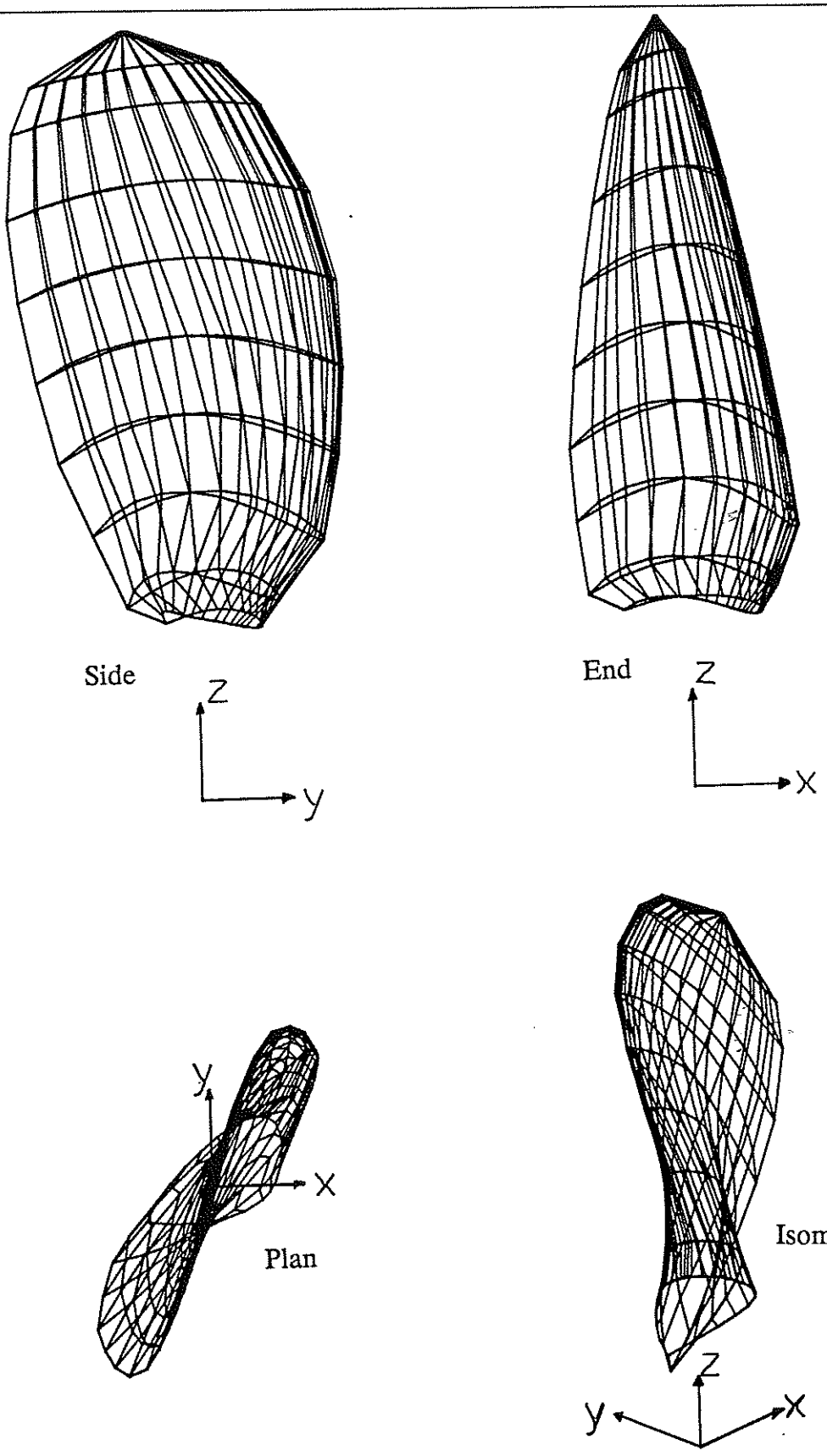


Figure 7.9 Three projections and isometric view of modified Wageninegn B4.40 propeller surface definition

definition geometry is shown, in plan and end on, in Figure 7.11.

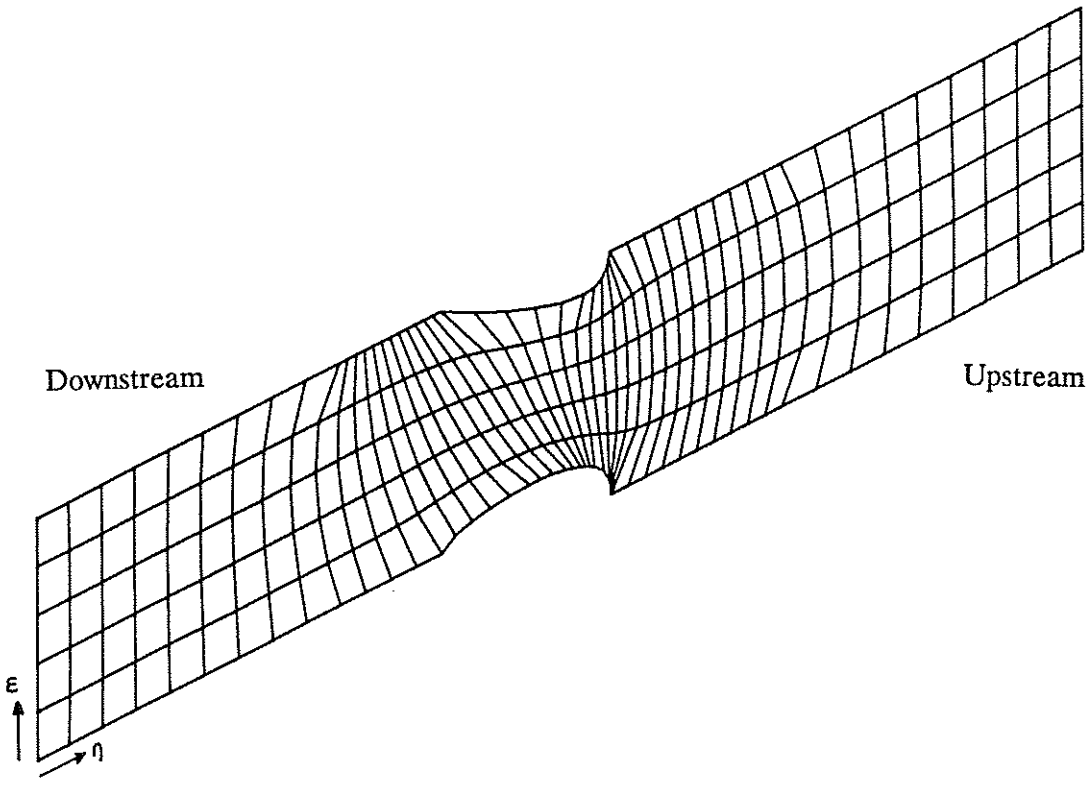


Figure 7.10 Elliptically refined grid for boss definition

Figure 7.12 is an isometric view looking upstream. The complete propeller and boss geometry is shown with 100 boss and 300 blade panels. All the image panels, 1200 in total, are shown using dotted lines.

7.4.3 Comparison of model and experiment

The basic model consisting of 25 by 4 boss panels and 25 by 12 blade panels, was run for a range of advance ratios between 0.2 and 1.0. A constant wind-speed condition equal to the experimental test speed of 10m/s was used. The different advance ratios were obtained by altering the propeller rate of revolution. Figure 7.13 compares the propeller thrust, torque and efficiency characteristics (K_T, K_Q, and η) between the surface panel model and experimental and Wageningen data, for a pitch ratio setting of P/D=0.95. As an initial first step for all the advance ratios tested the wake pitch at all sections was set equal to the geometric pitch of the corresponding blade section.

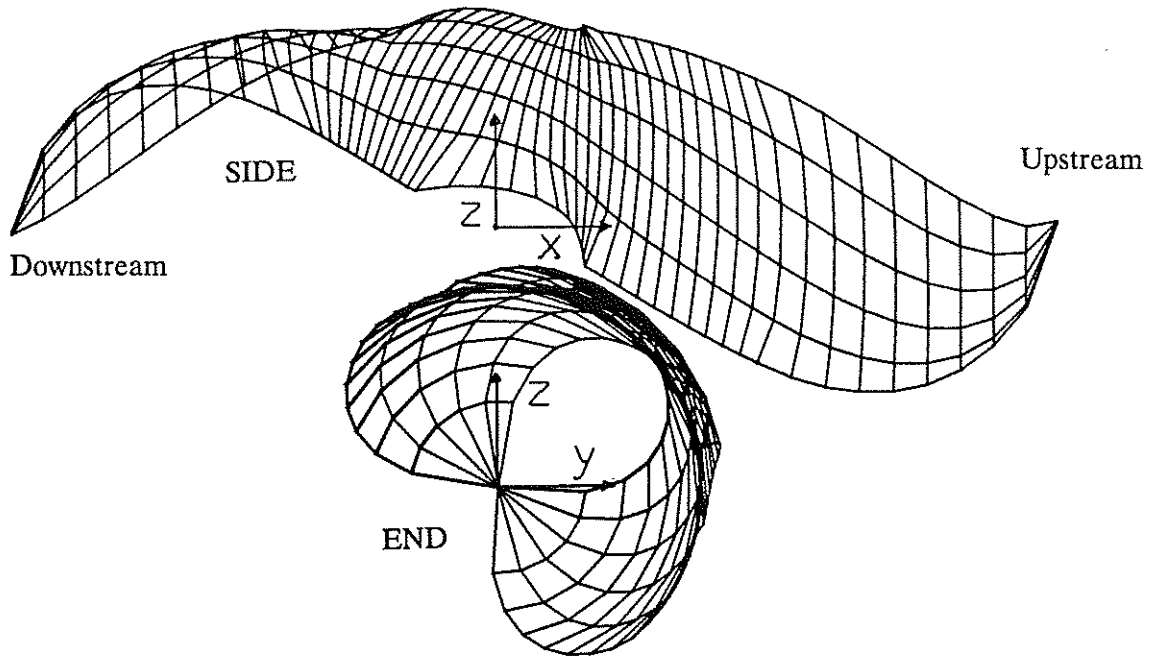


Figure 7.11 Plan and end view of boss surface definition

It can be seen that the efficiency gives good agreement but both thrust and torque are overpredicted. There are three main reasons for this:

1) the radial variation in actual wake pitch is important in determining the actual thrust/torque of a given blade section and the geometric pitch especially at low J, is a considerable over-prediction of the actual wake pitch which will result in higher blade loading;

2) the potential formulation neglects viscous effects (i.e. viscous pressure drag) which become especially important at low advance ratio when there is the possibility of stall/separation across areas of the blade;

3) the relatively low number of surface panels available for modelling the blade and boss.

Sensitivity studies were carried out into the panelling arrangement on the propeller blade alone. Table 7.1 shows the influence of total number of panels and the relative number of panels in the chordwise (Nt) and spanwise (Ns) directions on the propeller thrust coefficient at an advance ratio J of 0.35. It can be seen that the

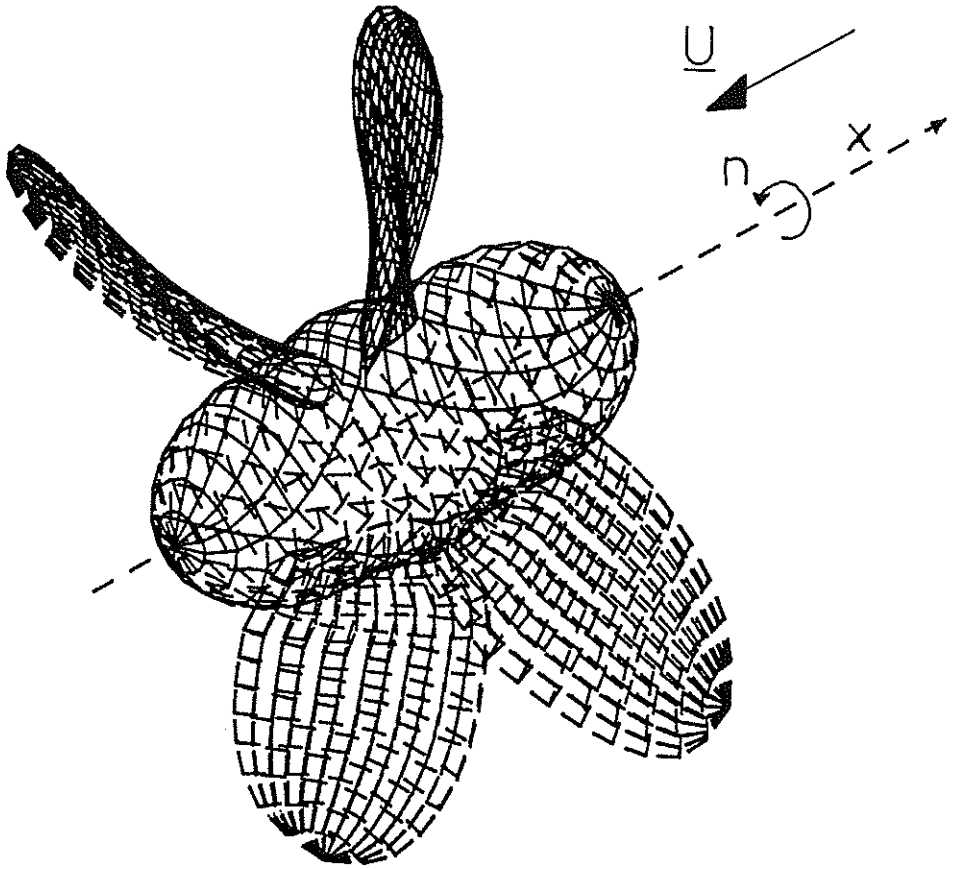


Figure 7.12 Isometric view upstream of 400 panel model of propeller blade and boss, also showing image panels(1200)

Table 7.1 Sensitivity of blade thrust coefficient to panelling arrangement for J=0.35

Total Number of Panels	100	200	300	400	296	360	392
	Variation with Ns				Variation with Nt		
Panel distribution(NtxNs)	25x4	25x8	25x12	25x16	37x8	45x8	49x8
Thrust Coefficient K_T	0.352	0.348	0.341	0.337	0.347	0.347	0.347

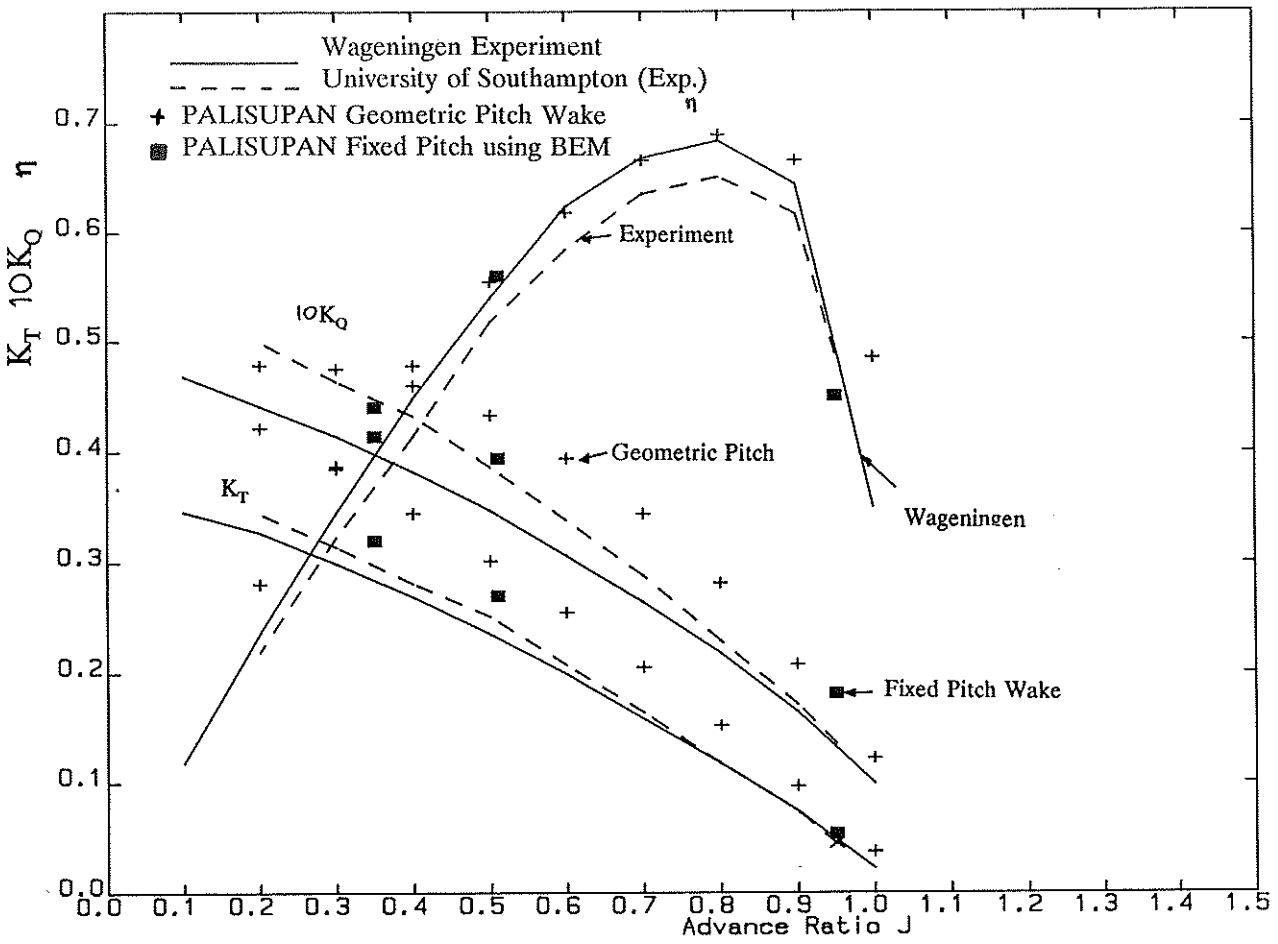


Figure 7.13 Comparison of model, experiment and Wageningen propeller characteristics

propeller, as opposed to the rudder, is more sensitive to the number of spanwise strips. Increasing the number of chordwise panels for a fixed number of strips has relatively little effect. It was found that, due to the asymmetric shape of the back and face of the propeller blade, a constant panel size rather than sinusoidal clustering gave the best result.

The effect of the total wake length on the performance of the propeller blade was quite marked, as is tabulated in Table 7.2. These values are for propeller blade thrust at a J of 0.35 and with 25 wake panels per metre at the propeller geometric pitch of 0.95. It can be seen that the thrust decreases with increasing wake length. Due to the curved nature of the wake the density of wake panels influences the wake thrust.

It was found that a panel density of at least 25 panels per metre of wake was required for convergence. For a fixed wake downstream length of 2000mm, increasing the number of wake panels from 25 to 100 reduced the propeller thrust by 1.5%. At present the number of wake panels per strip is limited to 100 and therefore the maximum possible wake length was restricted to 4000mm. Maitre[7.3] carried out a similar exercise and used a wake length of 8 radii. Therefore, a length of 10 radii(4000mm) was deemed acceptable.

Table 7.2 Effect of wake length on propeller thrust

Wake length	Number of panels	Propeller Thrust K_T
1000mm	25	0.345
2000mm	50	0.341
4000mm	100	0.337

The adaption of a propeller's wake to follow the stream surface which gives zero pressure loading across the wake is fraught with difficulties. The principal source of these problems is the non-physical modelling of the wake as an infinitely thin sheet of dipole panels. Recently, Maitre[7.3] demonstrated that wake adaption for a lifting-surface panel method would converge to a physically realistic solution. However, constraints based on empirical evidence were still applied in the region close to the blade root. Lee[7.4] used a fixed wake shape developed from empirical measurements and numeric analysis to obtain close agreement. In this work the structure of the wake is important not only in determining the correct flow over the propeller but also in specifying the velocity field downstream of the propeller for use as an inflow to a rudder.

The process of wake adaption is computationally intensive. involving the regeneration of the influence coefficient matrix and iterative solution after each wake adaption. Maitre[7.3] found that at least 20 iteration cycles were required to achieve

convergence. In this work, due to the current limitations on body surface and wake panel numbers it was found that the method was unstable except for very short total wake lengths. Therefore, for the modelling of rudder-propeller interaction it was decided to use a fixed wake. The specification of the wake pitch at a given radius was based on the blade-element momentum results for axial and circumferential velocity at the propeller plane derived by Molland[7.5] for the velocity downstream of the propeller. These were used to specify a wake pitch at a given radius. Figure 7.14 shows the radial distribution of wake pitch for the three experimental advance ratios of 0.35, 0.51, and 0.95 tested. It was found that using these pitch ratios improved the agreement between experiment and theory, especially at the lower advance ratio. The improved value of thrust, torque and efficiency are plotted on Figure 7.13 for comparison with the values obtained with a fixed geometric pitch. The sensitivity of propeller thrust and torque to wake pitch is demonstrated in Table 7.3. This shows the effect of different values of a radially constant wake pitch for an advance ratio of 0.35. It is interesting to note that the pitch of 610mm corresponds to the average pitch for the radially varying pitch. The fixed wake shape has no downstream radial contraction applied.

Table 7.3 Effect of a fixed pitch wake on propeller characteristics at J=0.35

Fixed Wake Pitch	Thrust K_T	Torque K_Q
823mm	0.333	0.0424
750mm	0.326	0.0420
700mm	0.319	0.0418
610mm	0.306	0.0411
575mm	0.299	0.0407

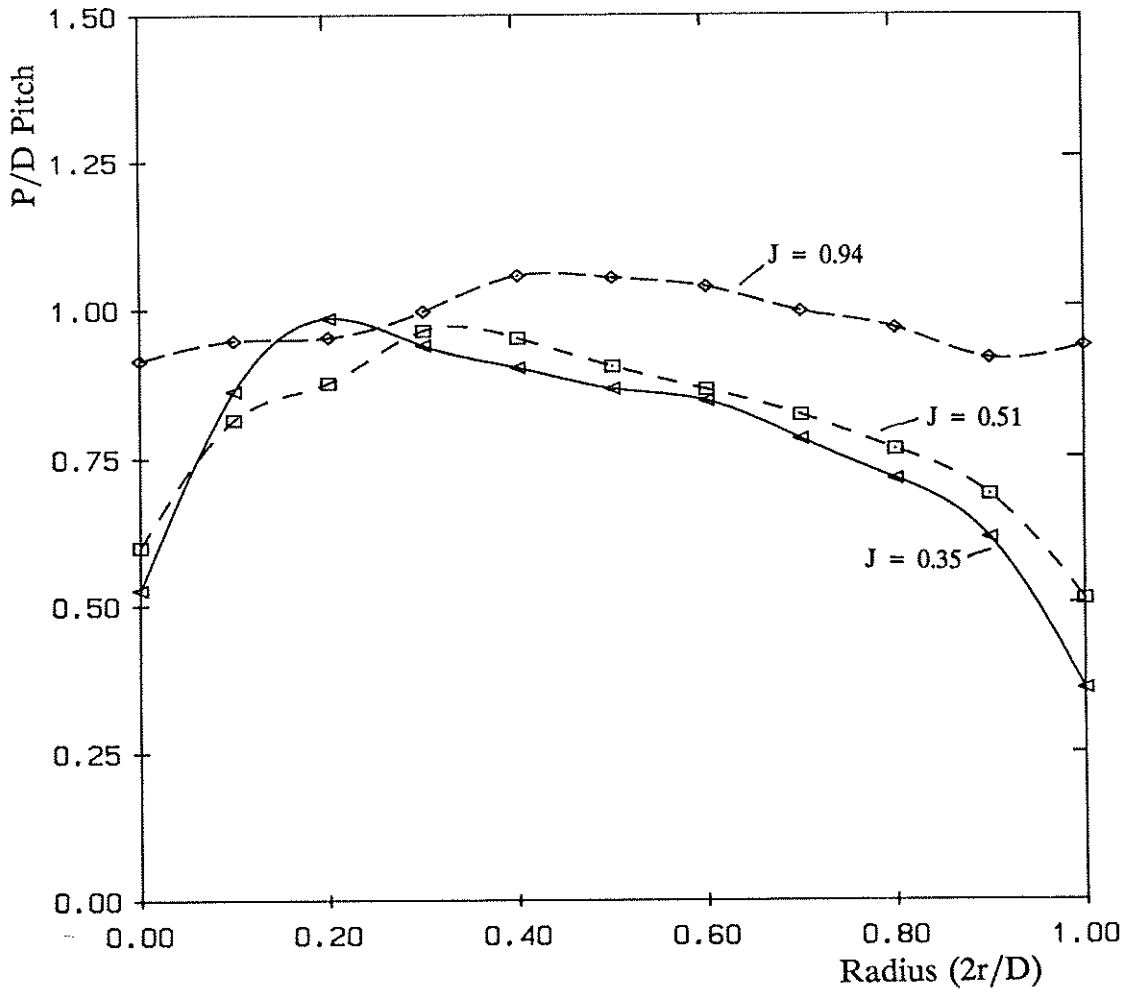


Figure 7.14 Radial variation of wake pitch for three advance ratio

It is accepted that the specification of the propeller wake as either a radially constant 'average pitch' or radially varying pitch is a simplistic representation. However, as shown in the next section, the downstream velocity field was deemed acceptable given the constraints on the number of body and wake panels. It is expected that using a larger transputer array to carry out a wake adaption a more realistic wake shape can be obtained within a reasonable time-scale. Also, if at a future date L.D.A. measurements are carried out such information could be used to further enhance the wake model.

In Figure 7.15 the radial circulation distribution for three advance ratios of 0.35, 0.51 and 0.95 are shown. Also plotted is the circulation distribution for the blade alone without the hub. It is interesting to see the effect of the hub holding up the value of circulation near the root and demonstrates the need for modelling of the hub geometry.

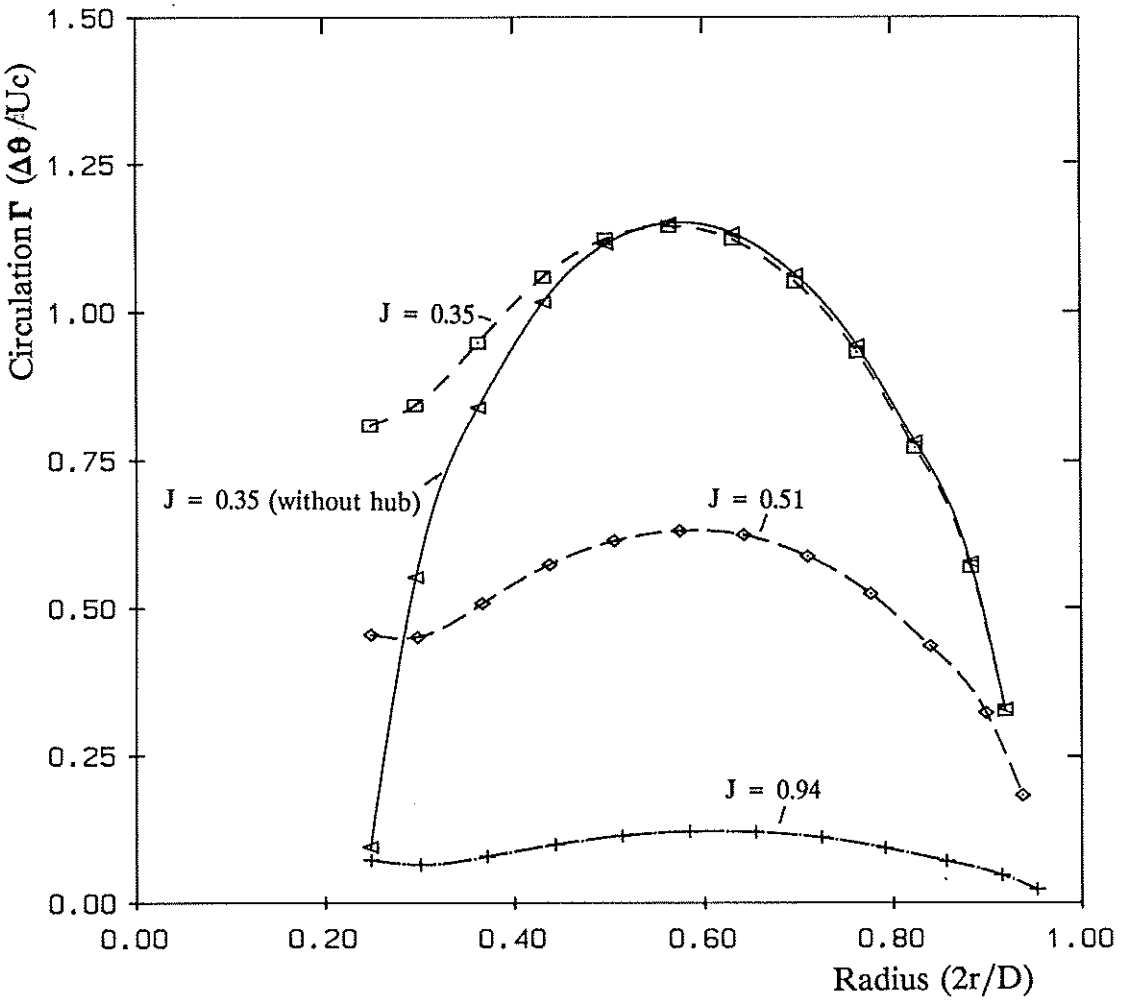


Figure 7.15 Radial distribution of circulation on propeller blade

Chordwise surface pressure distributions for 8 radii are plotted in Figure 7.16 for the four circulation distributions shown in Figure 7.15. The pressure coefficients are non-dimensionalised in terms of the local velocity not the inflow velocity. That is

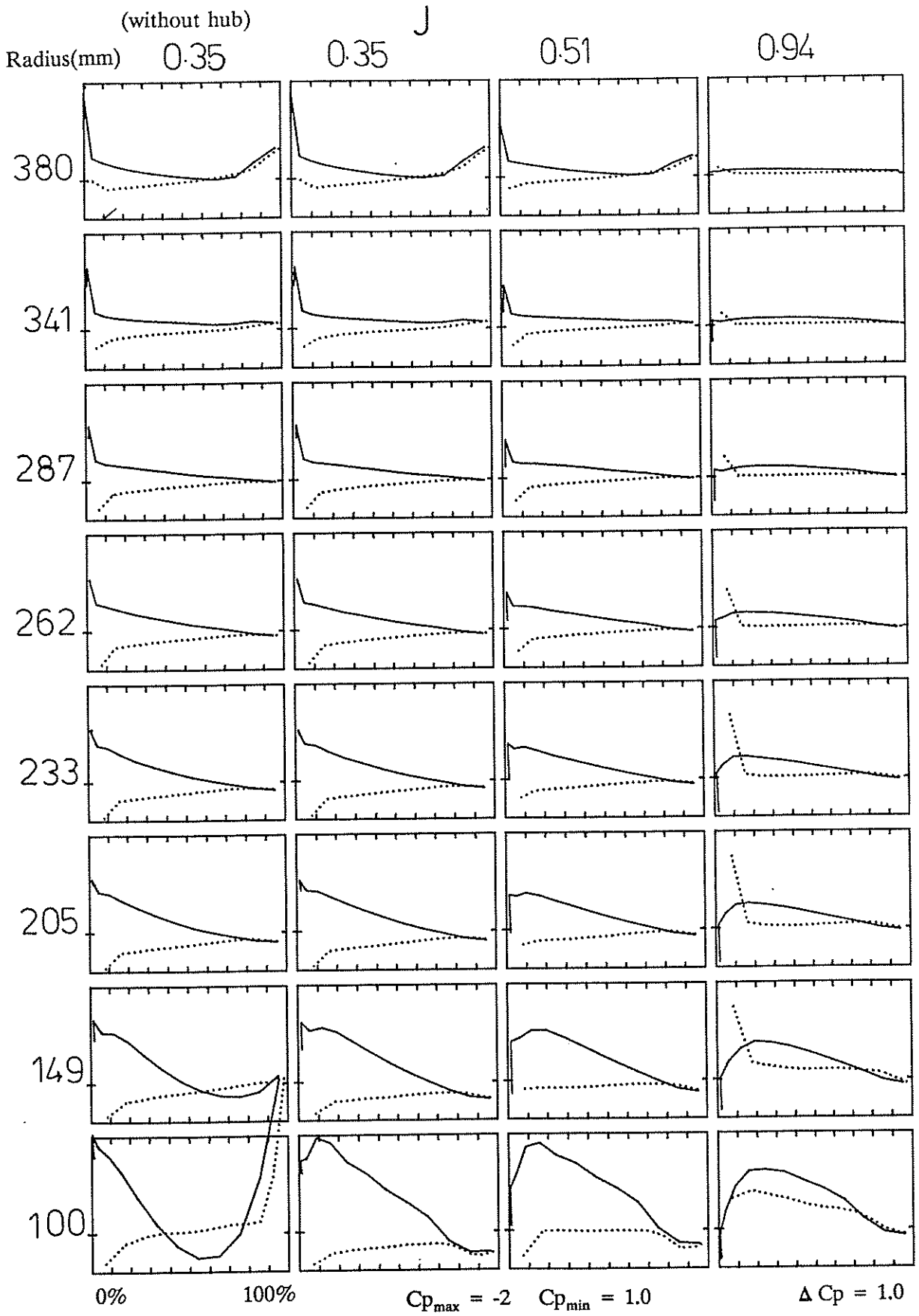


Figure 7.16 Chordwise pressure distributions for 8 radii on propeller blade

$$Cp = \frac{U_o \cdot U_o - U_t \cdot U_t}{U_o \cdot U_o} \qquad [7.2]$$

where U_t is the surface panel velocity, U_o the local velocity which is equal to:

$$U_o = U_\infty + r \times \omega \qquad [7.3]$$

It is interesting to compare in Figure 7.16 the pressure distributions at the root for J of 0.35 with and without the boss. It is thought that the heavy loading at the blade root may be indicating that the flow is actually stalled or separated in that region and explain some of the discrepancy in thrust between experiment and lifting-surface model. The effect of decreasing advance ratio is seen with the local loading for each section gradually increasing.

The corresponding hub strip pressure distributions for an advance ratio of 0.35 are shown in Figure 7.17. The values for the root strip of the blade are also plotted. A good match up between the blade root pressures is found with the adjacent hub strip. The discrepancy at the blade leading and trailing edge is probably due to the larger panel size on the boss. In the figure the difference in pressure downstream of the blade between adjacent strips 1 and 4 is due to the presence of the wake.

7.4.4 Velocity field downstream of propeller

In cylindrical coordinates, the fluid velocity induced by the propeller has three components. A component due to the axial acceleration, a radial component, and a circumferential or swirl component. The magnitude of each of these components is dependent on the distance from the propeller plane of revolution, radial position and the angular position relative to the blade generator. For the four-bladed propeller used the velocity field repeats itself every 90°.

The wake sheet downstream of the propeller determines the local structure of

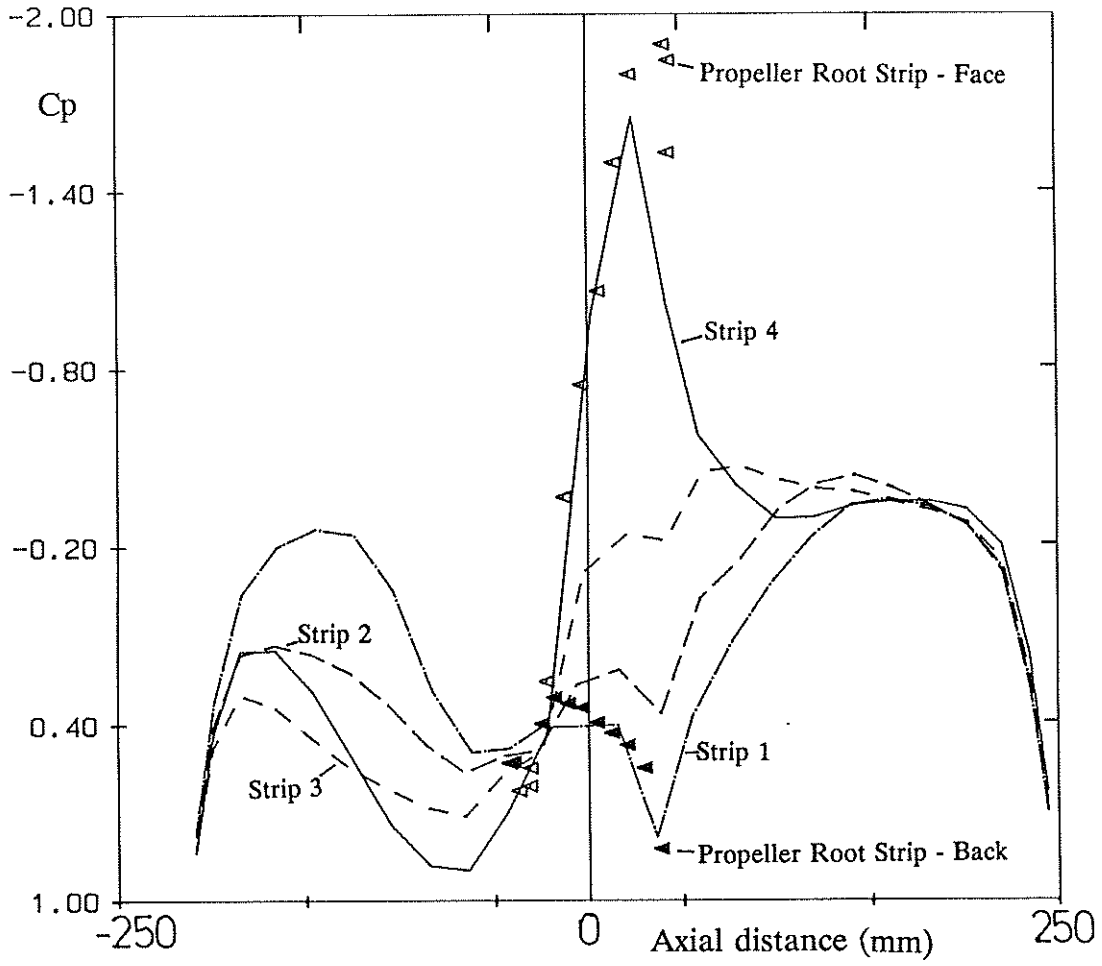


Figure 7.17 Pressure distribution on helical strips on propeller boss

the velocity field. The discontinuity of potential across the wake sheet gives rise to large differences in the magnitude of velocity tangential to the panel surface. Also, the presence of the wake edge can lead to large local velocities. The determination of the velocity field downstream of the propeller is carried out on a regularly spaced grid of points. These points are evenly spaced in an axial, radial, and circumferential directions. Points generated in such a manner may coincide with the edge of a wake panel and cause a high local velocity. In a circumferential averaging process these points should be removed. A filtering criterion that the swirl velocity was not greater than 5 times the free-stream velocity was found to adequately smooth the velocity field. The spatial averaging of velocity at a given radius generates the time-averaged velocity

for use as a rudder inflow velocity field.

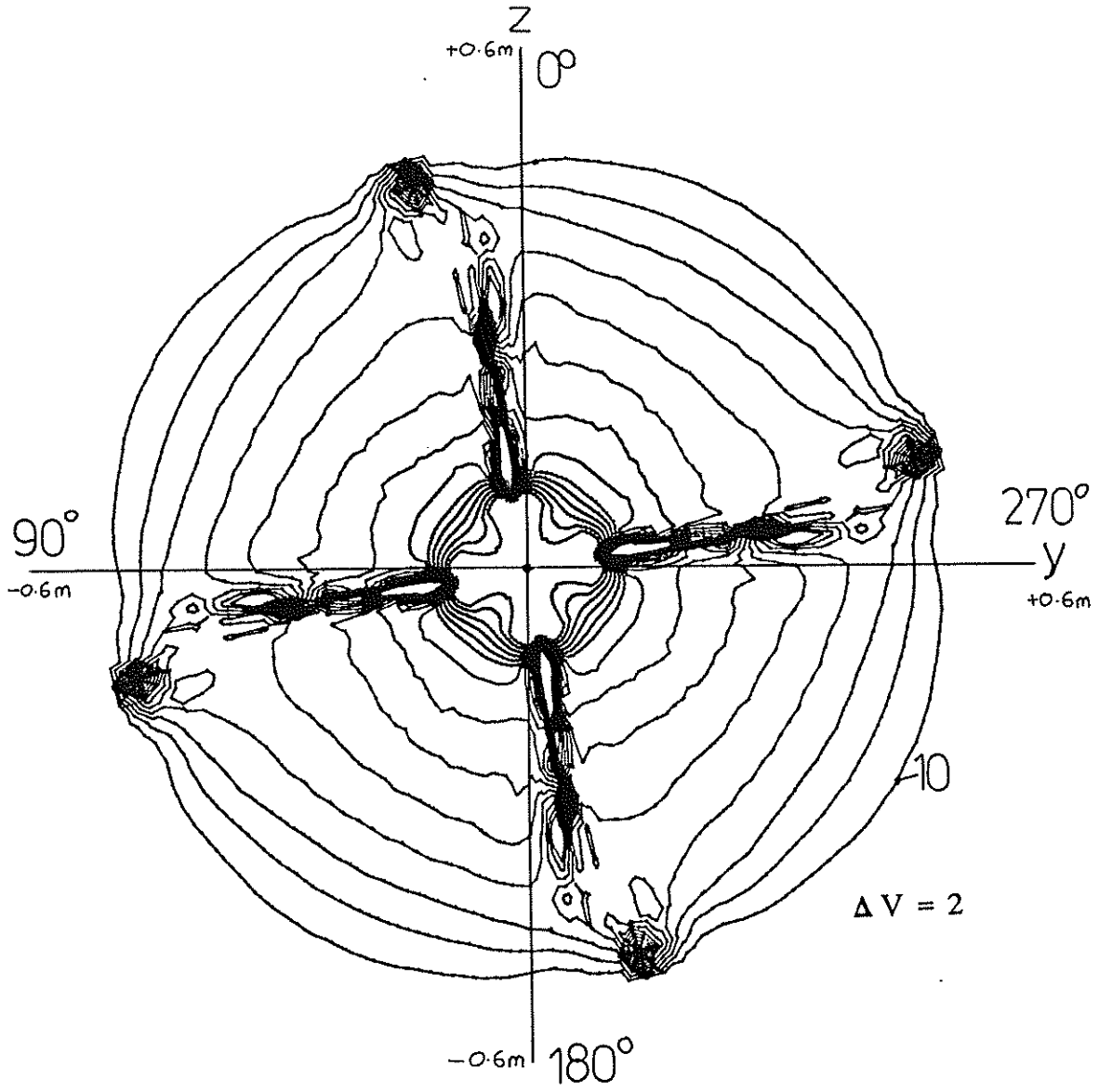


Figure 7.18 Contour plot of total velocity for vertical plane for J=0.35 at X/D=0.39 downstream of propeller

Figure 7.18 shows a contour plot of lines of total disturbance velocity for a complete section through the propeller race at an advance ratio J=0.35 at 0.39 of a diameter downstream of the propeller. The position of the four blade generators are at 0° , 90°, 180°, and 270°. These values were obtained using a constant wake pitch (610mm) of constant diameter. The location of the wake sheet is seen as local regions

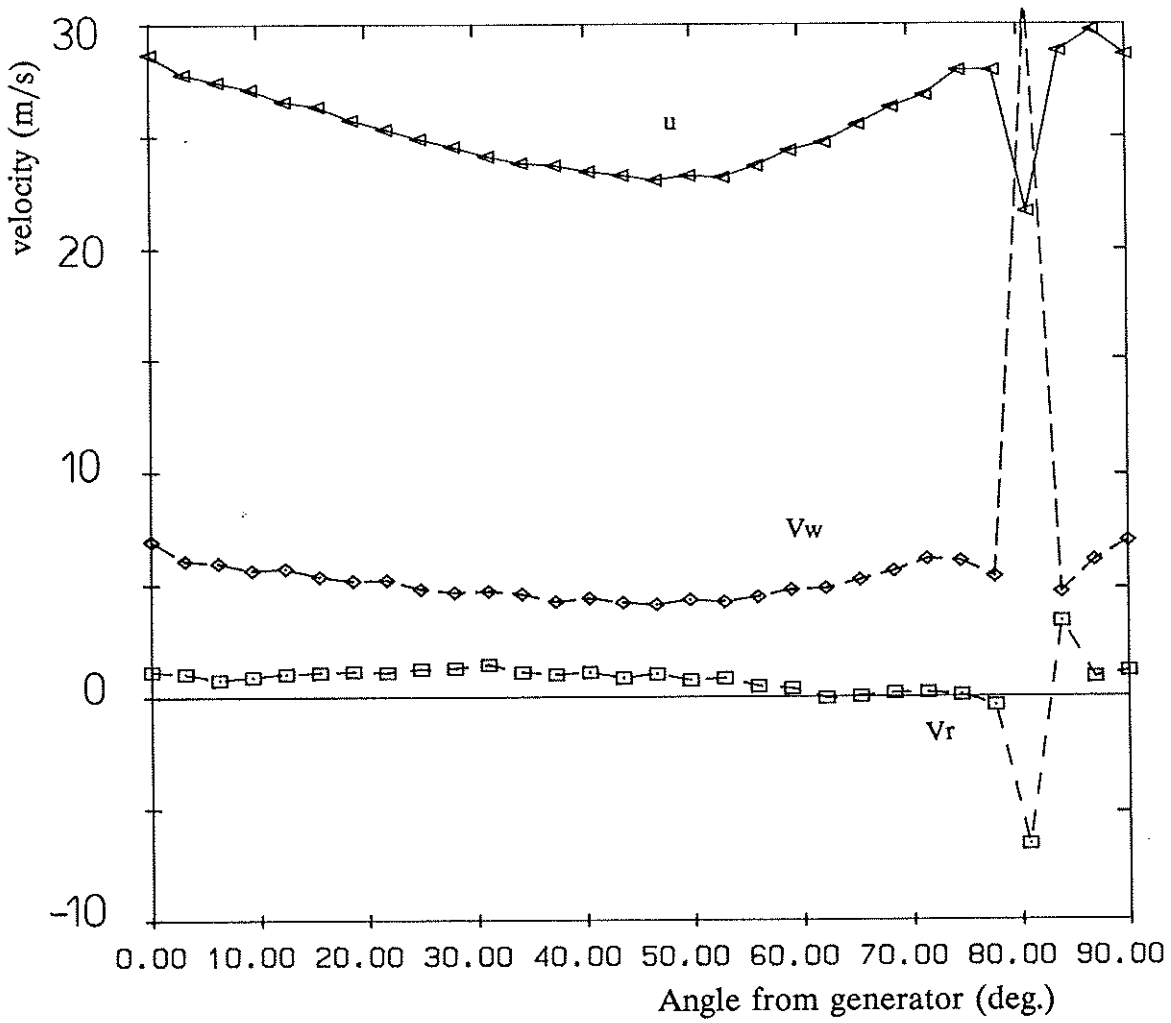


Figure 7.19 Circumferential variation of axial, radial and swirl velocity at X/D =0.39, r/D =0.35 and J=0.35

of rapid change of velocity. This is especially marked at the inner and outer radial extent of the propeller blade (r=100mm and r=400mm).

In Figure 7.19, for the same conditions at a radius of 0.35D, the angular variation for the axial, radial and circumferential components of velocity are shown. The effect of the wake sheet discontinuity can be clearly seen. The justification for the filtering criterion is the rapid change in velocity through the wake sheet whereas for the rest of the cycle the changes are gradual.

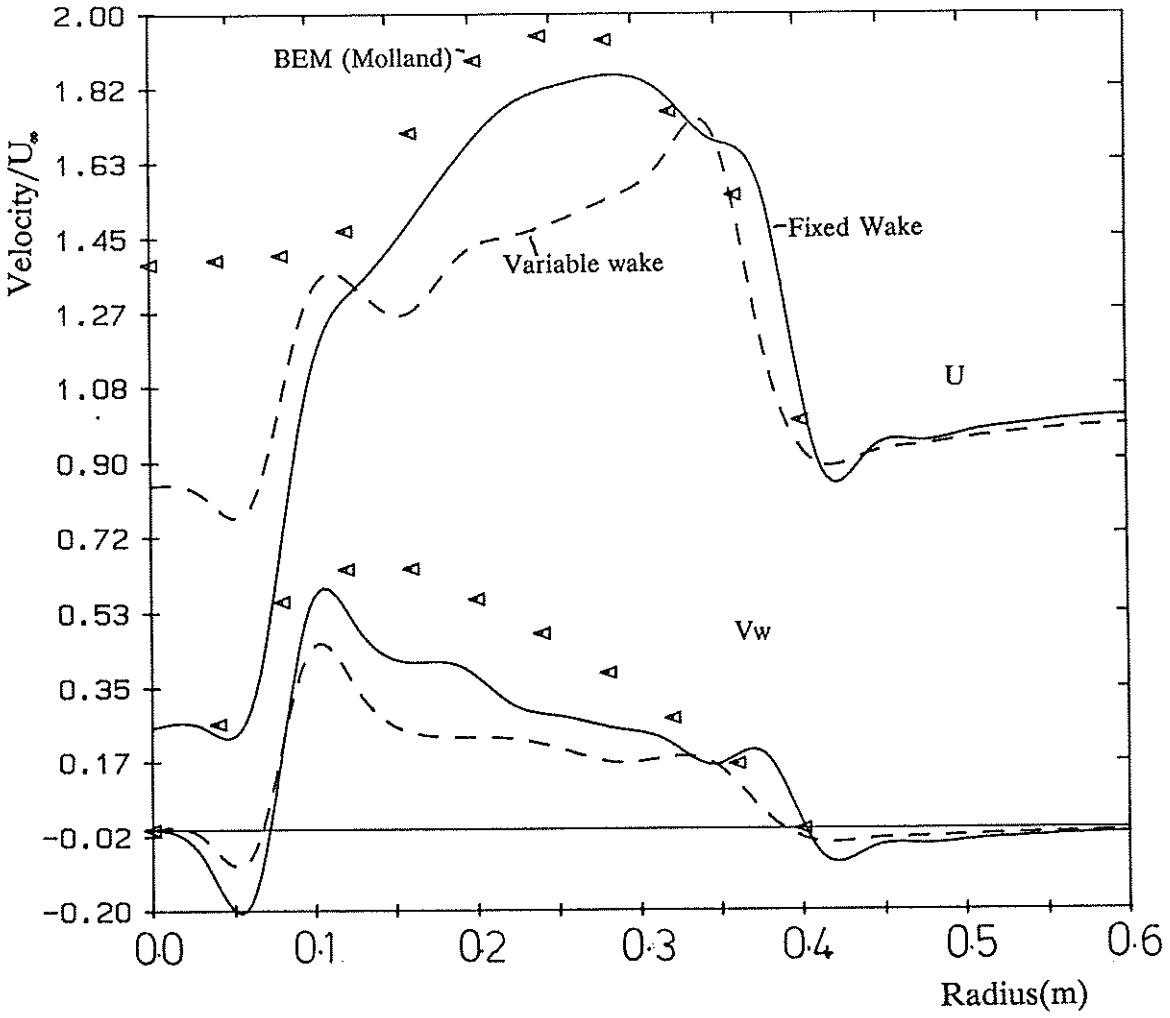


Figure 7.20 Radial variation of axial and swirl velocity at J=0.35 and X/D=0.39

The radial variation of the axial and circumferential (swirl) components of velocity at X/D=0.39 is given in Figure 7.20 for both the fixed pitch wake (610mm) and the radially varying pitch geometry. Also, shown are the Blade Element Momentum results from Molland[7.5]. The fixed pitch wake gives a better comparison with the B.E.M. result. This is thought to be due to the distorted radially varying wake shape being ill-defined by the relatively low number of panels. Also, it should be noted that the B.E.M. results do not allow for the presence of the hub and that in this region the velocities are simply estimated.

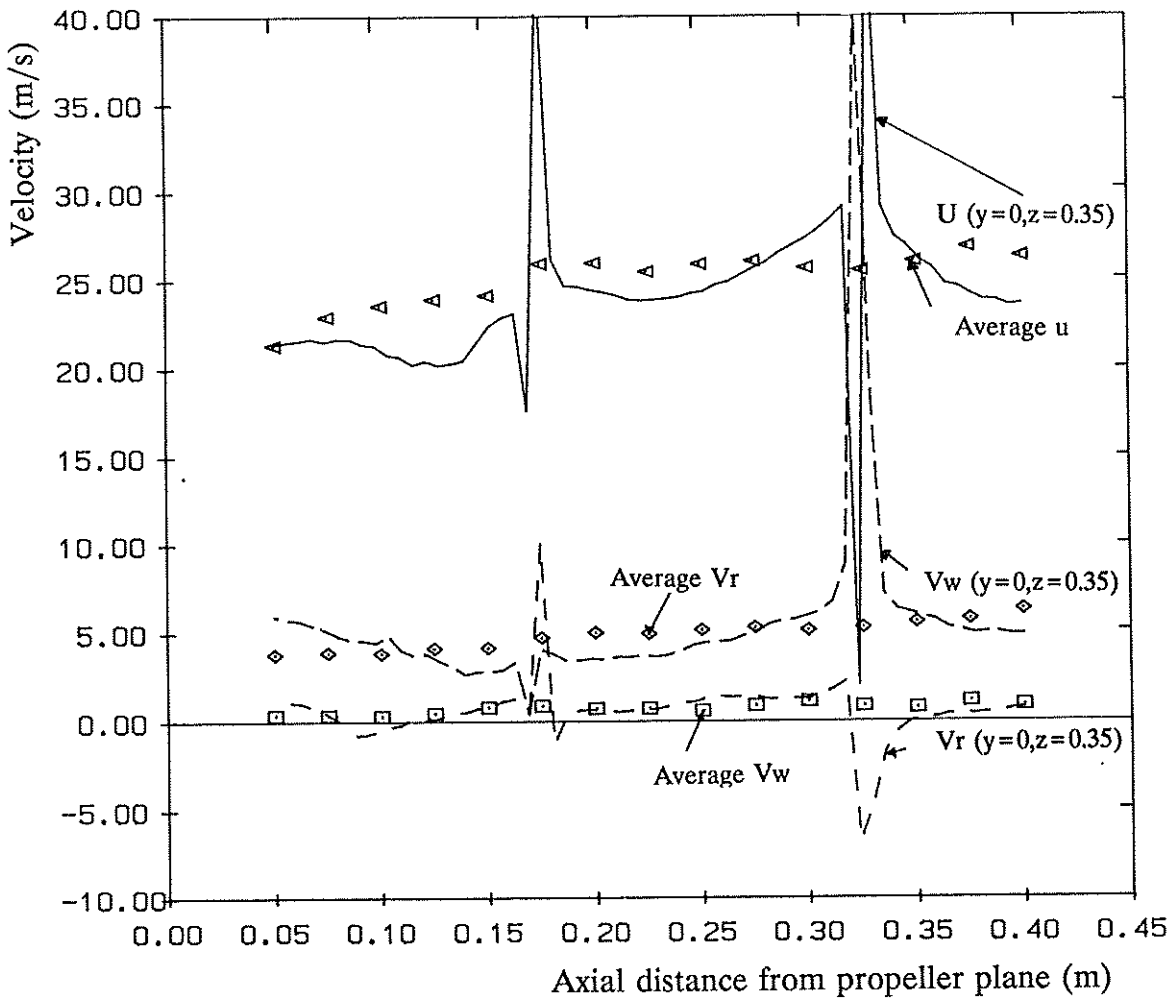


Figure 7.21 Longitudinal variation of axial, radial and swirl velocity at J=0.35

The longitudinal variation in the circumferentially averaged velocity components for a radius of 0.35D is shown in Figure 7.21, as is the variation in velocity at the position y=0, z = 0.35D. The position where the wake sheet cuts the plane y=0 can be clearly seen. Overall, as expected from momentum theory the average velocity shows a slow increase with distance. The rapid acceleration of the flow occurs over the first 150mm from the propeller plane.

7.4.5 Summary

The 400 panel blade and boss model of the modified Wageningen B4.40 propeller with a fixed pitch wake geometry based on Blade-Element Momentum theory gives a reasonable representation of propeller characteristics for the purposes of considering rudder-propeller interaction. The over-prediction by 10% is of the same order as that found for the rudder model. It is considered that the propeller blade loading and the resultant downstream velocity field is representative of that due to the actual blade loading.

The crude representation of the wake is acceptable at present, although future work should allow a more sophisticated representation to be developed. It is hoped that time accurate LDA measurements will aid in the development. In the absence of such information for validation it was decided that as simple as possible a model would at least allow a consistent approach to be used for all advance ratios.

The velocity field downstream of the propeller, for the purposes of modelling the interaction, can be found by inspecting 30 points per radius. A filtering process was used to remove spurious wake panel edge velocities. Overall, the fixed pitch wake appeared to give a better representation of the velocity field than the variable pitch model.

7.5 Interaction of Rudder and Propeller

7.5.1 Introduction

The method chosen to model this interaction is to separate the rudder and propeller and account for the interaction between them through the use of a modified inflow. The previous two sections have covered the application of a surface panel method to the analysis of a free-stream rudder and free-stream propeller. This section details how the interaction velocity field is applied to the separate models to account

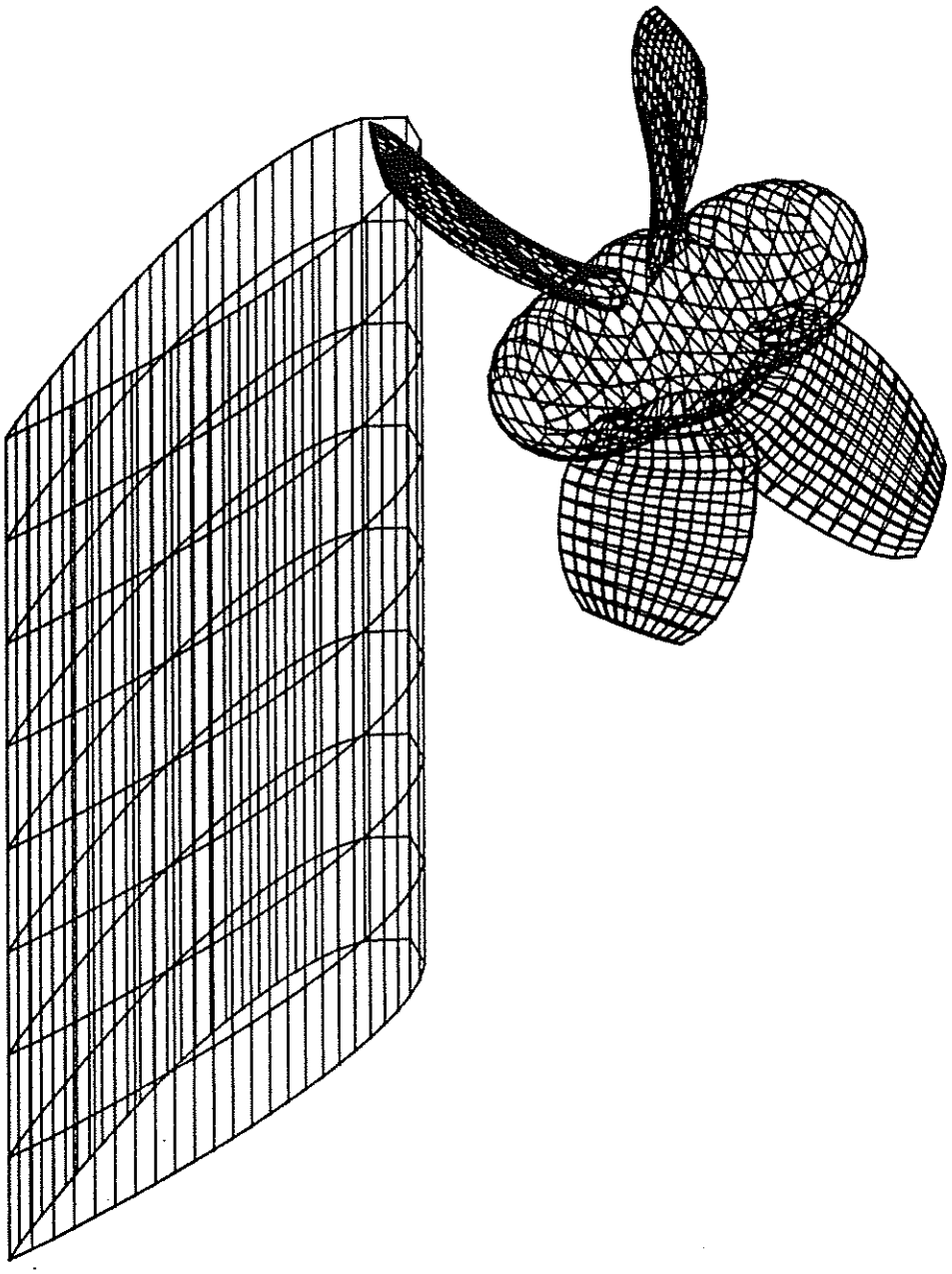


Figure 7.22 Isometric wire-frame view of the relative position of rudder and propeller for validation exercise

for their mutual influence.

To verify the interaction velocity field method a particular geometry is considered in detail. This case is that of the mid-longitudinal separation X/D=0.39 at the three experimental test advance ratios of 0.35, 0.51, and 0.94.

7.5.2 Geometry

The relative position of rudder and propeller model is shown in Figure 7.22. It should be noted that this is not how the flow is modelled. The rudder and propeller are separately modelled with their mutual influence accounted for by modification to their respective inflows. The numerical models used are identical to those described in the previous two sections. The propeller axis is 600mm above the tunnel floor. There is no lateral separation and the longitudinal separation of propeller plane and rudder stock is 512mm. The maximum height of the propeller blade is coincident with the tip of the rudder.

7.5.3 Interaction Velocity Field Algorithm

The development of the PALISUPAN software, as a series of procedures corresponding to the various stages in solving the perturbation potential method, allowed a specific program to solve rudder-propeller interaction to be written. Figure 7.23 is a flow chart of the overall process. The format of the geometry input file allowed the information defining both the rudder and propeller to be held in store on the host transputer without the need to re-load from file for each iteration cycle.

First the propeller geometry is loaded across the network and the panels generated. The propeller is solved first due to the stronger influence of the propeller on the rudder. The circumferential averaged velocity field at the rudder stock position is found and the rudder inflow velocity field modified accordingly. The rudder geometry is then loaded across the network, panels generated, and the flow about it solved. A circumferentially averaged propeller inflow velocity field at the propeller plane is found and the whole cycle repeated until convergence is deemed to have occurred. Relevant information is filed after each flow solution and velocity field generation. The program structure allows the flow to be solved for a series of rudder incidence.

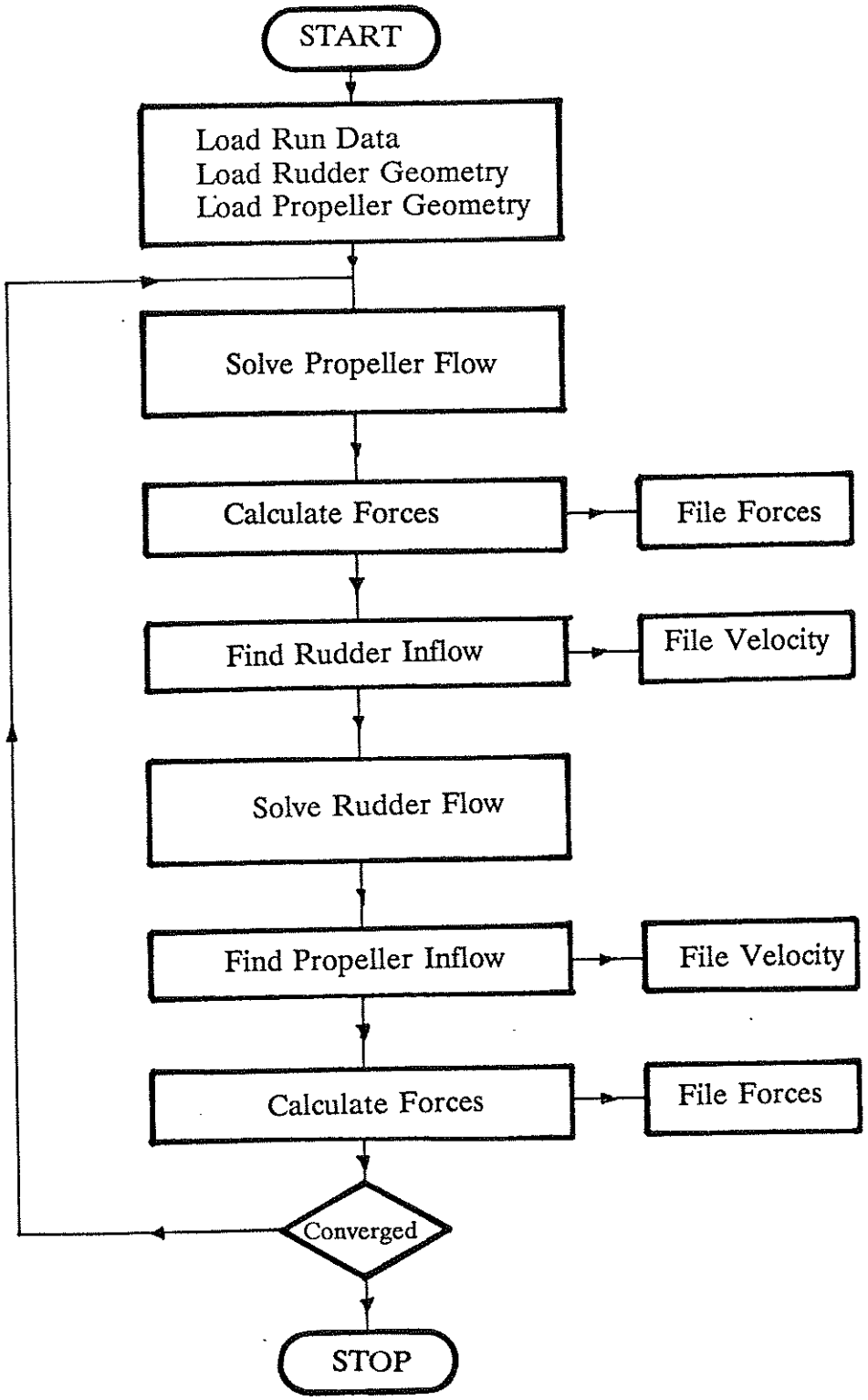


Figure 7.23 Flow chart of Interaction Velocity Field Algorithm

7.5.4 Rudder Interaction Velocity Field

The interaction velocity field downstream of the propeller at the rudder position is generated as described in the previous section. A circumferential average value for a given radial position is found with the filtering out of large velocity discrepancies. If an axially varying inflow field is to be used, this process is repeated at a regular axial spacing to give a series of time averaged velocities. At each axial station the rudder inflow velocity is the vector sum of the free-stream velocity and that due to the propeller interaction. The three components of velocity (axial U, radial Vr, and swirl Vw) are specified at a regular series of radially spaced points relative to the origin of the propeller axis (xo,yo,zo). The inflow velocity field store uses this information to specify the cartesian components of velocity for a given spatial location (x,y,z) in the following manner:

$$U(x, y, z) = U_o + U(x)\, a + Vr(x)\, r + Vw(x)\, w$$

$$\text{where} \quad a = (1, 0, 0)$$

$$r = (0, (y - yo), (z - zo)) \qquad [7.4]$$

$$\text{and } w = a \times r$$

The data defining the initial velocity field is included in the geometry input file. At subsequent iterations the velocity field calculation process modifies the values stored on the Host prior to the generate panel phase. An inflow velocity defined at 20 radial stations with averaging over 30 points per radius was found to be acceptable.

7.5.5 Propeller Inflow Velocity Field

A spatially averaged velocity field at points in a cylindrical tube upstream of the rudder is obtained. This allows the average reduction in axial velocity passing through the propeller due to the rudder to be found. Also, spatially averaged swirl, and radial components can be found. The velocity field including the inflow velocity is generated

on a cylindrical disk for inclusion in the geometry input file in an identical manner to that for the rudder. The velocity field at 10 radial points averaged over 20 points at 18° increments was found to adequately define the propeller inflow.

7.5.6 Iteration and Convergence

The downstream position of the rudder means the propeller has a far greater influence on the rudder than the rudder has on the propeller. Therefore, the first stage in determining the interaction is to solve the flow around a free-stream propeller, then generate the propeller's downstream velocity field and apply to the rudder. On solution of the rudder flow the rudder upstream velocity field is obtained. This field is then applied to the propeller and the process is repeated until convergence has occurred.

A number of convergence criteria are possible. The application of the work is for manoeuvring characteristics in which of prime interest is the rudder sideforce. Therefore, the convergence of total rudder lift was chosen.

For the representative geometry at a rudder incidence of -0.4° and J=0.35 the convergence behaviour of both the propeller and rudder forces were investigated. Figure 7.24 plots the percentage variation in rudder side-force C_L with each complete iteration cycle (for the first cycle the difference is between the free-stream value and the interaction result). The realtive change in C_L is plotted using a $\log_{10}$ scale. It can be seen that convergence is rapid. Similar behaviour was found at the other advance ratio and for different rudder incidence. The other components of rudder and propeller forces exhibited a similar rapid convergence. For this work a 0.5% variation in rudder sideforce was deemed to be an acceptable convergence criterion. In general, 2 or 3 iteration cycles were required for convergence.

For the parametric studies of Chapter 8 the final rudder inflow velocity field at zero incidence was used for all rudder incidence as the variation in sideforce due to the interaction was unlikely to alter the trends of the results as the performance was

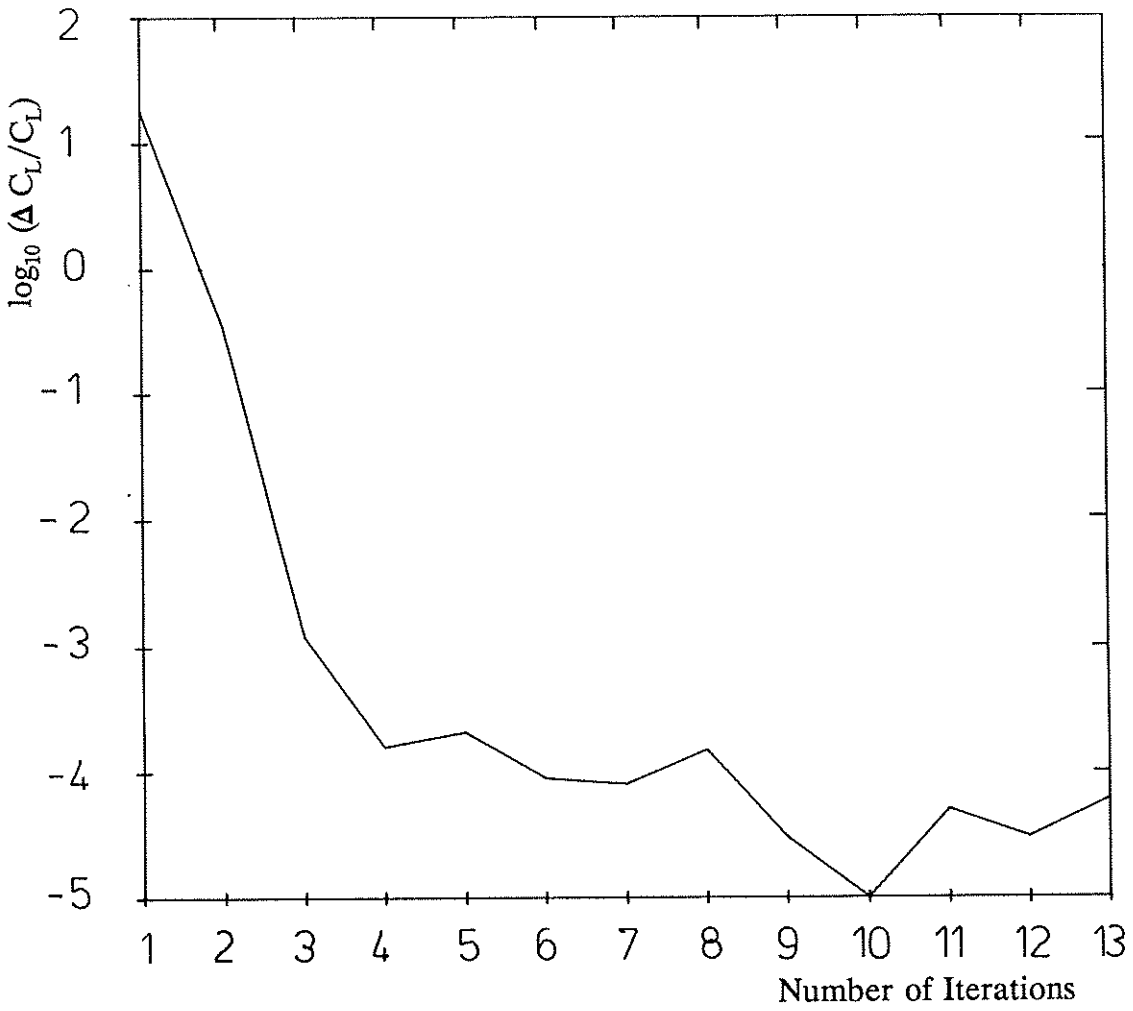


Figure 7.24 Convergence of rudder lift with number of iteration cycles

modelled for incidence close to zero (< 10°) where the velocity field does not change significantly.

7.5.7 Comparison of model and experiment

The numerical model results presented are for a series of rudder incidence corresponding to the values used in the experimental tests of -30.4°, -20.4°, -10.4°, -0.4°, 9.6°, 19.6°, and 29.6°. The experimental test advance ratios were used with a detailed comparison for an advance ratio of 0.51.

A series of sensitivity tests were carried out at J = 0.51 for a rudder incidence of +9.6°. The results for this condition were representative of the behaviour at other incidence and advance ratio. The convergence behaviour of rudder C_L and C_D with panel density was similar to that for the free-stream rudder. To investigate the satisfactory capture of the flow physics, figures for the spanwise variation of sectional lift coefficient will be shown.

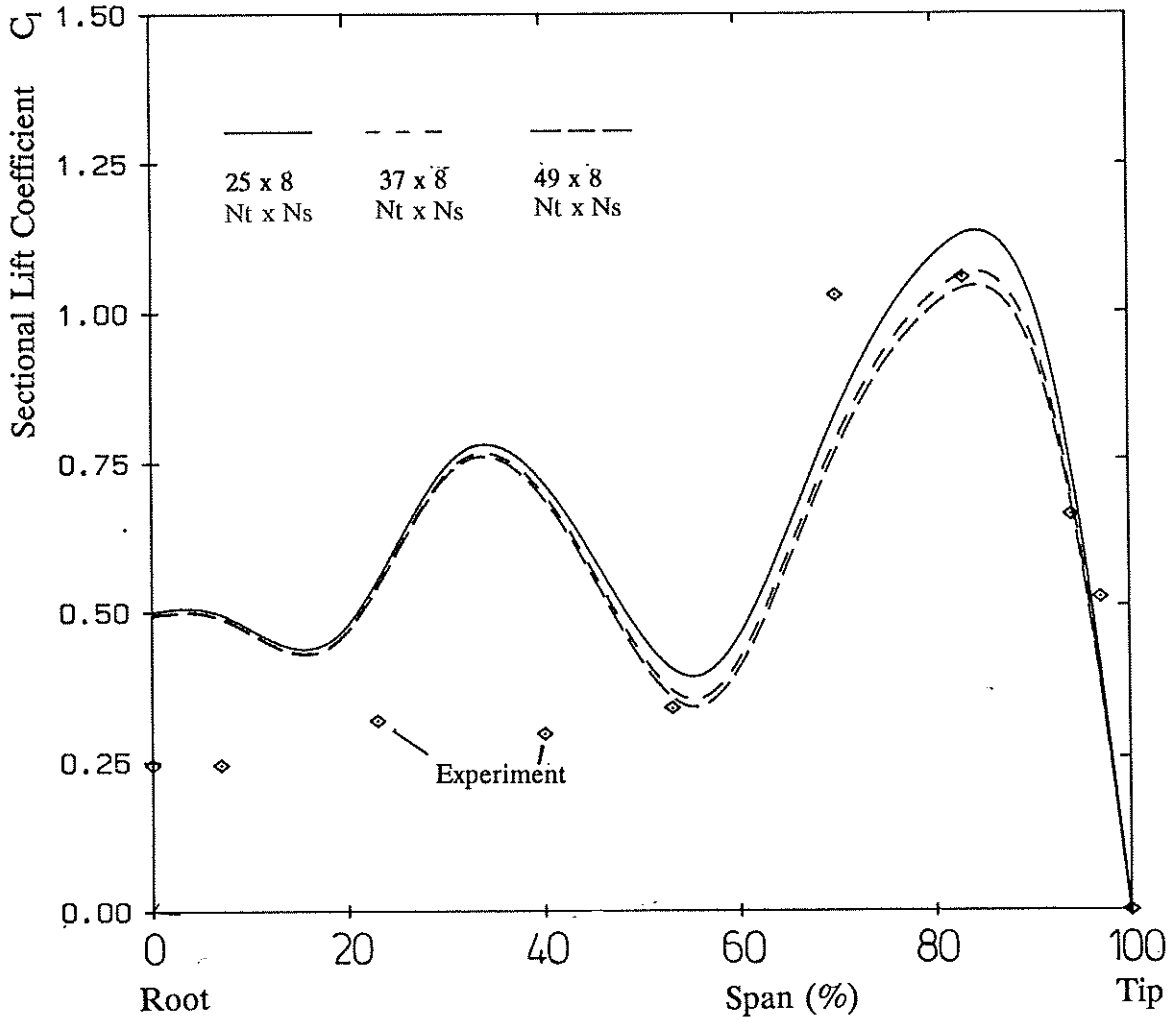


Figure 7.25 Variation of spanwise loading: Effect of the number of chordwise panels for Rudder No. 2 at J=0.51 and +9.6°

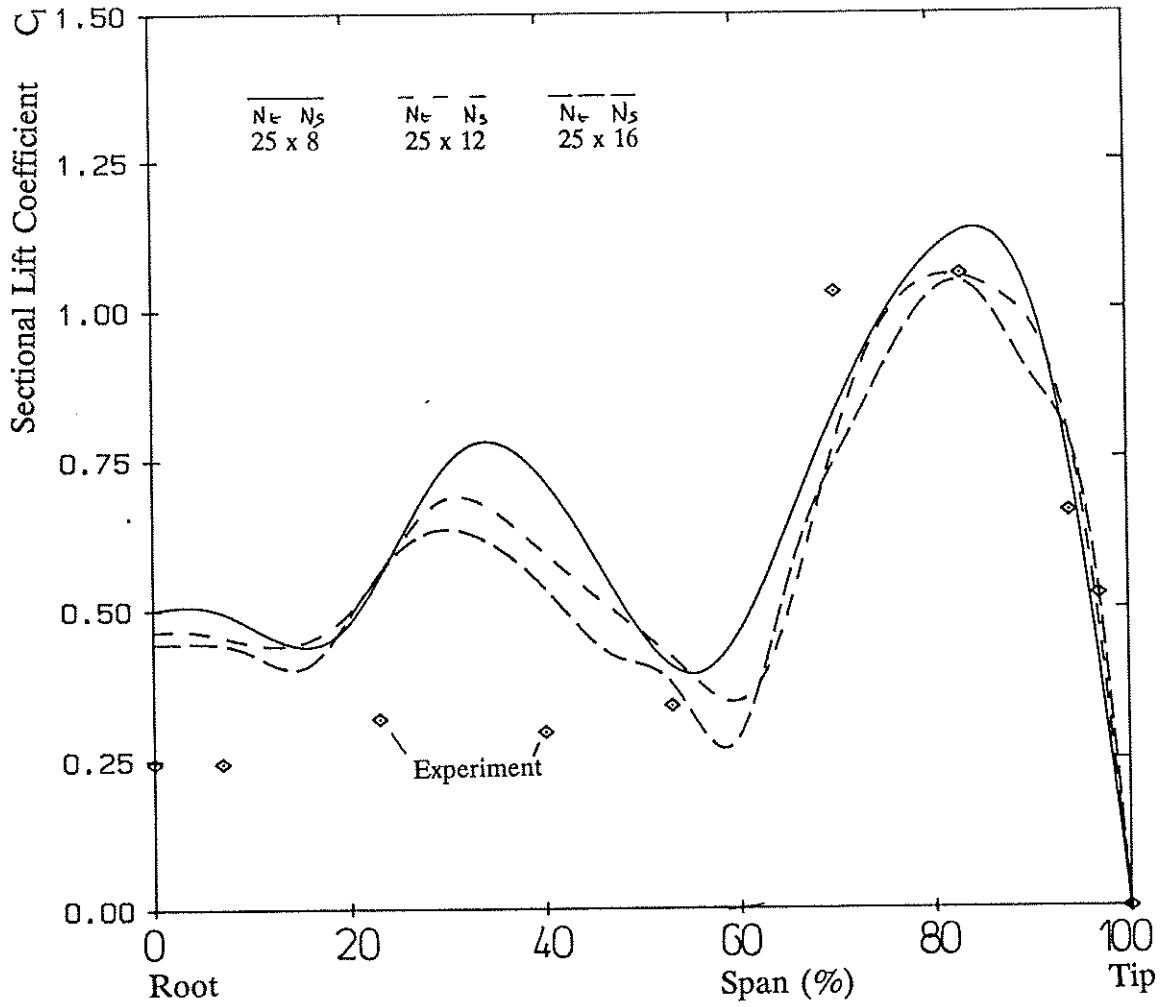


Figure 7.26 Variation of spanwise loading: Effect of the number of spanwise panels for Rudder No. 2 at J=0.51 and +9.6°

The variation of the spanwise loading with increased number of chordwise and spanwise panels is shown in Figures 7.25 and 7.26 respectively. In a manner similar to the propeller blade the shape is more sensitive to the number of spanwise panels. Therefore, a panel distribution with Nt = 25 and Ns =16 was used for all subsequent cases. Regular panel spacing was used in the spanwise direction and sinusoidal clustering at the leading and trailing edges in the chordwise direction.

The effect of the more regular velocity field generated by a fixed pitch wake is clearly seen in Figure 7.27, the radially varying wake velocity field results in a smearing

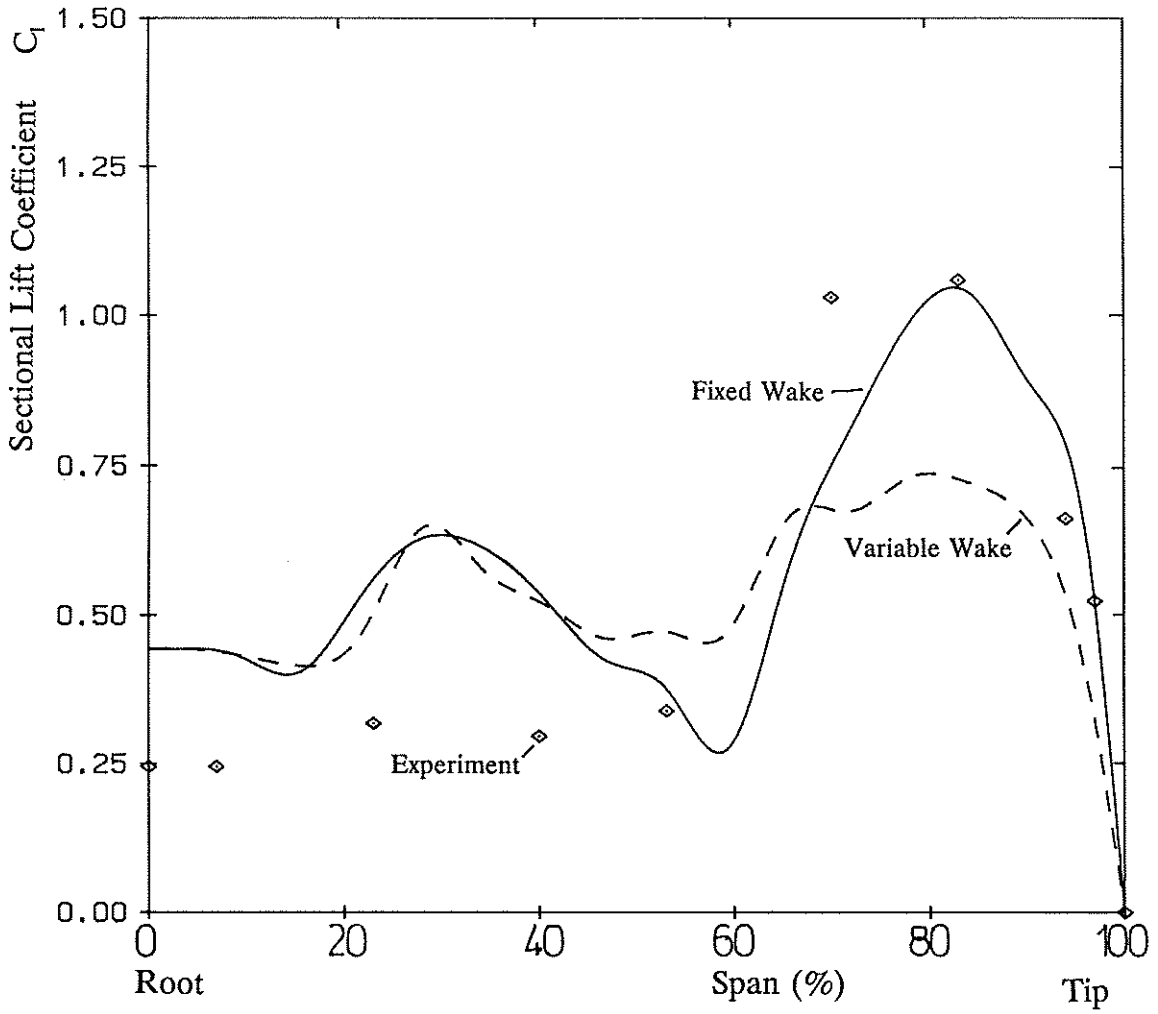


Figure 7.27 Variation of spanwise loading: Effect of the propeller wake model for Rudder No. 2 at J=0.51 and +9.6°

out of the spanwise load distribution. Therefore, the fixed pitch wake was used for subsequent tests.

The previous comparisons were for a rudder inflow velocity field generated at the position of the rudder leading edge. The effect of longitudinal variation in the axial position used to generate the velocity field is shown in Figure 7.28 for velocity fields generated at the rudder leading edge, rudder stock and trailing edge. It is likely that the velocity field at the stock is the most representative average velocity value; the value will be weighted toward the leading edge as this is the most sensitive region for

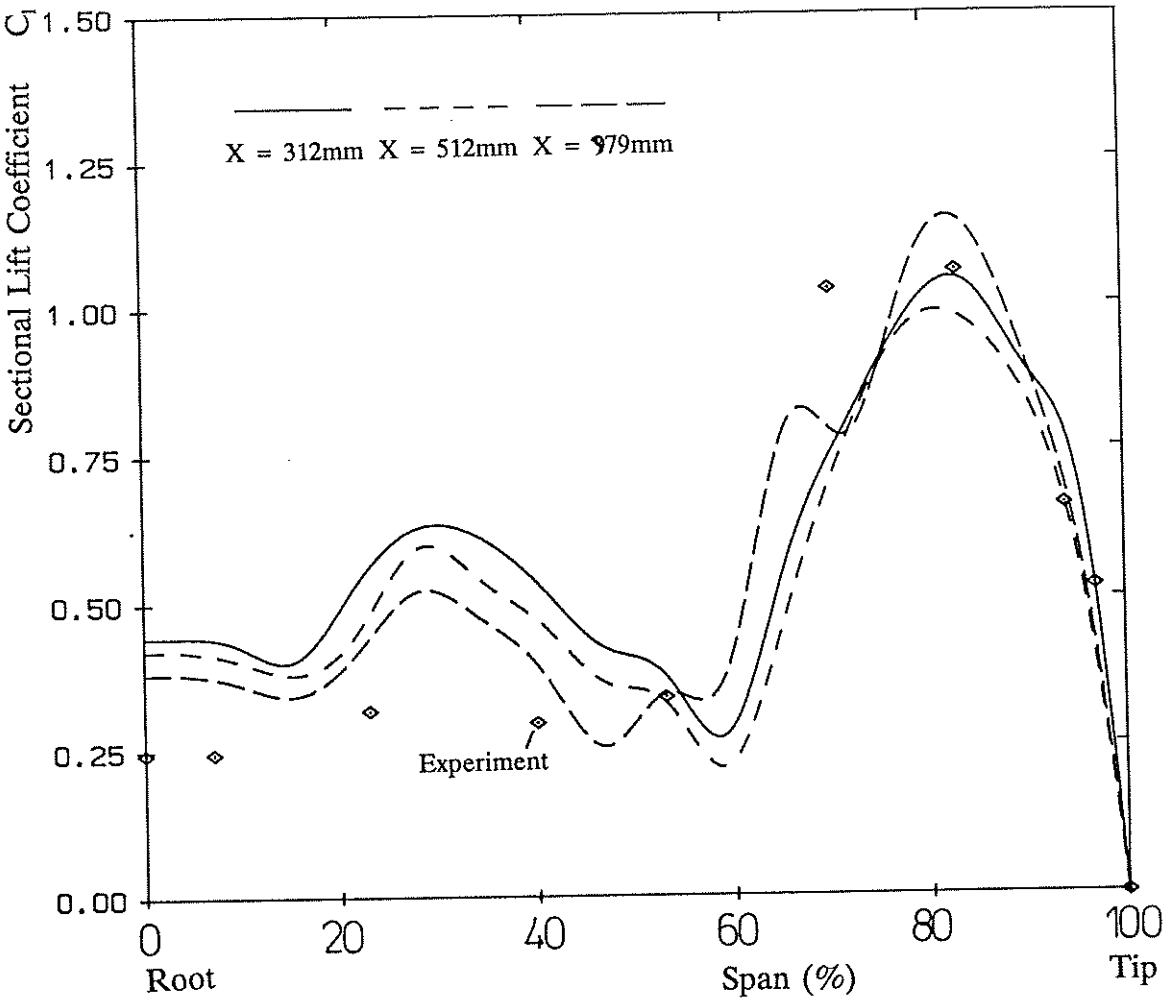


Figure 7.28 Variation of spanwise loading: Effect of the longitudinal position of velocity field for Rudder No. 2 at J=0.51 and +9.6°

rudder loading. Therefore, the velocity field used for subsequent tests is located at the rudder stock.

In the previous figures the numerical result under-predicts values at the tipwise hump (80% span) and over-predicts those at the inner hump (30% span). A similar result was found by Molland [7.5] using lifting-line theory. Both the inner and outer hump require extra experimental points to satisfactorily resolve the actual shape. However, it appears some of the flow physics is ignored. To attempt to resolve this the following were considered:

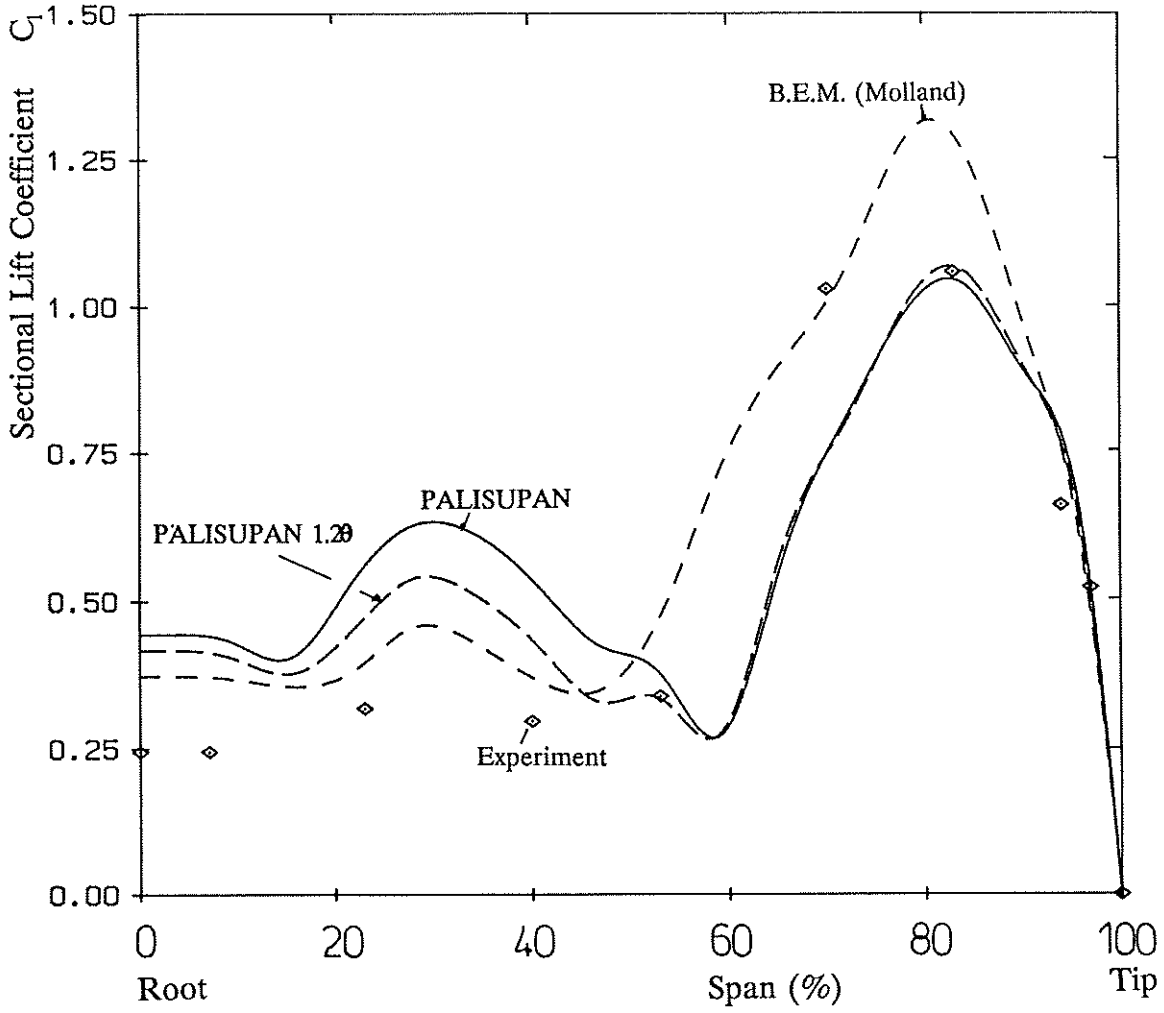


Figure 7.29 Variation of spanwise loading: Effect of modifying velocity field for Rudder No. 2 at J=0.51 and +9.6°

1) Figure 7.29 compares results for B.E.M. result of Molland[7.5] for the velocity field downstream of the propeller (see Fig. 7.20) and for a rudder inflow with a 20% increase in the inflow angle, defined thus θ:

$$V_T = \sqrt{U^2 + Vw^2} \quad where\ \theta = \tan^{-1}\left(\frac{Vw}{U}\right)$$

$$for\ \theta' = 1.2\,\theta \qquad [7.5]$$

$$U' = V_T \cos\theta' \quad and \quad Vw' = V_T \sin\theta'$$

with a constant total velocity at a given radius but with energy redistributed from the axial to the swirl direction. This attempts to see how sensitive the result was to the distribution of axial and swirl velocity within the race. The B.E.M. result over-predicts the 80% hump while reducing the 30% hump excess. The 20% increase in θ does not affect the 80% hump while it does reduce the 30% hump. It was decided to not modify through the use of the empirical data to obtain a better fit as the modification would only have limited application.

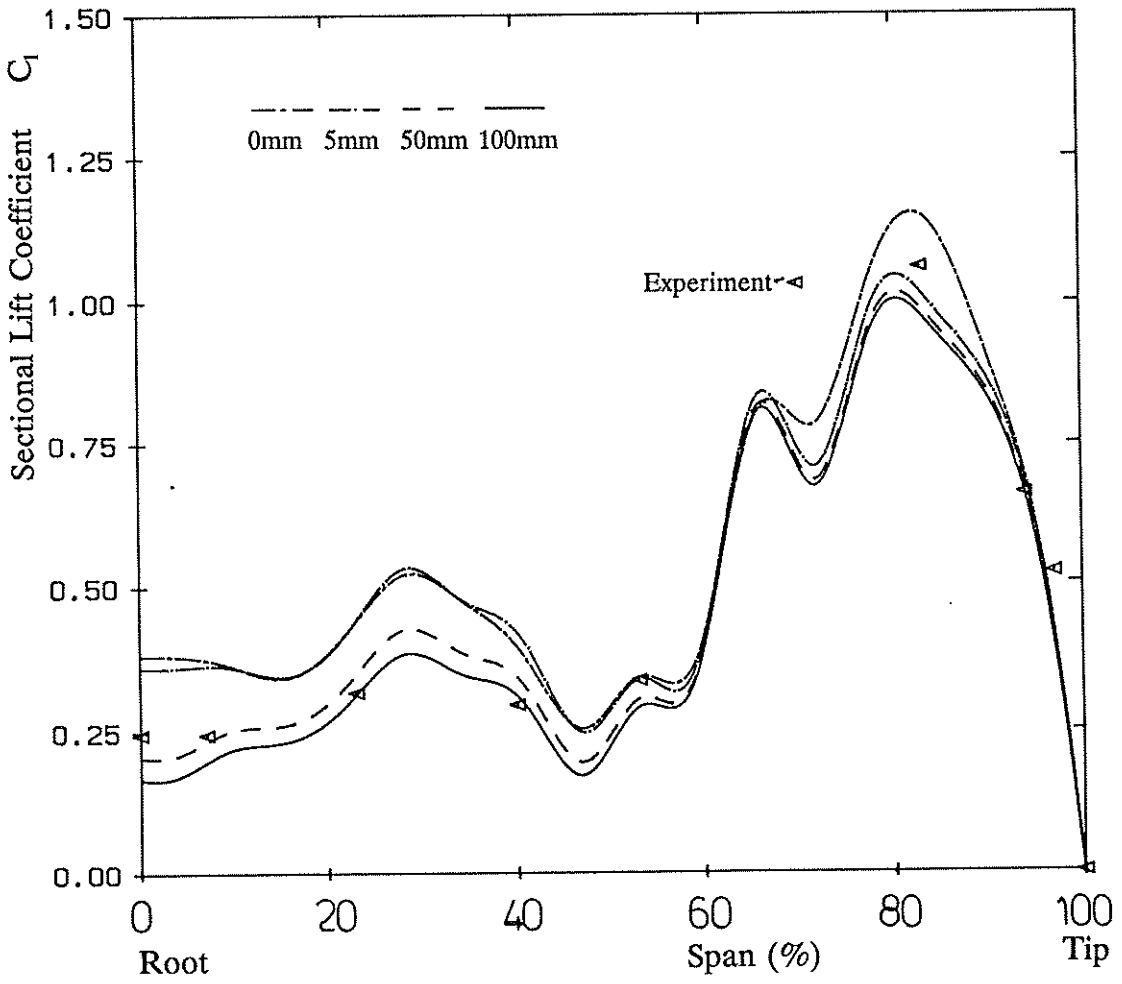


Figure 7.30 Variation of spanwise loading: Effect of rudder root gap for Rudder No. 2 at J=0.51 and +9.6°

2) In Figure 7.30 the effect of gap between the rudder root and wind tunnel floor was investigated. In the experimental tests this gap was 5mm although there was some flow through the hole through which the rudder stock passed to the dynamometer which could potentially lead to a greater gap effect. It can be seen that the non-physical 50mm and 100mm gaps are effective at reducing the 30% hump and match the root region lift. However, the 5mm gap, as expected, only has a minimal effect. As it is physically realistic it was retained in the model by setting the reflection plane at z= -5mm.

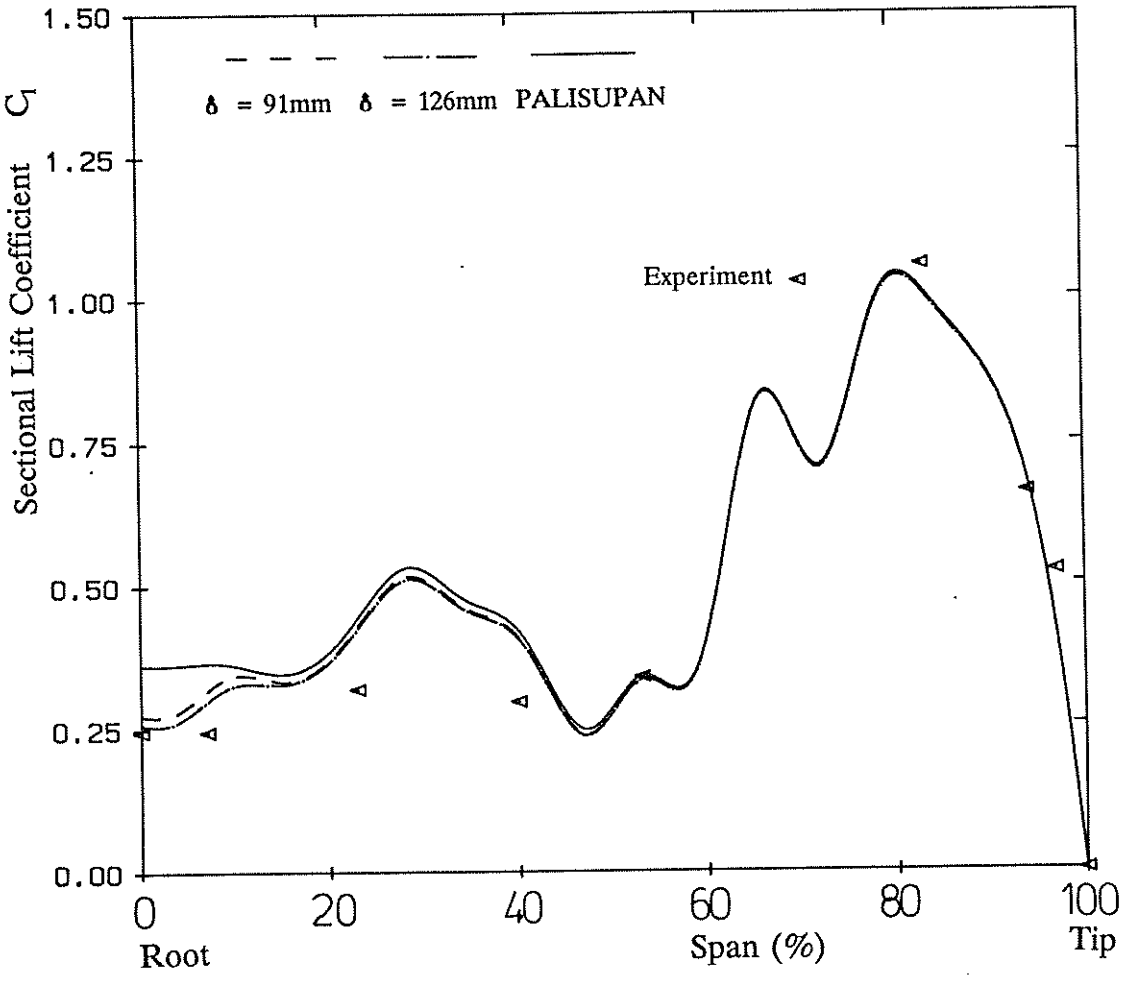


Figure 7.31 Variation of spanwise loading: Effect of tunnel boundary layer for Rudder No. 2 at J=0.51 and +9.6°

3) The effect of the turbulent boundary layer flow along the wind tunnel floor was investigated by reducing the axial velocity in the boundary layer. The standard relationship for turbulent boundary layer development with an assumed $1/7^{th}$ power velocity variation with height was used. Figure 7.31 shows the effect of two assumed boundary layer thicknesses of 91 and 126mm corresponding to the realistic boundary layer development length at the rudder of 5m and a value double that of 10m. These both appear to model the root region well. The 5m long turbulent boundary layer correction to the inflow velocity field was therefore retained . However, there is little effect on the load distribution in way of the propeller.

Remaining physical reasons for this disparity between experiment and theory are potentially the tip vortex holding up the outer hump although this effect is included in the lifting-line model and does not appear to affect the 30% hump. More probable contenders are the quasi-steady assumption, especially for the propeller. If each blade was modelled with the actual cross-flow induced by the rudder there might be an alteration of the propeller race direction away from the propeller axis. The direction of this alteration would depend on the sign of the rudder incidence and be symmetric about $\alpha = 0°$. A similar effect could occur due to the presence of the ground board inducing a cross-flow due to the highly rotational propeller wake and its image. Such an effect would be likely to depend mainly on the direction of rotation and thrust of the propeller and be non-symmetric about $\alpha = 0°$.

Although the predicted spanwise load distribution does not exactly follow that measured, the correspondence was considered sufficiently close that the Interaction Velocity Field method was considered acceptable for modelling rudder-propeller interaction for the purposes of predicting manoeuvring performance and for parametric design studies.

In Figure 7.32, the numerical and experimental results are compared for seven angles between -30.4° and +29.6°. The comparison between -10.4° and +9.6° is good. The shape gradually distorts away as the magnitude of incidence increases further. For

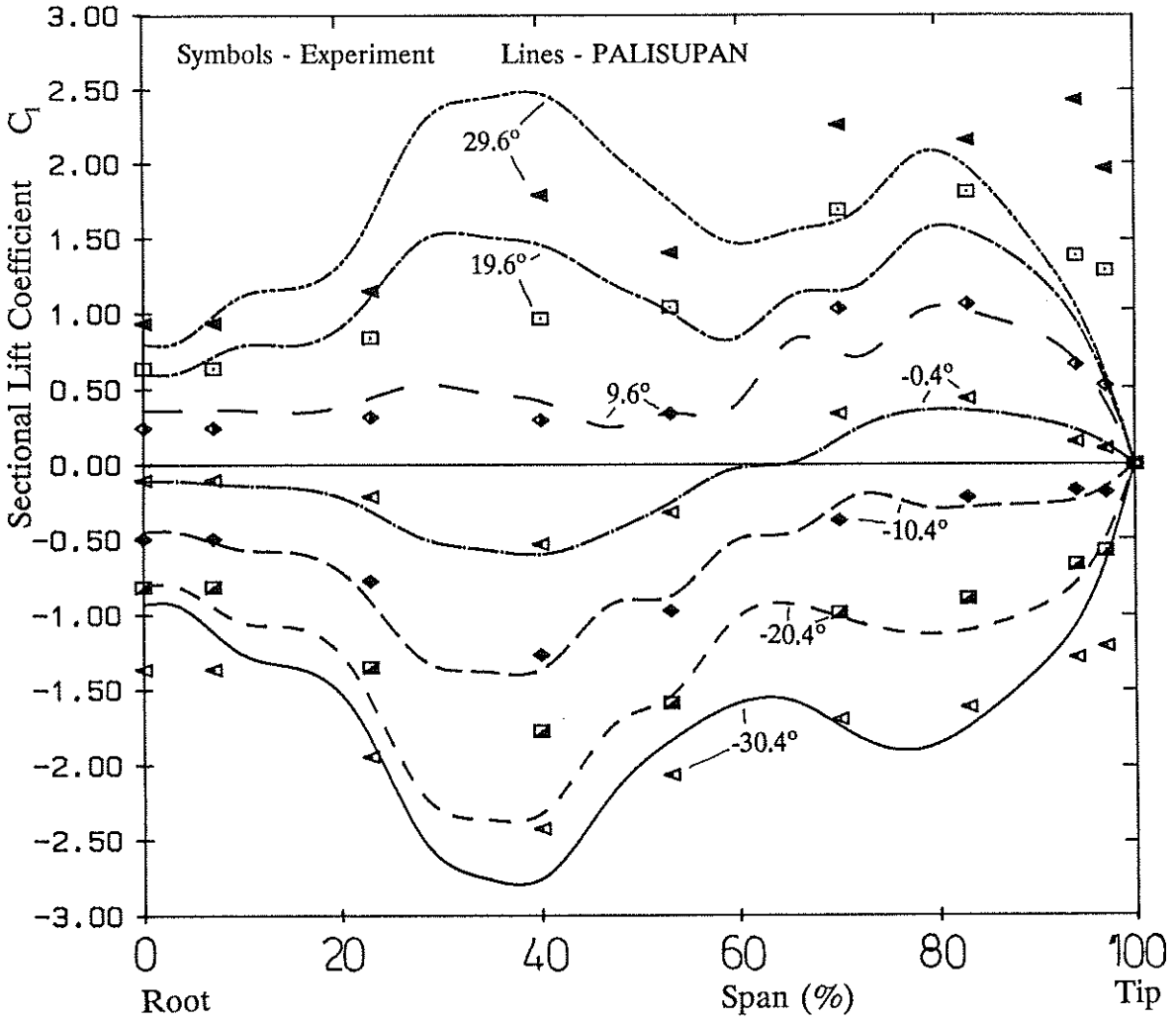


Figure 7.32 Comparison of lifting-surface and experimental spanwise load distribution for Rudder No. 2 at J=0.51

positive incidence the 30% hump is over-predicted and the 80% hump under-predicted. For negative incidence, the 30% hump is overpredicted and the 80% hump fairly well modelled although as with the free-stream comparison the clear tip vortex effect is not modelled. The symmetry of behaviour with incidence suggest the effect most likely to account for the difference in spanwise loading is likely to be due to the rudder induced cross-flow at the propeller.

In Figure 7.33 the actual and predicted quasi-steady pressures are compared for three advance ratios at α = -0.4°. The overall shape and values compare well apart

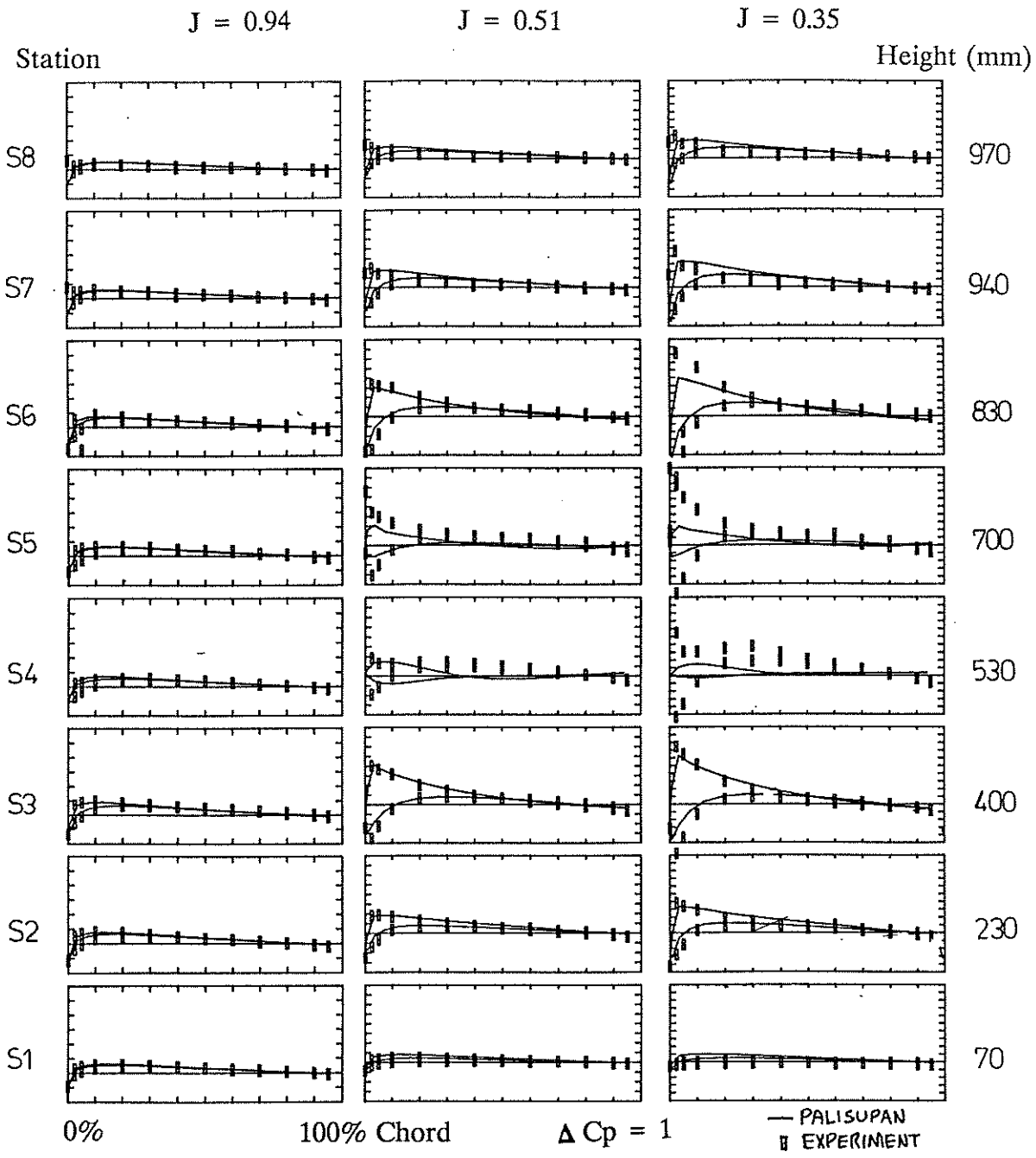


Figure 7.33 Comparison of lifting surface and experimental pressure distributions for Rudder No. 2 at J= 0.94, 0.51 and 0.35

from the 53% and 70% span positions. The experimental shape at 53% is unusual and is likely to be strongly affected by viscous/separation effects in the shadow of the hub. The predicted hub region/blade root velocity flow obviously does not reproduce this although the value of sectional lift (see Fig 7.32) is found. The high value of leading edge load at 70% span indicates the predicted velocity field just outside the hub region

is underpredicted and reflects on the crude representation of the propeller wake.

Finally, for Rudder No. 2, Figure 7.34 presents the force characteristics comparison for three advance ratios. For rudder lift the variation both with incidence and advance ratio is very good. This is true even up to 30° incidence for J=0.35 and J=0.51. This is considered to be due to the hysteresis effect induced by the cyclical variation of flow preventing stall and thereby maintaining potential flow to higher incidence. Although the exact load distribution is not predicted the total appears a reasonable representation. It is noted that the total energy of the propeller race which helps generate the lift is predicted well and so although the energy distribution within the race is not correct the overall induced load is.

The drag prediction is reasonable at low incidence but at higher incidence (>10°) only gives about 70% of the experimental value. This is due to the viscous force estimate not correctly representing the complex boundary layer flow. A similar behaviour was seen for the free-stream rudder performance.

The chordwise position of centre of pressure is within 5% of the experimental value between $-20° < \alpha < +20°$ with a better prediction at J=0.35. The deviation for greater incidence is due to the viscous flow/stall effects altering the shape of the pressure distribution and hence CMz and CPc.

The spanwise position of the centre of pressure is within 5 to 10%. It is likely the tip vortex has a strong effect especially at high incidence. The movement with change in J is identified by the numerical model.

The effect of the rudder on propeller thrust at different rudder incidence is shown in Figure 7.35 for J=0.51. The magnitude of the increase in thrust at $\alpha = -0.4°$ appears to be predicted well. As rudder incidence increases cross-flow effects (not modelled) at the propeller plane dominate the blockage effect of the rudder and the value of predicted propeller thrust reduces, an effect not seen in the experimental data.

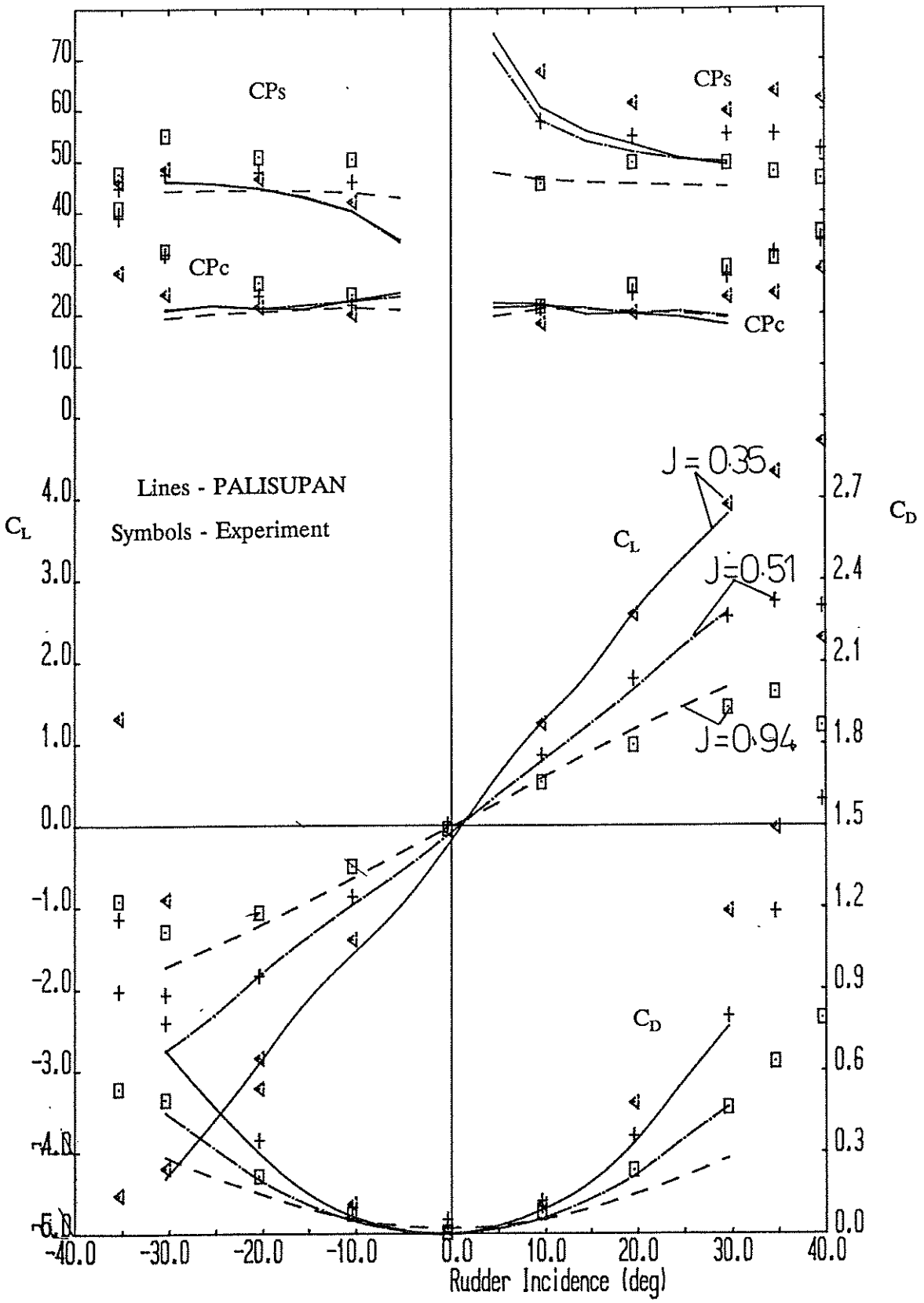


Figure 7.34 Comparison of lifting surface and experimental Rudder No. 2 force characteristics for J =0.94, 0.51 and 0.35.

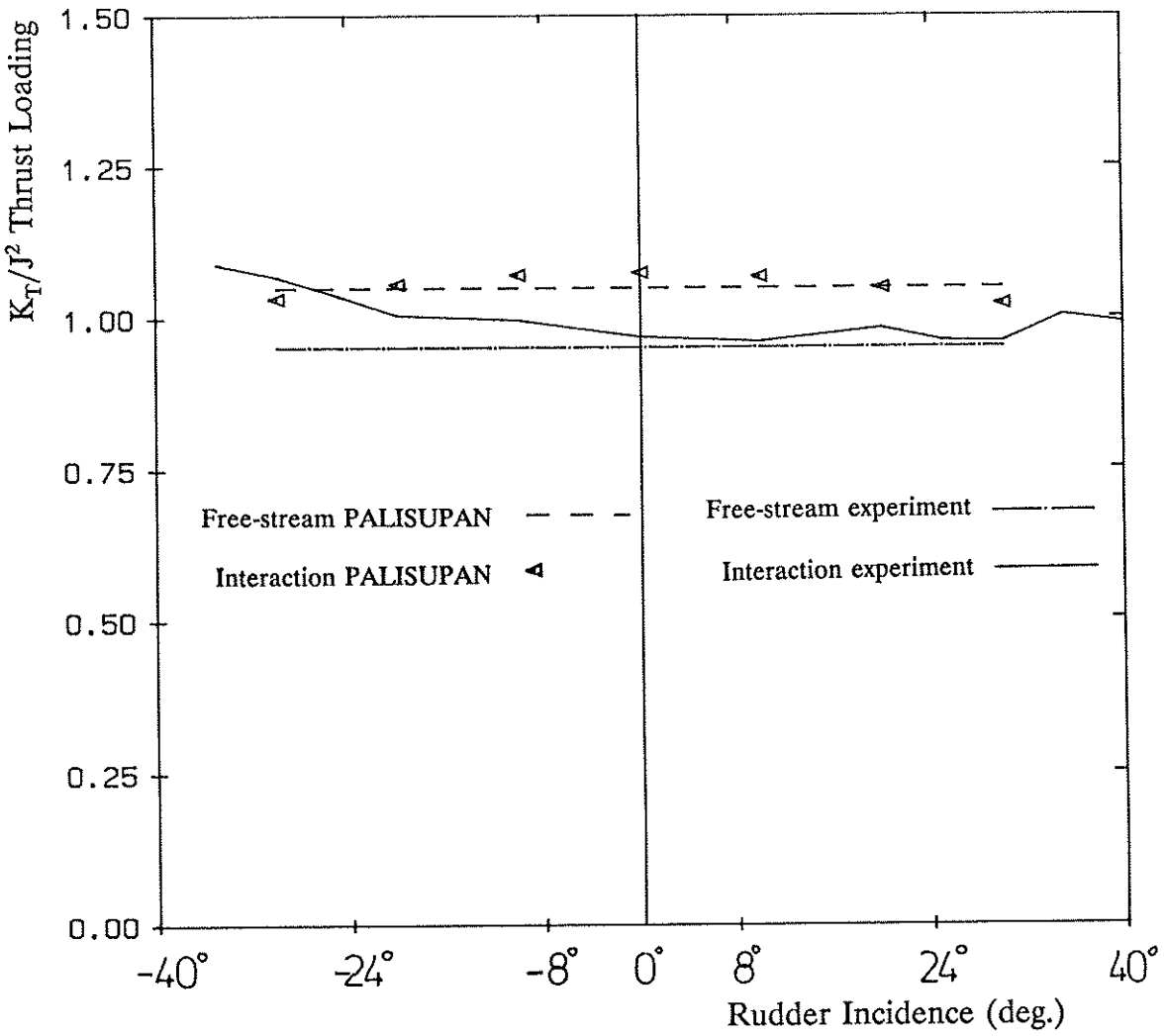


Figure 7.35 Variation of predicted and experimental propeller thrust loading with rudder incidence at J=0.51

The limitation on the number of panels available to model the hub and propeller has restricted the validation of the interaction effect on the propeller's performance. However, it appears that the flow effect produced on the propeller is acceptable for low values of rudder incidence (<10°).

7.5.8 Summary

The main aim of this validation has been to ensure that the number of panels available during the development of PALISUPAN are adequate to model rudder-

propeller interaction for the prediction of rudder characteristics for use in ship manoeuvring studies.

The restrictions are acceptable as side-force is predicted within a few percent for a realistic range of propeller thrust. The limited number of empirical corrections should allow confidence in using the method for different rudder/propeller arrangements.

7.6 Future Extension of work to include hull effects

A ship rudder-propeller combination works in the presence of a hull. The flow over the hull influences both the inflow velocity fields of the rudder and propeller. The method developed for the interaction of the two lifting-surfaces can be extended to include the ship hull. The inflow velocity field for each body would consist of the vector sum of the interaction velocity field of the other two bodies. This has not yet been attempted. No major problems are anticipated with such an approach.

7.7 Conclusion

The lifting surface panel code has been validated against the experimental test cases for free-stream rudder and propeller and for the interaction of a rudder-propeller combination. A good comparison was obtained. It is considered that the difference between theory and experiment is due mainly to the inviscid potential approximation to the flow and the limited number of panels (400) available during the development of PALISUPAN. Other discrepancies result from the use of a time-averaged inflow velocity field and the fixed propeller wake. The latter two constraints are imposed to allow the results to be obtained in a reasonable time and the obtained differences are acceptable. Overall, the method is a useful tool which should model parametric trends well, although an error of up to 10% in calculating rudder side-force is to be expected. The next chapter discusses the parametric experimental study and the use of the IVF method to derive rudder-propeller design data.

CHAPTER EIGHT

PARAMETRIC STUDY OF SHIP RUDDER-PROPELLER INTERACTION

8.1 Introduction

The development of test rig for carrying out wind tunnel tests of ship rudder-propeller interaction allowed a detailed study of the various parameters which control the manoeuvring performance of rudders and propellers to be carried out. This chapter details the parametric wind tunnel study. The validation of the numeric method allowed the range of the parameters varied in the wind tunnel study to be extended theoretically.

A first stage of the parametric study was the determination of the geometrical and flow parameters which control the interaction of an isolated ship rudder and propeller (see Appendix A). The relative importance of these parameters on the performance of the rudder and propeller could then be assessed.

On their own, the performance characteristics of a rudder and propeller are of limited use in ship manoeuvring studies and simulations. The penultimate section to this chapter describes how the performance characteristics obtained in the parametric study were used to derive relationships for use in ship manoeuvring. Relationships are presented for the side force, drag and stall angle of a ship rudder and propeller combination for varying geometrical and flow parameters. The underlying physical basis of these relations allows confidence in their use.

The data presented in this report comes from Molland & Turnock[6.15,6.18] which gives the complete set of data acquired during the six weeks of wind tunnel tests. Some of the work presented in this chapter was used as a basis for a paper presented at the 2nd International conference on the Manoeuvring and Control of Marine Craft (Molland & Turnock[8.1]).

8.2 Parameters which govern interaction

In addressing the rudder-propeller interaction problem it is necessary to identify the various independent parameters on which the rudder forces depend (see Appendix A). In the work to date the influence of the flow over the ship hull has not been considered. This allows the underlying physics of the flow for an isolated rudder and propeller to be investigated. Once this is understood the interaction between a hull and rudder-propeller combination becomes a more tractable problem. Free-surface effects and cavitation are not included. These could not be investigated in a wind tunnel but are normally considered to be of secondary influence in determining the operating conditions of the rudder and propeller.

For a given ship (fixed rudder-propeller geometry) the two fundamental controlling parameters (for a fixed pitch propeller) are the advance ratio J and rudder incidence α. Reynolds number and yaw angle γ will be of less importance. Momentum theory indicates that propeller induced velocities are a function of propeller thrust loading (K_T/J^2) which is independent of P/D and this is a parameter commonly used in propeller-rudder interaction studies.

The geometrical groups of parameters (rudder, propeller, and position) will determine the magnitude of the influence of J and α on the performance of the rudder and propeller. To provide this information on a systematic basis was one of the aims of this work. In the experimental tests a representative ship propeller, based on the Wageningen B4.40 series, and all-movable rudders with a constant NACA0020 section were used. At present the parameters not investigated were the effect of yaw angle and rudder thickness ratio. The theoretical methods developed allow the influence of the actual propeller and rudder geometry to be investigated in more detail.

8.3 Experimental Investigation

All the tests to date have been carried out with a four-bladed propeller of

diameter 800mm which was modelled on a Wageningen B4.40. Rudder spans have ranged from 1000mm up to 1300mm with a constant thickness ratio of 0.20. Six rudders of varying geometry have been employed together with various longitudinal (X), vertical (Z) and lateral (Y) positions of the propeller relative to the rudder. Propeller revolutions were adjusted such that J varied from 0.94 down to 0.35 with nominal K_T/J^2 values of 0.05 up to 2.30.

The rudder-propeller combinations tested to date are given in Table 8.1 and are shown diagrammatically in Figure 8.1.

Table 8.1 Alternative rudder-propeller arrangements tested

Rudder			2				3	4	5	6
S mm			1000				1200	1300	1000	1000
c mm			667				667	667	800	556
AR			3.0				3.6	3.9	2.5	3.6
TR	BASE		1.0				1.0	1.0	1.0	1.0
X_1/c	0.30	0.30	0.30	0.30	0.30	0.54	0.30	0.30	0.30	0.30
X/D	0.39	0.30 0.52	0.39	0.39	0.39	0.39	0.30 0.39 0.52	0.39	0.39	0.39
Z/D	0.75	0.75	1.125	0.75 0.25	0.75	0.75	0.75	1.125	0.75	0.75
Y/D	0	0	0	-0.25	0	0	0	0	0	0
λ	1.0	1.0	0.625	0.866	1.0	1.0	1.0	1.0	1.0	1.0
ξ	0.8	0.8	0.5	0.693	0.8	0.8	0.667	0.615	0.8	0.8
P/D	0.95	0.95	0.95	0.95	0.69 1.34	0.95	0.95	0.95	0.95	0.95

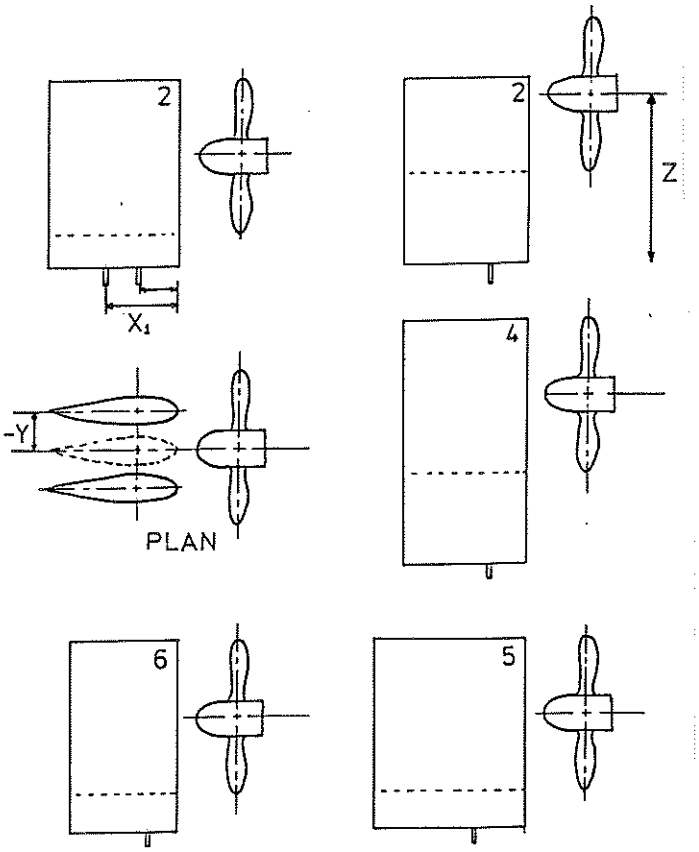


Figure 8.1 Alternative rudder-propeller arrangements tested

8.4 Discussion of the parametric experimental investigation

8.4.1 General

The variation of rudder performance with propeller advance ratio and rudder incidence was discussed in detail in Chapter 6. Overall, the shape of the rudder lift and drag curves varied little for the flow and geometric parameters tested. For rudder sideforce, the features which changed were the lift-curve slope, rudder incidence for zero lift, and the positive and negative stall angle. The quadratic shape of the drag curve was maintained although the position of minimum drag altered away from zero incidence and both above and below the free-stream drag at zero incidence.

The effect of Reynolds Number on the results obtained was not investigated in detail. However, a limited number of cases were tested to confirm that no significant scaling problems exist. These showed little difference in rudder performance as wind tunnel speed was increased and indicated that the model test regime was above the critical value Reynolds number for a fully turbulent boundary layer. Therefore, it can be inferred that full scale results are unlikely to differ to any great degree. The most sensitive feature to Reynolds Number, for manoeuvring purposes, is the stall angle. It is expected that the model tests will give conservative results such that the full scale stall angle for a given J will be greater than the model results.

The variation of rudder stock position did not give any appreciable change in rudder characteristics. With the change in rudder stock position the rudder stock torque increased but this was due to the increase in moment arm, the position of the chordwise centre of pressure remaining the same.

8.4.2 Propeller Pitch Ratio setting (P/D)

For a given advance ratio, changing the propeller pitch ratio setting alters the thrust coefficient (K_T) of the propeller. In these tests, the advance ratio was varied to maintain the same open-water thrust loading (K_T/J^2) to allow direct comparison of the rudder characteristic between the three pitch ratio settings tested. It was found, see Figure 8.2 for $K_T/J^2=2.30$, that for all the thrust loadings (0.05, 0.88, and 2.30) tested there was little change in lift characteristic and position of the centre of pressure. This confirms that it is principally the propeller thrust loading which controls the rudder's performance. However, variations were observed in the drag characteristics which suggests a dependence on the flow structure within the propeller race.

8.4.3 Rudder Aspect Ratio (AR)

The comparison of three rudders with constant coverage ξ = 0.80, and at the same longitudinal separation showed for all three advance ratios an increase of lift-

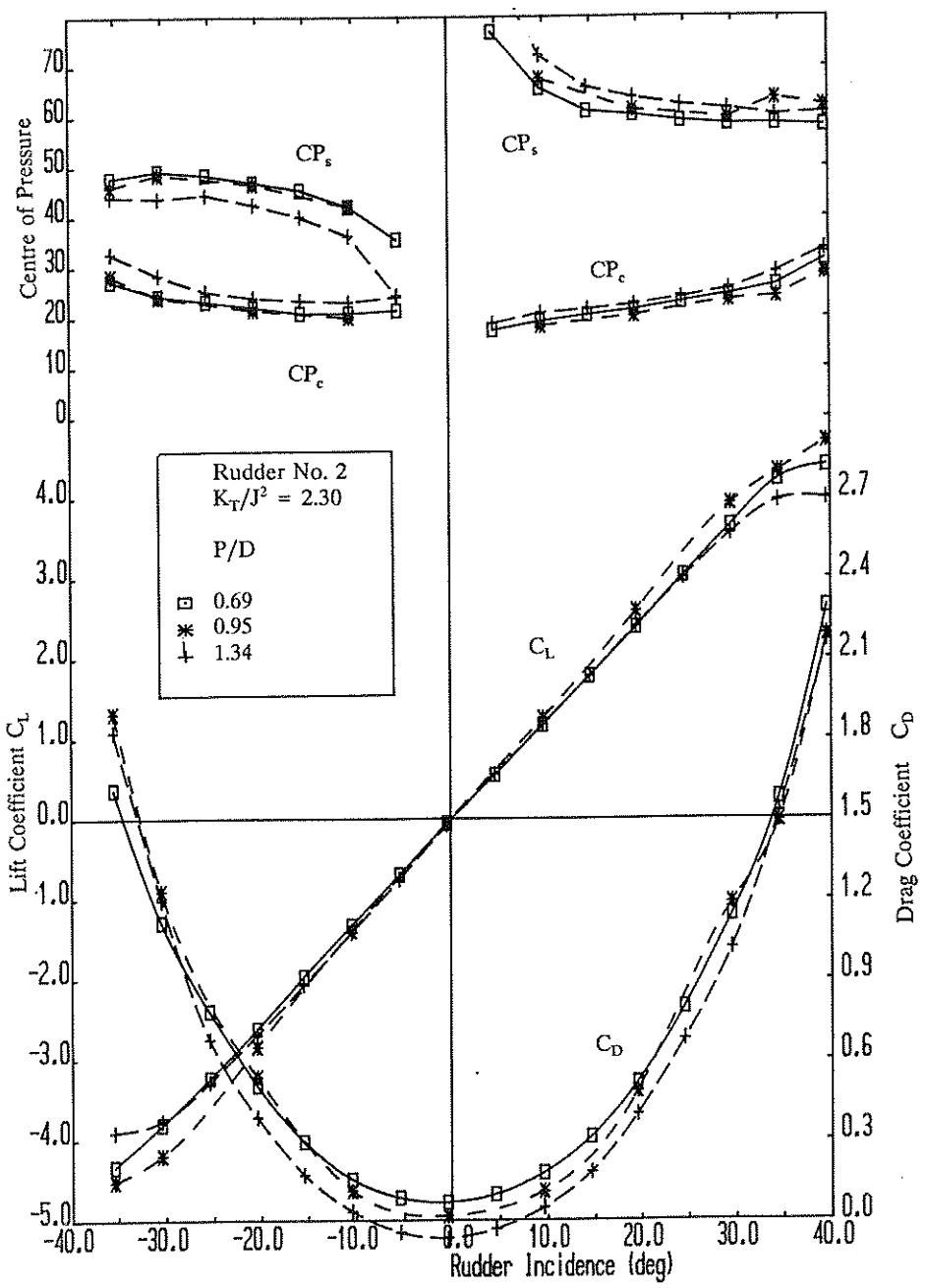


Figure 8.2 Lift, Drag and centre of pressure characteristics Effect of propeller pitch ratio **X/D=0.39**

curve slope with increase in aspect ratio. This reproduces the effect of aspect ratio observed in the free-stream and is therefore not shown. The drag and centre of pressure follow the trends with change in aspect ratio. Again, at low rudder incidence the drag appears sensitive to the structure of the propeller race. Also with decrease in J, the effect of aspect ratio is greater than that for a free-stream rudder.

8.4.4 Longitudinal Separation (X/D)

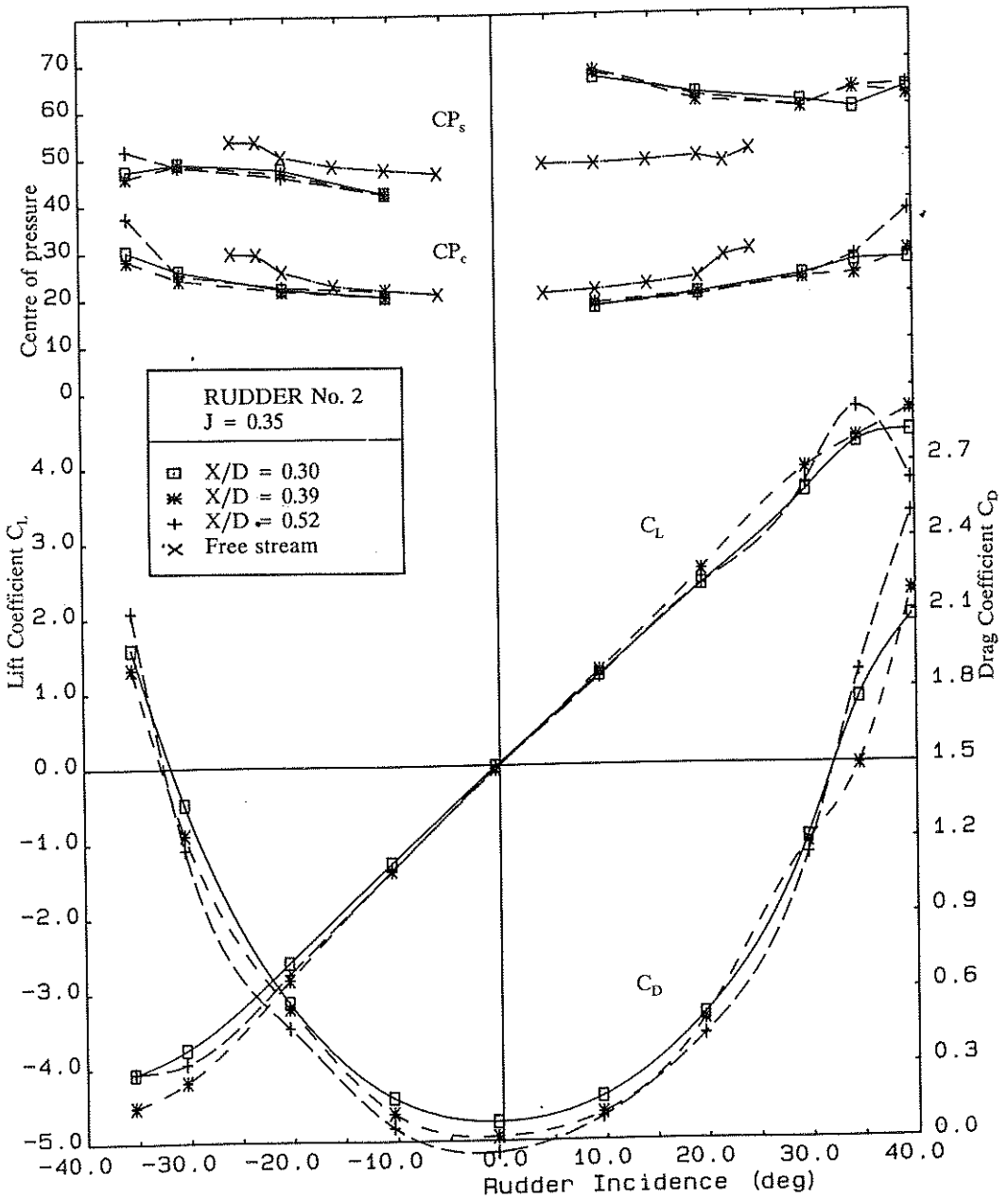


Figure 8.3 Lift, Drag and Centre of Pressure Characteristics
Effect of longitudinal separation, X/D P/D=0.95

At all advance ratios for the three separations tested for Rudder No.'s 2 and 3 there were only small changes in the rudder lift characteristics. The lift-curve slope did increase a small amount with separation, especially at low J which is shown as Figure 8.3. It should be noted that although the performance of a rudder aft of a propeller is of principal interest the rudder both blocks and diverts the flow arriving at the propeller. This acts to reduce the effective propeller inflow velocity and hence increase propeller thrust for a given advance ratio. For the minimum separation this effect is greater and this may explain the small change in lift-curve slope with longitudinal separation. The effect of the induced change in K_T on the slipstream will work in the opposite sense to the acceleration within the slipstream for fixed K_T. Taken overall the results suggest that the two effects broadly cancel each other for a fixed value of J.

8.4.5 Lateral Separation (Y/D)

In Figure 8.4, the effect of lateral separation on the performance characteristics of Rudder No. 2 is shown for an advance ratio J =0.35. The principal effect with lateral separation is the shift in the rudder incidence for zero lift. For Y/D=-0.25 zero lift occurs at +5° and for Y/D=+0.25 at an incidence of -5°. Corresponding to this is an increase in maximum lift in one direction and a decrease for the other. Likewise for the point of action of the sideforce as the rudder is moved from Y/D=-0.25 to Y/D=+0.25 the chordwise position moves forward. The spanwise position moves towards the root for negative incidence and towards the tip for positive incidence.

The lift-curve slope is no longer symmetric about the origin and for positive incidence its value decreases as Y/D changes from -0.25 to +0.25 and with the opposite shift for negative incidence.

8.4.6 Vertical Separation (Z/D)

For Rudder No. 2, moving the propeller axis height from 600mm to 900mm changed Z/D from 0.75 to 1.125 and reduced the fraction of the propeller race in way

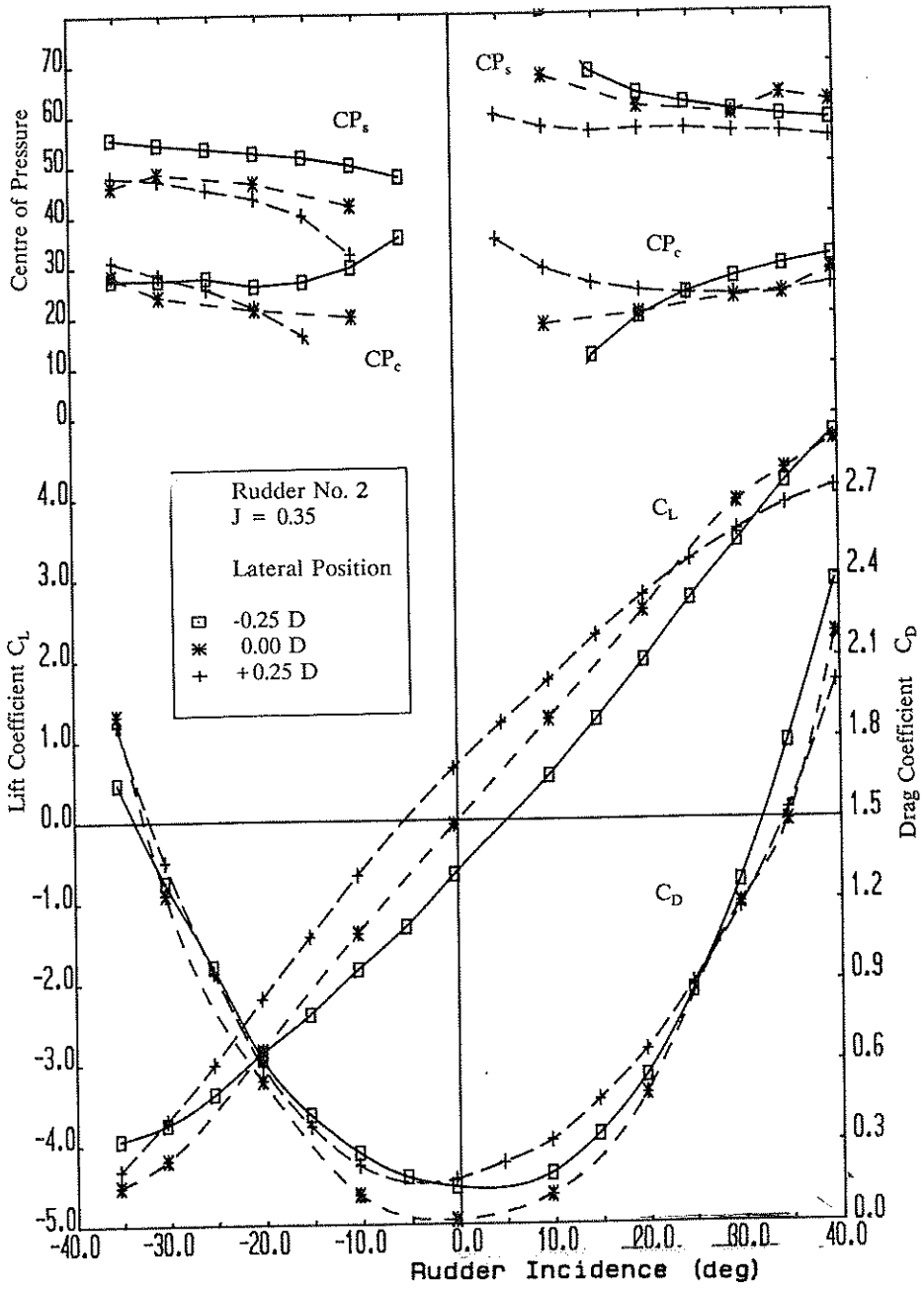


Figure 8.4 Lift, drag and centre of pressure characteristics
Effect of lateral separation Y/D

of the rudder span (λ) from 1.0 to 0.625 leading to a coverage factor (ξ) of 0.50. The effect on the rudder characteristics, as shown in Figure 8.5, is similar to that of a lateral movement. The lift characteristic is shifted so that a positive rudder incidence is required for zero lift. The positive stall angle is decreased as is the negative stall angle. The lift-curve slope decreases which is to be expected as the coverage decreased.

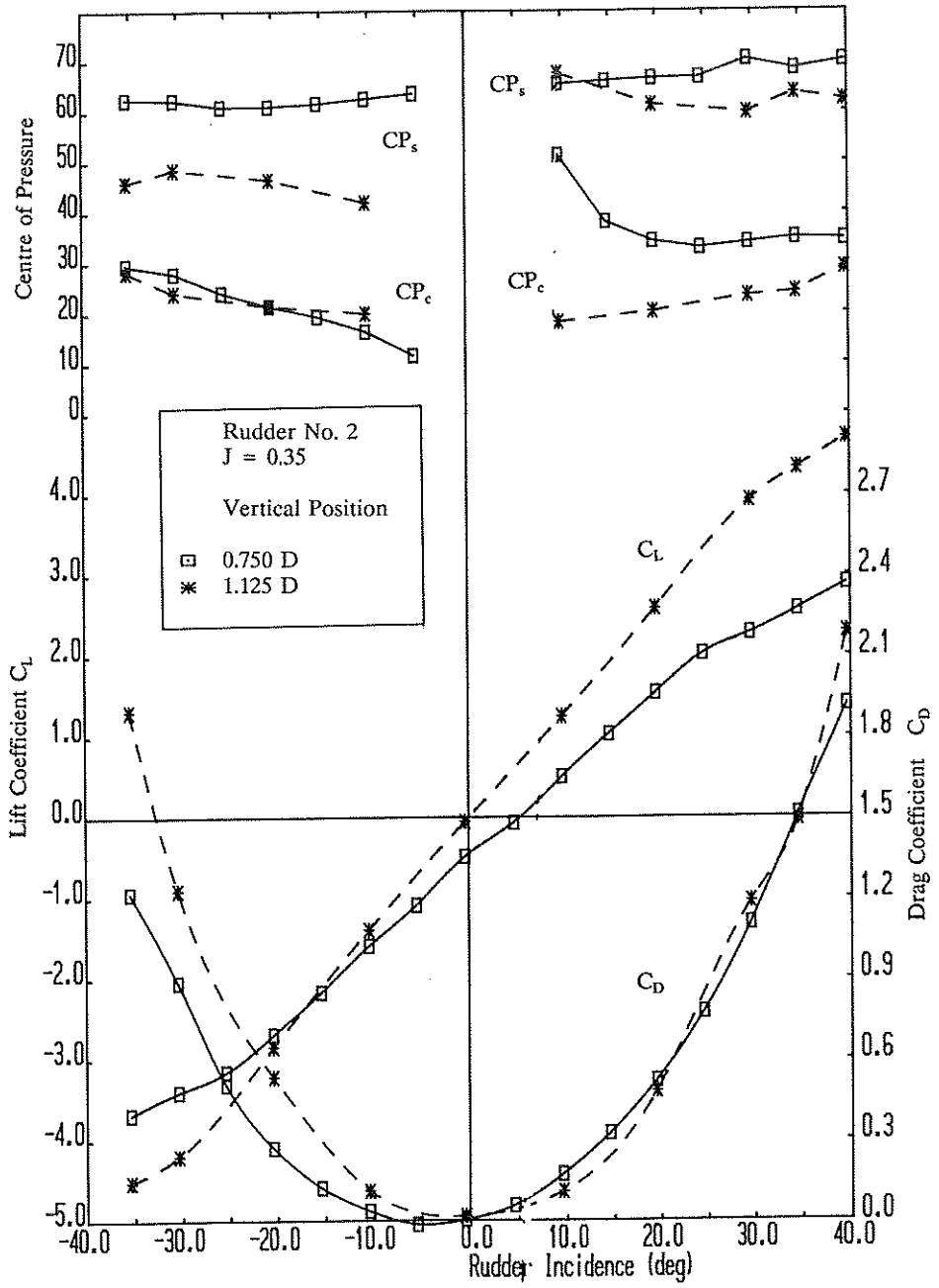


Figure 8.5 Lift, drag and centre of pressure characteristics
Effect of vertical separation

8.4.7 Coverage (ξ)

Two rudders with constant chord but spans of 1000mm and 1300mm (Rudders 2 and 4, Figure 8.1), were used to investigate the effect of coverage. For these tests the

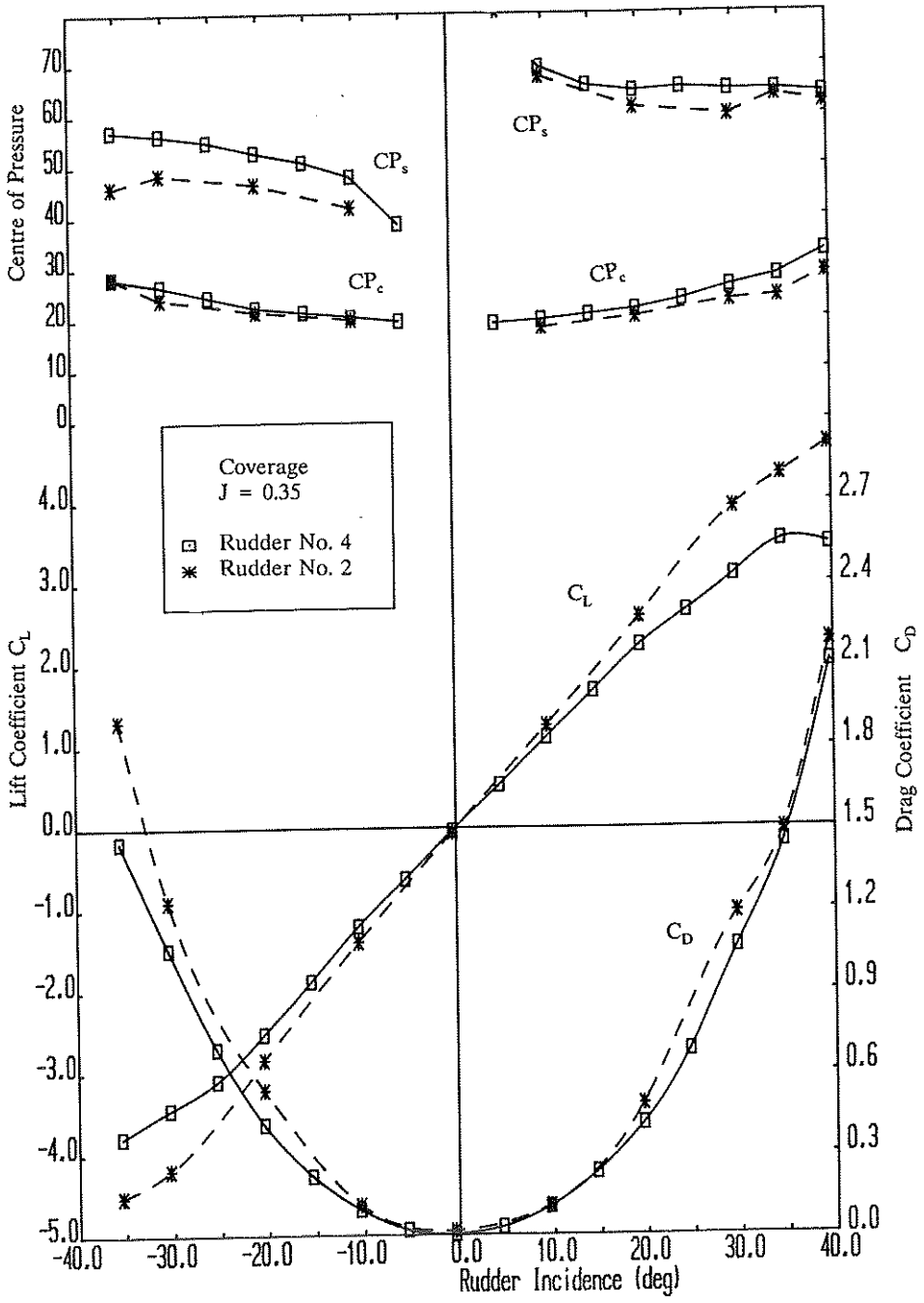


Figure 8.6 Lift, drag and centre of pressure characteristics
Effect of coverage, ξ

rudder was in way of all the propeller race so λ = 1.0. The effect of tip flow was the same for both rudders. From the results for aspect ratio it would be expected that with increase in span the lift-curve slope would increase. This is indeed observed at high J, but as J is decreased the lift-curve slope for the high aspect ratio rudder becomes progressively less than the other rudder. This effect is shown in Figure 8.6 for an

advance ratio J of 0.35. Also, the stall angle is slightly reduced. Chordwise centre of pressure is the same for both rudders for all J values. However, there is a significant shift outboard of the spanwise centre of pressure for Rudder No.4, especially at low J.

8.5 Parametric relationships for ship manoeuvring

8.5.1 General

The axial velocity at the rudder induced by the propeller is commonly taken to be a function of that predicted by simple momentum theory, and then used to scale the forces developed in the free-stream. This results in a prediction of the form

$$(V_R/V)^2 = (1 + (8/\pi)(K_T/J^2)) \qquad [8.3]$$

where V_R is the axial velocity at the rudder.

It is generally accepted that in practice the term $(1 + (8/\pi)(K_T/J^2))$ needs attenuation, a suitable amount normally being applied empirically. The method also takes no account of the influence of propeller induced rotational velocities on rudder forces. Typical applications of this approach, cited here simply as examples, are seen in Refs. [8.2,8.3,8.4].

The aim of the current work has been to understand the physics of the flow controlling the rudder performance and hence to produce relationships which are physically realistic and which reflect interaction effects within the propeller-rudder combination.

In this section relationships are given for sideforce, drag and stall. Expressions for centre of pressure have not been finalised to date. Investigations indicate that an approach similar to that for sideforce will produce relationships for rudder stock torque

and root bending moment. These will then allow CP_c and CP_s to be found.

The proposed relationships for sideforce, drag and stall, based on the work to date, are good working fits to the experimental data and are intentionally kept as simple as possible for practical applications. For this reason the relationships are given in terms of the nominal open-water propeller thrust loading rather than that actually generated by the propeller due to the blocking effect of the rudder.

It is envisaged, at this stage, that corrections to velocity due to hull wake effects would be incorporated in the normal way as corrections to K_T/J^2 and that oblique flow corrections would be applied as an incidence correction to that part of the rudder not in way of the propeller race.

8.5.2 Side force

The overall shape of the rudder lift (side force) characteristic in way of a propeller is similar to that of its free-stream performance. Figure 8.2 illustrates the characteristic shape of the lift curve between positive and negative stall angle. If a linear relationship is assumed the lift coefficient at a given incidence α becomes:

$$C_L = (\alpha - \alpha_o)\, dC_L/d\alpha \qquad [8.4]$$

where α_o is the rudder incidence for zero lift and α has not exceeded the stall angle.

To determine the relevant lift-curve slope to use for a given rudder-propeller geometry the coverage ξ of the rudder by the propeller race should be considered. The fraction of rudder not in way of the propeller ($1 - \xi$) can be considered as working with the free-stream lift characteristic $(dC_L/d\alpha)_V$. The actual lift-curve slope may then be expressed as follows:

$$dC_L/d\alpha = (dC_L/d\alpha)_V\, [\, (1 - \xi).\, 1 + \xi\, .\, k_o\,] \qquad [8.5]$$

where k_o is a function of the propeller thrust loading and is equivalent to the ratio $(V_R/V)^2$ where V_R is the effective velocity over the rudder in way of the propeller.

Assuming that when K_T/J^2 is zero k_o is unity, a quadratic in K_T/J^2 provides a good fit to the data and can be expressed as:

$$k_o = 1.0 + 1.16(K_T/J^2).(1 - 0.12.(K_T/J^2)) \quad [8.6]$$

Substituting and rearranging gives the final expression for C_L as

$$C_L = (\alpha - \alpha_o) .(dC_L/d\alpha)_V . [1.0 + \xi\ 1.16(K_T/J^2)(1 - 0.12.(K_T/J^2))] \quad [8.7]$$

The free-stream lift-curve slope $(dC_L/d\alpha)_V$ of the rudder may be found from the following empirical expression which corresponds satisfactorily with the free-stream results of the present work and with other free-stream published data:

$$(dC_L/d\alpha)_V = 1.9\pi \ / [(57.3 (1 + 3/AR)] \quad [8.8]$$

For lateral or vertical displacement of the propeller where $\lambda < 1$ the zero-lift incidence can be expressed empirically as a quadratic function of thrust loading.

$$\alpha_o = 9.0 [1 - 2 \text{ ABS } (\lambda - 0.5)]^{0.3} \xi\ (K_T/J^2) (1 - 0.18 (K_T/J^2) \quad [8.9]$$

The multiplication factor in terms of λ has a maximum when $\lambda = 0.5$ which corresponds to the maximum value of α_o and is zero when $\lambda = 0.0$ or 1.0.

For zero lateral separation and when the fraction of the propeller race arriving at the rudder (λ) is unity the zero lift incidence α_o is zero. The value of λ is unity if the rudder is in way of all the propeller race. However, if the vertical displacement of the propeller axis (Z) results in some of the propeller being clear of the rudder, λ can be found as follows:

$$\lambda = 0.5 + ((S - Z)/D) \qquad [8.10]$$

Using the properties of a circle a lateral separation Y<>0.0 can also be expressed as an equivalent λ as follows:

$$\lambda = [\, 1 - 4\,(Y/D)^2 \,]^{0.5} \qquad [8.11]$$

The proposed expressions model the variation in rudder performance with rudder aspect ratio, coverage, and lateral and vertical separation. The effect of longitudinal separation was small and has not been included. Also, the difference from the mean lift-curve slope for positive and negative incidence with lateral separation has not been included, but can be found by inspection of Fig 5 and is of the order of 10% of k_o.

It should be noted that the sign of α_o depends on the direction of rotation of the propeller.

8.5.3 Stall Angle

The periodic nature of the propeller race delays stall on the rudder. Even for high J ($K_T \rightarrow 0$) the stall can occur at up to +35° and - 28° compared with about 20° in the free-stream. The actual stall angle measured is sensitive to free-stream Reynolds number but only by a small amount (+3° to +4° for instance when Rn increases from $1x10^6$ to $3x10^6$ and very little thereafter). By estimating the stall angle from the shape of the lift curve an approximate relationship for stall was developed with a form similar to that for lift. That is:

$$+\alpha_{stall} = (10 + 30/AR)\,[\, 1 + \xi\,(\, 0.84 + 0.33\,(K_T/J^2)(\, 1 - 0.1\,(K_T/J^2))] \qquad [8.12]$$

$$-\alpha_{stall} = -(10 + 30/AR)\,[\, 1 + \xi\,(\, 0.26 + 0.69\,(K_T/J^2)(\, 1 - 0.2\,(K_T/J^2))] \qquad [8.13]$$

It should be emphasised that the above are approximate relationships, but they reflect the nature of the asymmetry of rudder stall in way of the propeller and its dependence on thrust loading.

8.5.4 Drag

In the manoeuvring situation rudder drag will be a significant component of resistance at larger rudder angles. Many manoeuvring models make simplified assumptions for this component, such as assuming it to be the longitudinal component of the rudder normal force. The level of the detailed measurements in the current work allows a more accurate assessment of rudder drag to be made for change in rudder incidence (hence C_L) and propeller thrust loading K_T/J^2.

Assuming total drag to be made up in the usual way,

$$C_D = C_{Do} + C_{Di} \qquad [8.14]$$

C_{Do} is assumed to remain constant for different angles of attack and, for given AR, $C_{Di} = f(C_L^2)$ hence

$$C_D = C_{Do} + k_1.C_L^2 \qquad [8.15]$$

Although there is considerable movement of C_{Do} at zero incidence the available data indicate that in the presence of the propeller for all cases investigated, an average practical value for C_{Do} and a working fit for k_1 may be assumed as follows:

$$C_{Do} = 0.05 \qquad [8.16]$$

$$k_1 = 0.105 (2.0 - (K_T/J^2)^{0.35}) \qquad [8.17]$$

noting that k_1 decreases as K_T/J^2 increases. The high value for C_{Do} reflects the fact

that due to the flow incidence imparted by the propeller there will be a net induced drag even at zero incidence.

Inspection of the results for the various rudder and propeller configurations indicates that, in the presence of the propeller, AR has a small influence on the factor k_1 (AR in the range 2.5-3.9) and, similarly, coverage ξ has only a small influence on k_1. In both cases the change in C_L has adequately reflected the geometric changes in rudder properties.

Overall this expression models C_D well except in the case of $\lambda < 1$ for vertical separation and to a lesser extent lateral separation. In these cases the C_D curve is displaced with incidence (eg. see Figs 8.5 and 8.6).

8.6 Design of a ship rudder-propeller arrangement

8.6.1 General

The parametric wind tunnel tests have provided a valuable set of data which can be used for the derivation of practical relationships for use in manoeuvring simulators (see previous section) and for the detailed design of an effective rudder-propeller combination. The lifting surface panel program (PALISUPAN) allows the range of geometric and flow parameters to be extended. In this section, a range of design data is presented. This has been derived from the experimental tests and PALISUPAN. As an introduction a description is given of the physical mechanisms which control the total rudder lift and drag.

8.6.2 Physics of Interaction

As has been shown previously, the flow acceleration and incidence induced by the propeller has a significant effect on the spanwise sectional lift coefficients of the rudder. The results presented have been for the quasi-steady or average value of

sectional coefficient. However, the actual surface pressure at a given instant may vary considerably from its mean value. It is the continual variation in local surface pressure (velocity) which is thought to be the mechanism which delays the sectional stall to a higher incidence. The results indicate that as the frequency and strength of the velocity fluctuation increase (decreasing advance ratio J) the sectional stall angle increases. It has been found that though the flow is highly three-dimensional and unsteady the use of the quasi-steady (circumferentially) averaged inflow gives a reasonable approximation to the flow physics thus confirming the conclusion of Cho and Williams [8.6].

Table 8.2 Spanwise variation of lift-curve slope for free-stream and J = 0.94, 0.51 and 0.35

Station	Span (mm)	Free-stream	Advance Ratio		
			J = 0.94	J = 0.51	J = 0.35
		$(dC_l/d\alpha)$	(Ratio of $(dC_l/d\alpha)$ to free-stream value)		
S8	970	0.0304	0.939	1.17	1.42
S7	940	0.0258	1.078	1.616	2.24
S6	830	0.0278	1.137	2.302	4.317
S5	700	0.0317	1.063	2.205	3.710
S4	530	0.0352	0.952	1.861	2.777
S3	400	0.0365	1.101	2.145	3.548
S2	230	0.0373	0.971	1.464	1.745
S1	70	0.0360	0.939	1.025	1.110

Table 8.2 presents the ratio of sectional lift slope to the free-stream lift slope at a given spanwise position for the three test advance ratio at the base geometry. The increase in sectional lift slope with advance ratio and position within the propeller race is clearly demonstrated. These values were obtained from the experimental values of spanwise load distributions. In addition, the experimental data shows lift at zero incidence thus indicating the presence of induced drag.

The total drag of a free stream rudder at incidence can be sub-divided into components resulting from:

1) pressure form drag;
2) induced drag due to the generation of lift;
3) skin friction due to viscous boundary layer;
4) viscous pressure drag due to modification of external flow by boundary layer;
5) root and tip vortical flows.

At zero incidence the induced drag is zero and there is no tip vortex. However for a rudder in way of a propeller race even at zero incidence the sectional data indicates that lift is generated. Therefore, there is an induced drag component. The total lift will be zero if the rudder is completely in way of the propeller race as the lift due to the two halves of the race will cancel. This explains why there is lift at zero incidence for lateral separation (Y/D), where the propeller extends beyond the rudder, and for a tapered rudder.

Returning to the rudder drag, unlike the lift the induced sectional drag is additive and therefore at zero incidence there will be an induced drag component. The

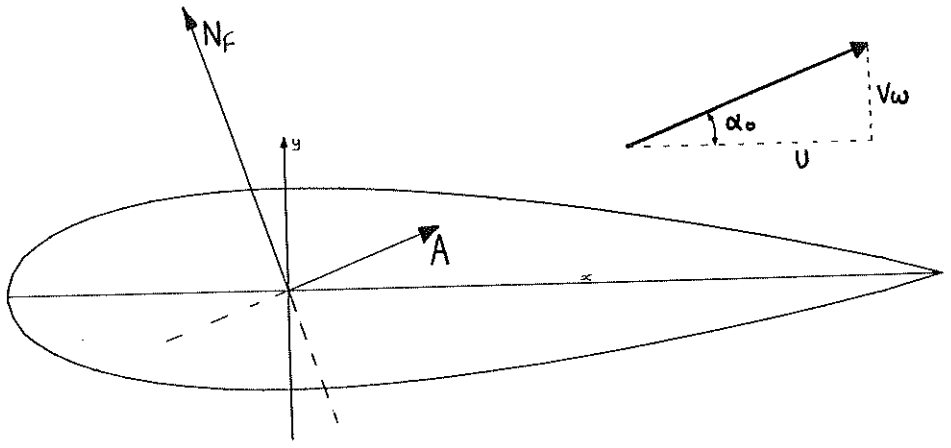


Figure 8.7 Schematic showing forward rotation of Normal and axial force vectors for rudder section with inflow incidence α_o

accelerated flow field will also increase the skin friction and alter the form and viscous pressure drag. This analysis would indicate that drag should always increase with decreasing advance ratio. However for change in pitch ratio setting, as found in this work and by Stierman[8.5], rudder drag can reduce and even result in a propulsive thrust with decreasing J. The main mechanism which gives rise to this drag reduction close to zero incidence is thought to be due to the non-zero effective sectional incidence. Figure 8.7 illustrates the principle which is similar to the operation of a sail. Normally, the sectional normal force N_f is defined perpendicular to the rudder chord and the axial force A along the rudder chord. Lift L is measured in a direction perpendicular to the main ship axis (or in this case the wind tunnel centre line) and the drag d along the main ship axis. For a rudder at incidence α the two systems are related by the following relations:

$$L = N_f \cos\alpha + A \sin\alpha$$
$$d = -N_f \sin\alpha + A \cos\alpha \qquad [8.18]$$

At zero incidence, the lift and normal, and axial and drag forces are in the same directions. However, if there is an effective flow incidence α_o at zero incidence the drag d will be:

$$d = -N_f \sin\alpha_o + A \cos\alpha_o \qquad [8.19]$$

If the effective incidence α_o or the lift/drag ratio becomes large enough a net propulsive thrust can be developed. This explains the propulsive thrust experienced at the higher propeller pitch ratio setting (α_o increases with pitch ratio setting). A similar effect of increasing α_o occurs with decreasing longitudinal separation. This reduction in zero incidence drag at low J can be seen in Figure 8.3. Whether this effect can be used to benefit the overall vessel performance will depend on the magnitude of the change in propeller thrust and power requirement.

An accurate knowledge of the position of the total chordwise centre of pressure

is essential for the correct sizing and design of rudder steering gear. The acceleration of flow and its unsteady nature alters the shape of the chordwise pressure distribution. The effect is generally to move the sectional centre of pressure forward of the free-stream value with decreasing advance ratio. Although at high J (low thrust) it can be aft of the free-stream value.

The root bending moment is important in determining the rudder stock size. As discussed previously the behaviour of the spanwise centre of pressure is asymmetric with rudder incidence. This is a result of the loading due to the propeller. For one rudder direction loading will move toward the tip and the other towards the root. The magnitude will increase with propeller thrust loading.

8.6.3 Design data

A parametric study was carried out using PALISUPAN to demonstrate its use in deriving data for ship design. Four geometric parameters were investigated and the results compared with experimental data and the empirical curve fits of Section 8.5. A limited number of panels (400 per body) were used for these tests. The effect of coverage ξ was not studied as it requires a greater number of spanwise strips to ensure the detail of the flow within the race is not lost. For each test case, results were obtained at -5.4°, -0.4°, and 4.6° rudder incidence and used to find the lift-curve slope at 0° and the zero lift incidence α_o. The tests used advance ratios of 0.35 and 0.51 corresponding to typical vessel low-speed and service conditions.

1) Rudder Thickness/Chord Ratio (t/c)

In Figure 8.8 the lift-curve slope is seen to increase with thickness/chord ratio. At this stage it is not possible to decide whether this is a physical or numerical result. One possible physical reason could be the extra thickness of the rudder in way of the propeller race which is dependent on the non-dimensional parameter (t/d) changes the local velocity at each part of the rudder surface. However, it is more likely that there

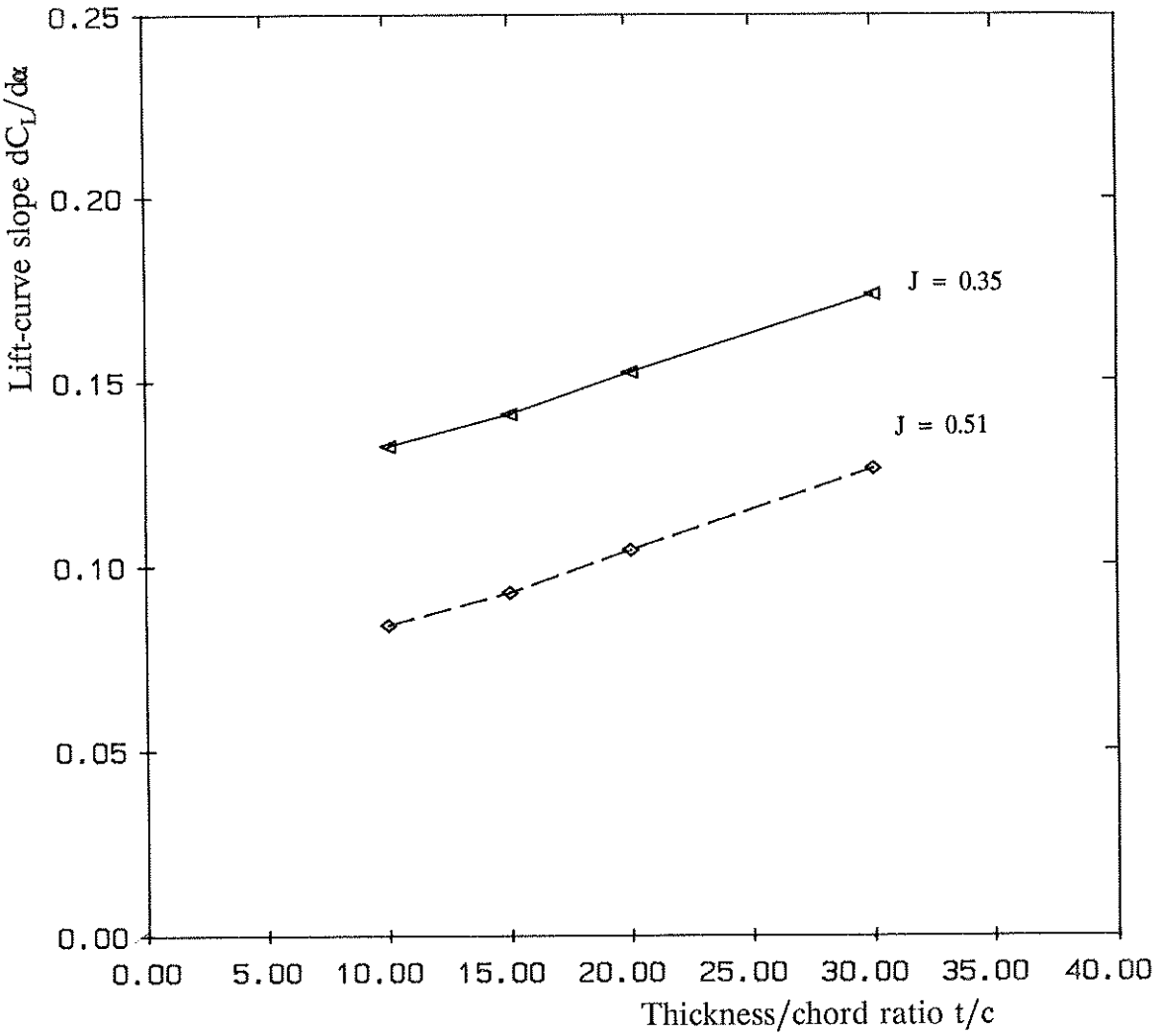


Figure 8.8 Variation of lift-curve slope with rudder thickness/chord ratio

is some numerical dependency as the thick sections will generally have a more pronounced boundary layer development and hence deviate more from potential flow conditions. This effect is noted in Ref. [8.2] for thick aerofoil theory which predicts an increase in lift-curve slope with t/c, whereas the limited number of tests for 2-D flow over the NACA 4 series sections showed a decrease with increasing t/c.

2) Effective Rudder Aspect ratio AR

The behaviour of lift-curve slope with AR is shown in Figure 8.9. the PALISUPAN results have been corrected to give a constant (t/D) ratio using the

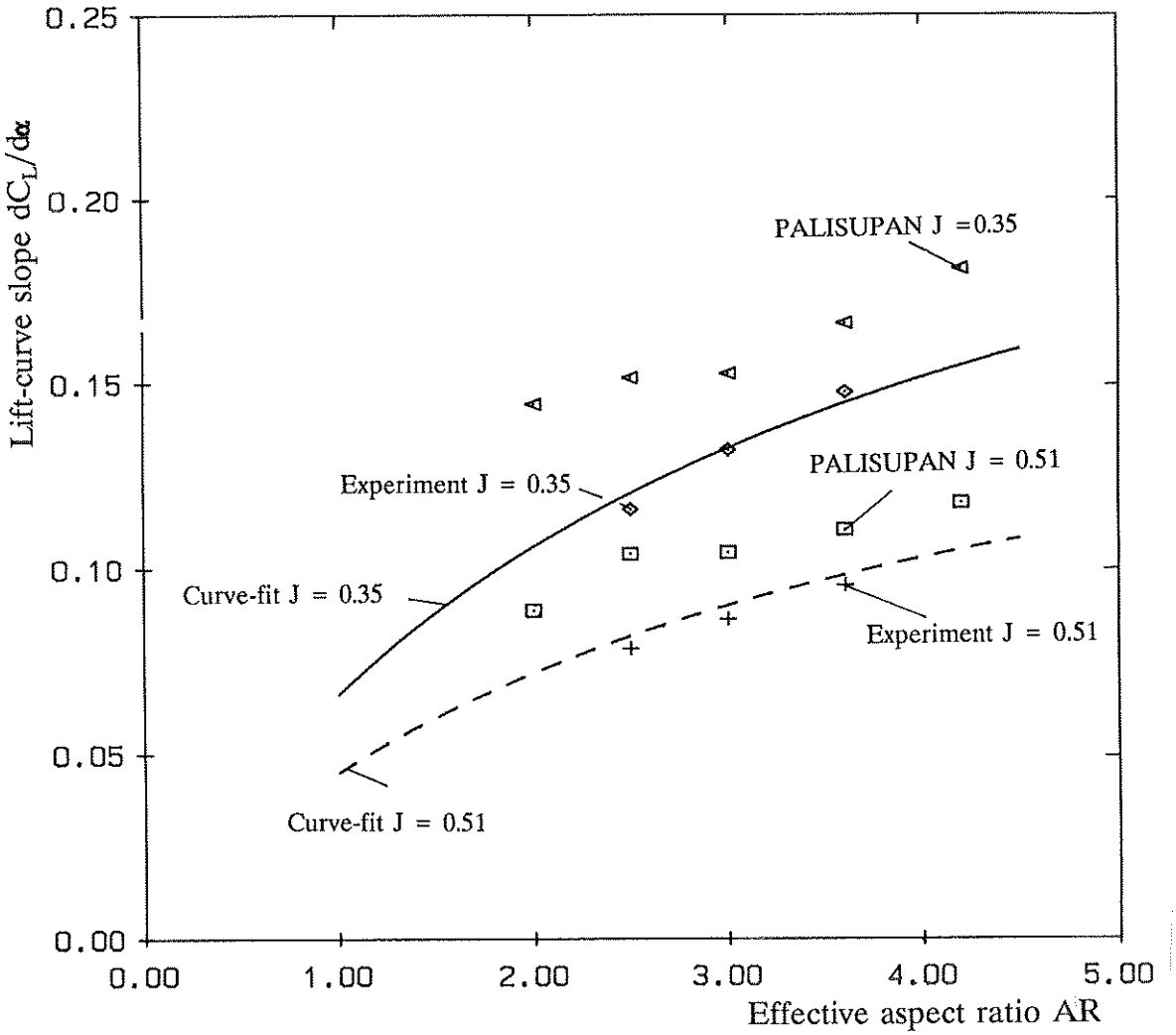


Figure 8.9 Variation of lift-curve slope with rudder effective aspect ratio

results for the lift-curve dependency on t/c. It can be seen that the empirical curve fit (as expected) gives a good fit to the experimental data. The PALISUPAN results follow the shape of the curve-fit although it appears to diverge at the lower aspect ratio. The predicted lift-curve slope is over-predicted by 10% to 20%. Reasons for this are thought to be due to the potential nature of the solution. Also, the experimental slope was calculated for values obtained at -10.4° and 9.6° whereas the numerical results were between -5.4^0 and 4.6° and the experimental slope local to zero incidence could be higher.

3) Lateral Separation (Y/D)

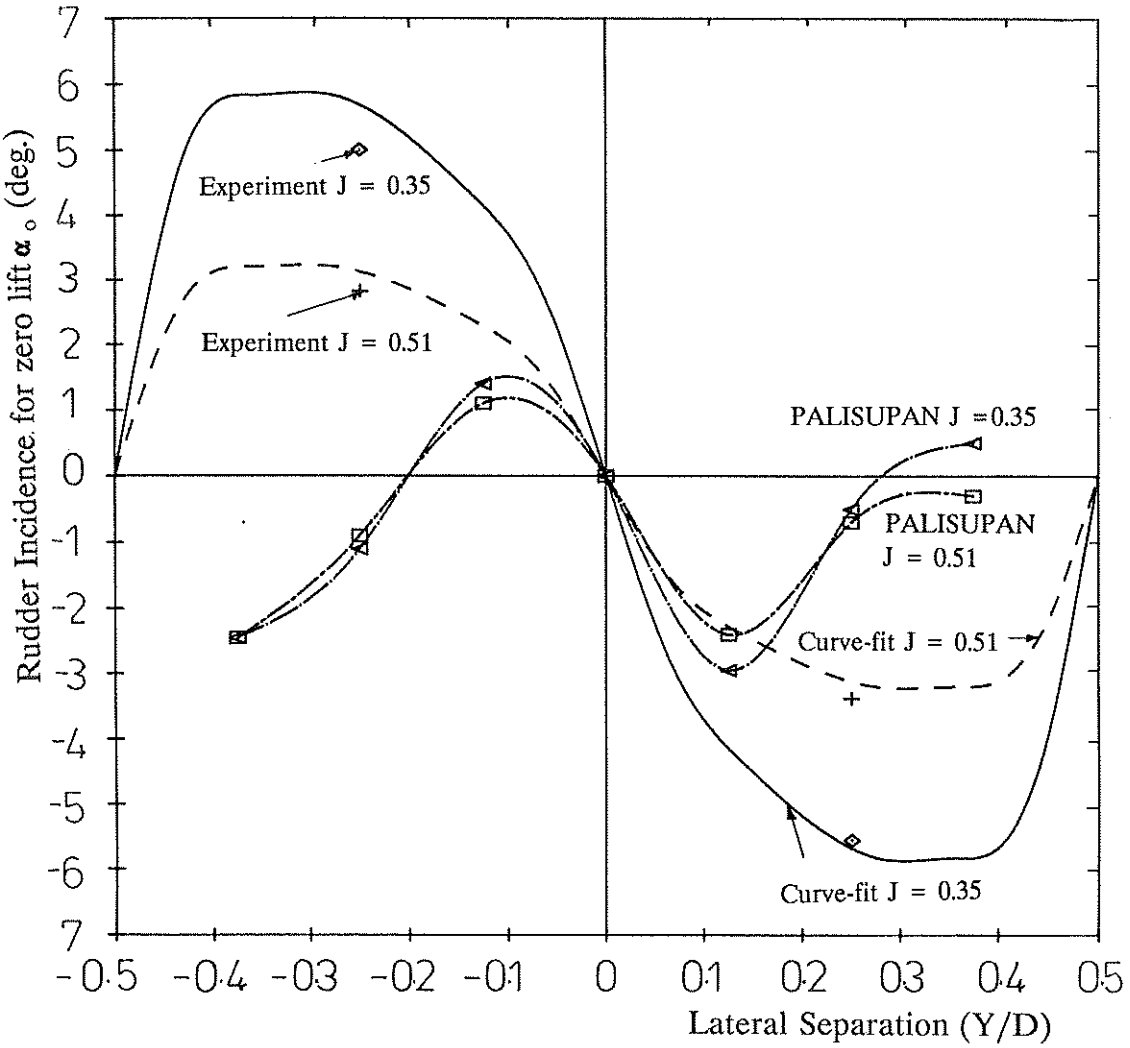


Figure 8.10 Variation of zero lift incidence with lateral separation of rudder stock and propeller axis

Figure 8.10 appears to show that in the region of a typical ship rudder-propeller shaft offsets of Y/D=0.1 PALISUPAN slightly under-predicts the zero lift angle. At the experimental test condition (Y/D=0.25) the PALISUPAN results do not predict the correct value although the shape of the curve-fit is followed. The shape of the curve-fit is also only based on three-points. The difference between theory and experiment probably indicates that, as in the discussion of spanwise loading, the rudder is deflecting the propeller race and is not accounted for in the quasi-steady interaction model.

4) Vertical Separation (Z/D)

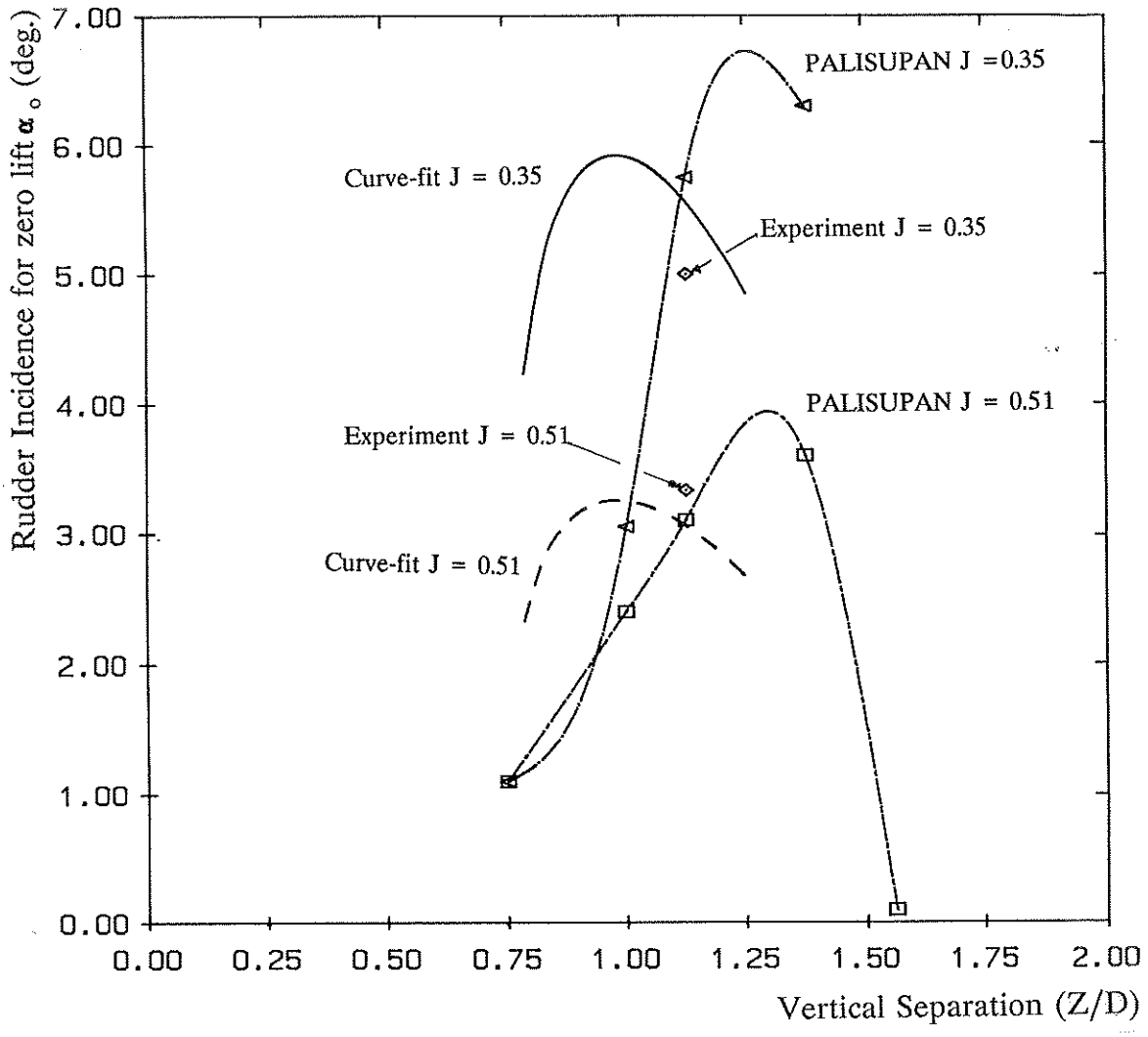


Figure 8.11 Variation of zero lift incidence with vertical separation of rudder root and propeller axis

PALISUPAN predicts the zero lift angle for a vertical offset well, especially at J=0.51, as shown in Figure 8.11. The shape of the curve-fit prediction may be in error as the numerical result seems to give a result more in keeping with the flow physics. It would be expected the maximum zero lift angle would occur when Z/D=1.25 (Z=1.0) as the rudder in this position would see only half of the propeller race which would have an effective flow incidence all in the same direction. This behaviour is found with the numerical model and not with the empirical curve-fit.

8.6.4 Summary

The results presented are for a ship at zero angle of yaw and without the effect of hull wake on propeller inflow. It is considered that if an appropriate wake velocity correction is made to the free-stream velocity the results will remain valid.

The effect of lateral and vertical separation of the propeller and rudder has a significant effect on performance. The development of rudder sideforce at zero incidence may give rise to vessel handling problems if it is not considered at the ship design stage.

8.7 Conclusion

The discussion of the results provides a deeper insight into the dependence of the interaction effects on the various geometrical properties of the rudder and propeller and their relative position. In particular:

It is found that rudder sideforce and drag have a non-linear dependence on propeller thrust loading. Stall angle is increased significantly in the presence of the propeller with different values for positive and negative incidence.

The empirical relationships derived for lift, drag and stall provide an improved model for use in ship manoeuvring simulators which takes into account the non-linear variation of rudder performance with propeller thrust loading.

The use of PALISUPAN and the experimental data to derive design data confirms the approach taken in this work and provides practical information necessary for ship design. It is seen that the Interaction Velocity Field method offers a practical design tool to allow a detailed assessment of the merits of a particular rudder-propeller arrangement. In the final chapter, overall and detailed conclusions about the work are drawn with reference to the stated aims and objectives.

CHAPTER NINE

CONCLUSIONS

9.1 Introduction

This chapter presents a compendium of the detailed conclusions resulting from the completion of the stated objectives of the work. Section 9.2 covers the conclusions resulting from the use of transputers to carry out computational fluid dynamics. The capabilities of the developed PALISUPAN code are given in Section 9.3. In Section 9.4 the effectiveness of using a large scale wind tunnel facility to investigate rudder-propeller interaction is assessed. The conclusions from the validation study of Chapter 7 are given in Section 9.5. In Section 9.6 conclusions are drawn from the theoretical and experimental study of the variation of geometric and flow parameters. Finally in Section 9.7 four overall conclusions are given with respect to the stated aims of this investigation. Recommendations are made as to further areas of study with regard to the Interaction Velocity Field method and the PALISUPAN code.

9.2 Transputers and computational fluid dynamics

[9.2.1] The transputer has been found to be a versatile computational machine suitable for solving a wide range of CFD problems. The software language Occam2 was used throughout this work. It was found that, once learnt, it provided an ideal vehicle to fully exploit the parallel capabilities of the transputer.

[9.2.2] The choice of a particular parallel algorithm depends on the application, size of problem, and the capabilities of the parallel computer to be used. On transputer based machines with distributed memory, a geometric approach to both the numeric algorithm and data distribution minimises software development time. It also allows scaling of problem size and the number of transputers used for a given size of problem.

[9.2.3] The formal separation of the C.F.D. algorithm into two parallel processes

of a communications harness and a numerical algorithm was found to allow new CFD codes to be developed independent of transputer hardware and communications. This approach appears to be the way in which the manufacturers of general parallel computers are developing their products. In the future it is expected that the harness process will be provided either as operating system software or as part of the machine hardware. The harness developed in this work provided good performance for the explicit Euler solver and implicit PALISUPAN code. However, the performance of the now commercially available harness codes are significantly better. Therefore, the CFD code performance obtained in this work will be a conservative one.

[9.2.4] The implementation of the two-dimensional Euler solver demonstrated the benefits of using a harness and the ease with which Fortran code could be transcribed into Occam2. The explicit Euler solver provided a good method of testing the performance of an array of transputers. It was found that for transputer array sizes greater than four the individual Code efficiency η was independent of the transputer array size. This implies that the speed of solution should be inversely proportional to the number of transputers which is as expected for geometric parallelism of an explicit algorithm. With a small domain of 16 by 16 finite-volume cells the code ran 9.7 times faster on 16 transputers compared to a single transputer. For a 48 by 48 cell problem a speed-up of 13.4 was obtained with 16 transputers. For transputer arrays with more than 4 transputers the time to carry out one time step was inversely proportional to the number of transputers in the array.

[9.2.5] The parallel algorithms developed for the PALISUPAN code showed that a geometric approach to data distribution and algorithm allowed software to be rapidly developed. It also demonstrated that good performance will be obtained even for implicit algorithms. The developed live-memory store process was found to be an extremely useful method for data distribution and for easy access across the array to the distributed data. Again, such techniques could potentially be included in the next generation of parallel machines as part of their operating system.

[9.2.6] The investigation into parallel methods for solving large sets of simultaneous linear equations found that the approach of direct Gaussian elimination could be made to work well across an array of transputers. However, the developed iterative block Jacobi scheme worked very well in parallel. The chosen block size influenced the performance: when the block size was equal to the number of panels in a chordwise strip the system converged most rapidly.

[9.2.7] The high code efficiencies (70% to 90%) obtained for the PALISUPAN component processes indicate that:

1) The Harness communications process performs well for the large communications requirements of implicit algorithms.

2) The use of geometric data distribution and geometric algorithms allows implicit methods to perform as well as explicit algorithms and to benefit from the use of parallel processors.

[9.2.8] The transputer used in this work is now a 5 year old design. The rapid developments in similar devices will result in even better performance of CFD codes and greater cost saving.

9.3 PALISUPAN code

[9.3.1] The use of a low-order perturbation potential approach allowed a robust code to be developed suitable for modelling a ship rudder, propeller and their interaction thus confirming the conclusion of other researchers. A Newton-Raphson iterative process was used to remove trailing edge pressure loading and satisfy the Kutta condition. The numerics of the code were satisfactorily verified against an analytic solution for flow about an ellipsoid, and against other potential methods for flow about a thin circular wing and a NACA0012 unswept wing.

[9.3.2] The developed method of surface definition using parametric cubic splines

provides a flexible and rapid means of defining complex multi-body geometries. The approach of using a geometry input file and built in surface panel generator was confirmed by the ease with which new geometries and multiple-body problems could be solved. It allowed the code performance to be easily assessed. Also, the availability of different panel size distributions allowed panels to be concentrated in areas of rapid flow variation to give a result with a minimum number of panels. The ability to define a reflection plane/axis of symmetry provided through the use of panel reflection and rotational images a method of at least doubling the number of panels available to define a given symmetrical problem. Surface pressure/velocity distribution information can be easily obtained from the numerical solution and this allows total force and moment coefficients to be evaluated. A second order finite difference scheme was found to be adequate to calculate the surface velocity and hence pressure.

[9.3.3] Expressions were derived to allow the evaluation of the velocity influence of both source and dipole panels. This allowed the velocity at any point in space around a body to be determined. This forms the fundamental process for modelling rudder-propeller interaction. It was found that a circumferential averaging process over 200 points was adequate to define the propeller inflow velocity field due to the rudder and 600 points the rudder inflow velocity field due to the propeller.

[9.3.4] The method of storage for the solution and panel geometry allowed easy access for post-processing and calculation of the overall potential pressure force. A simple empirical relationship on each panel was used to give a reasonable estimate of the contribution to total force due to the viscous skin friction.

[9.3.5] An iterative wake adaption method was implemented. It was found to be computationally expensive and was not suitable for use when calculating rudder-propeller interaction. However, with further development, it should provide a tool for developing better propeller wake models.

[9.3.6] The suite of software procedures which form PALISUPAN were

developed as independent modules allowing specific applications to be easily written. This allowed a program to be written which can carry out the four stages necessary to model rudder-propeller interaction. The overall performance of PALISUPAN in solving rudder-propeller interaction illustrates the benefits of parallel processing. On a 16 transputer array the time per complete iteration cycle would be of the order of 20 minutes. This is considered a reasonable timescale for the investigation of detailed design information.

[9.3.7] From the analysis of the calculation influence coefficient performance and that of Harness it is expected that there will be little or no degradation with increasing size of transputer array as long as there are at least 25 panels located on each transputer. The expressions given should allow reasonable estimates of the performance of these algorithms to be made for different sized arrays of transputers.

[9.3.8] A machine with 8 Mbyte per transputer could solve a 10,000 panel problem if it were of dimension 11 (121 transputers). A surface panel problem containing this number of panels is the order of the largest scale of lifting-surface problems currently solved on supercomputers (for instance a complete aircraft and associated lifting surfaces). The estimated time to solve such a system, with the order of 200 million calls to Newman panel, would be of the order of 60 minutes. A performance comparable with that of existing super-computers for a fraction of investment and running costs

9.4 Wind tunnel tests of rudder-propeller interaction

[9.4.1] A rig has been developed which allows the influence of a propeller upstream of a rudder to be accurately investigated. An automated data acquisition system allows forces, moments and surface pressures on the rudder to be measured and provides information about the thrust and torque of the propeller. The rig can be separated to carry out free-stream tests on both rudders and propellers. The rig allows parametric studies to be made of rudder-propeller interactions which provides a basis

for developing theoretical and empirical formulations for use in ship manoeuvring calculations. The results obtained are consistent with the work of others reviewed in Chapter 1. The advantage of carrying out the tests in the wind tunnel with low blockage and with large models to allow the easy measurement of quasi-steady surface pressure has been demonstrated.

[9.4.2] The rudder spanwise load distributions, obtained from a chordwise integration of surface pressure, indicate the effect of both axial and rotational rudder inflow velocities induced by the propeller. These load distributions are considered essential for a proper understanding of the flow over the rudder due to the propeller. They also illustrate the presence of the rudder tip vortex identified in earlier free-stream rudder tests. The strength of the tip vortex is increased due to the accelerated flow induced by the propeller.

[9.4.3] There is an increase in rudder lift curve slope with increase in propeller thrust loading which follows reasonably well documented trends. These increases are less than those predicted by simple momentum theory indicating that momentum theory alone does not adequately model the complex physics of the flow.

[9.4.4] There is a significant increase in stall angle, compared with free-stream, when the rudder is working aft of the propeller. This is also documented elsewhere, but reinforces the need to appreciate that very large rudder angles can be usefully employed in the lower speed/high thrust loading manoeuvring situation.

[9.4.5] There is some movement in the centre of pressure with changes in propeller thrust loading. Chordwise centre of pressure, CP_c, moves forward with increasing thrust loading and in most cases is forward of the free-stream CP_c. Spanwise centre of pressure, CP_s, moves towards the tip for one direction of rudder incidence whilst moving towards the root for opposite incidence.

[9.4.6] A large amount of well-documented data on rudder-propeller interaction

has been obtained from the wind tunnel tests. This provides a useful resource for the derivation of empirical relationships and in the validation of theoretical methods.

9.5 Validation

[9.5.1] The comparison of experimental and lifting-surface results for rudder sideforce is satisfactory for rudder incidence up to stall for all advance ratios investigated. It is considered that the prediction of lift-curve slope within 5 - 10% of the experimental value is acceptable for parametric investigations of rudder-propeller interaction. At present the method has been restricted to the use of only 400 panels to define the individual rudder and propeller geometries. The method does not predict tip loading due to the shed vortex or the effect of a viscous boundary layer. The results could be improved through the use of empirical corrections for more detailed study of rudder performance but would restrict the method to the rudder-propeller arrangement tested.

[9.5.2] The 400 panel blade and boss model of the modified Wageningen B4.40 propeller with a fixed pitch wake geometry based on Blade-Element Momentum theory gives a reasonable representation of propeller characteristics for the purposes of considering rudder-propeller interaction. The over-prediction of thrust of up to 10% is of the same order as that found for the rudder model. It is considered that the propeller blade loading and the resultant downstream velocity field is representative of that due to the actual blade loading. However, a larger number of panels than were available during development are necessary to obtain true convergence of the predicted result.

[9.5.3] The results obtained demonstrate that the crude representation of the wake is acceptable at present, although future work should allow a more sophisticated representation to be developed. It is hoped that time accurate LDA measurements will aid in the development. In the absence of such information for validation it was decided that the straight-forward definition of the wake at least allowed a consistent

approach to be used for all advance ratio.

[9.5.4] The lifting surface panel code has been validated against the experimental test cases for free-stream rudder and propeller and for the interaction of a rudder-propeller combination. A good comparison was obtained, it is considered that the difference between theory and experiment is due mainly to the inviscid potential approximation to the flow and the limited number of panels (400) available during the development of PALISUPAN. Other discrepancies result from the use of a time-averaged inflow velocity field and the fixed propeller wake. The latter two constraints are imposed to allow the results to be obtained in a reasonable time and the obtained differences are acceptable. Overall, the method is a useful tool which should model parametric trends well, although an error of up to 10% in calculating rudder side-force is to be expected.

9.6 Parametric study

[9.6.1] For a given ship (fixed rudder-propeller geometry) the two fundamental controlling parameters (for a fixed pitch propeller) are the advance ratio J and rudder incidence α. Reynolds number and yaw angle γ will be of less importance. Momentum theory indicates that propeller induced velocities are a function of propeller thrust loading (K_T/J^2) which is independent of P/D and this is a parameter commonly used in propeller-rudder interaction studies.

[9.6.2] It was found for the geometry parameters tested that for all the thrust loadings (0.05, 0.88, and 2.30) tested that there was little change in shape of lift characteristic and position of the centre of pressure. This confirms that it is principally the propeller thrust loading which controls the rudder's performance. However, variations were observed in the drag characteristics which suggests a dependence on the flow structure within the propeller race. It is found that rudder sideforce and drag have a non-linear dependence on propeller thrust loading. Stall angle is increased significantly in the presence of the propeller with different values for positive and

negative incidence.

[9.6.3] The comparison of three rudders with constant coverage ξ = 0.80, and at the same longitudinal separation showed for all three advance ratios an increase of lift-curve slope with increase in aspect ratio. This reproduces the effect of aspect ratio observed in the free-stream. The drag and centre of pressure follow the trends with change in aspect ratio. Again, at low rudder incidence the drag appears sensitive to the structure of the propeller race. Also with decrease in J, the effect of aspect ratio is greater than that for a free-stream rudder.

[9.6.4] The principal effect of changes in the lateral and vertical separation of the rudder and propeller is the shift in the rudder incidence for zero lift. The result of this is the development of rudder sideforce at zero incidence which may give rise to vessel handling problems if it is not considered at the ship design stage.

[9.6.5] An important part of this work are empirical relationships derived for lift, drag and stall provide an improved model for use in ship manoeuvring simulators which takes into account the non-linear variation of rudder performance with propeller thrust loading.

[9.6.6] The results indicate that as the frequency and strength of the velocity fluctuation increase (decreasing advance ratio J) the sectional stall angle increases. It has been found that though the flow is highly three-dimensional and unsteady the use of the quasi-steady (circumferentially) averaged inflow gives a reasonable approximation to the flow physics.

[9.6.7] The use of PALISUPAN and the experimental data to derive design data confirms the approach taken in this work and shows how the Interaction Velocity Field method can be used to provide practical information necessary for ship design.

9.7 Overall Conclusions and Recommendations for Future Work

[9.7.1] The performance of a rudder in way of a propeller can differ markedly from one operating in a free-stream. The physical reasons for the change in performance are due to the rotational and accelerated nature of the flow generated by the propeller. This work has demonstrated that a lifting-surface panel method can be used to predict the interaction of an isolated ship rudder and propeller.

[9.7.2] The Interaction Velocity Field method, developed in this work, provides a straight forward means of obtaining rudder-propeller performance characteristics. By separately modelling the rudder and propeller their mutual influence is accounted for by the modification of their respective inflow velocities. The level of accuracy of prediction allows this method to be used for carrying out parametric studies of the effect on the overall rudder side-force and propeller thrust developed. Therefore, the method is suitable for both design purposes and use in vessel manoeuvring studies.

[9.7.3] The numerical model runs on a transputer based parallel computer, which provides a versatile programming environment. The development of parallel algorithms for the lifting-surface panel method has applicability to a wide range of other fluid dynamic problems. The cost-effectiveness of transputer based parallel computers was demonstrated through the code performance obtained and the rapid software development. Future generations of parallel computers will have better performance and in the long-term will be the preferred platform for algorithm development in computational fluid dynamics.

[9.7.4] This work has only considered an isolated ship rudder and propeller. The numerical method and experimental test rig are suitable for carrying out an investigation of the modifying effect of the ship hull on the performance of the rudder-propeller combination.

Further development of the IVF method will allow even better correspondence

between theory and experiment. In particular the following enhancements should result in closer prediction:

1) modification to section shape/Kutta condition to account for reduction of circulation due to viscous flow;

2) increase in the number of panels used to ensure convergence, especially with relation to the panel density over the hub;

3) modelling of individual propeller blades in cross-flow due to rudder to enable deflection of propeller wake to be included;

4) use of L.D.A. data to develop a physically accurate representation of the propeller wake.

In summary, it has been shown that the Interaction Velocity Field method offers a practical design tool to allow a detailed assessment of the merits of a rudder-propeller combination with a particular geometry and under a range of flow conditions. The work presented has been restricted to 400 panel models of the rudder and propeller. This has limited the accuracy of prediction to the order of 5-10%. However, the developed PALISUPAN code can be directly ported onto larger size arrays of transputers. These machines will allow more detailed modelling and thereby more accurate predictions. In particular, with a greater number of panels it will be possible, with minor modifications, to model un-steady propeller flow and this will allow the effects of ground-board and propeller race deflection to be assessed. The PALISUPAN code has been developed as a general purpose surface panel code allowing easy definition of complex three-dimensional bodies. In addition to assessing the effect of rudder-propeller interaction with further development of the wake adaption it can be used to investigate detailed aspects of propeller design. The code is already in use as a tool for the investigation of flow around yacht mast-sail and hull-keel geometries. The modular approach to software design should allow future enhancement of the code to include a free-surface modelling capability. The ease of software development and the ability to use different parallel machines will make possible an evolutionary approach to the development of the PALISUPAN suite of software.

REFERENCES

CHAPTER ONE

[1.1] 18th International Towing Tank Conference, Proceedings Volume 1, Kobe, October 1987, The Society of Naval Architects of Japan.

[1.2] 19th International Towing Tank Conference, Proceedings Volume 1, Madrid, September 1990.

[1.3] 100A1, The Magazine of Lloyds Register, Issue 1 1991, pp. 22-27.

[1.4] Stierman, E.J., The Influence of the Rudder on the Propulsive Performance of Ships - Part I", International Shipbuilding Progress, Vol. 36, No. 407(1989), pp. 303-334.

[1.5] Stierman, E.J., "The Influence of the Rudder on the Propulsive Performance of Ships - Part II", International Shipbuilding Progress, Vol 36, no. 408 (1989) pp. 405-435.

[1.6] Suhrbier, K.R., An experimental investigation on the propulsive effect of a rudder in the propeller slipstream", International Shipbuilding Progress, Vol. 21, no. 234 (Feb. 1974) pp. 31-39.

[1.7] English, J.W., & Bain, D.C., " Some manoeuvring devices for use at zero and low ship speed", Trans. NECEIS, Vol. 88. 1971/1972 pp31-50.

[1.8] Mathis, P.B., & Gregory, D.L., "Propeller slipstream performance of four high-speed rudders under cavitating condition", U.S. Naval Ship Research and Development Centre, Report 4361, May 1974.

[1.9] Kerwin, J.C., Lewis, S.D., & Oppenheim, B.W., " Experiments on Rudders with small flaps in free-stream and behind a propeller", M.I.T. Department of Ocean Engineering Report 74-16, Oct. 1974.

[1.10] Kracht, A.M., "Rudder in the slipstream of a propeller", ISRP International Symposium on Ship Resistance, Shanghai China, April 11-14, 1989, No.5 ,pp. 66-71.

[1.11] Lottveit, M., " A study of rudder action with special reference to single screw ships", Trans. NECIES No. 75, 1959, Part 3.

[1.12] Van Berlekom L.B., " Effects of propeller loading on rudder efficiency", Royal Netherlands Naval College. Proceedings of fourth ship control systems symposium, the Hague, 27-31 Oct. 1975.

[1.13] Okada,"On the performance of rudders and their designs", The Society of Naval Architects of Japan, 60th Anniversary series, Vol. 11, 1966.

[1.14] Tsakonas, S., Jacobs, W.R., & Ali, M.R., " Propeller-rudder interaction due to loading and thickness effect ", Journal of Ship Research, Vol. 19, No. 2, June 1975, pp.99-117.

[1.15] Jacobs, W.R., & Tsakonas, S., " Propeller-induced velocity field due to thickness and loading effects", Journal of Ship Research, Vol. 19, No. 1, March 1975, pp.44-56.

[1.16] Molland, A.F.," A method for determining the free-stream characteristics of ship skeg-rudders", International Shipbuilding Progress, Vol. 32, No. 370, June 1985.

[1.17] Jong, J-M., "A Comparison of Parallel Implementations of the Flux corrected Transport Algorithm", Master of Science Thesis, Utah State University, 1989.

CHAPTER TWO

[2.1] INMOS Ltd.,"Transputer Reference Manual",Prentice-Hall, 1988.

[2.2] Robinson, G. & Lonsdale, R., " Fluid Dynamics in parallel using an unstructured mesh", in "Parallel processing in engineering applications" ed. Ackley, R.A., Computational Mechanics Institute, 1990.

[2.3] Lonsdale, R.D., & Webster, R., "The application of finite volume methods for the modelling three-dimensional flow on an unstructured mesh", Proceedings of the 6th international conference on numerical methods in laminar and turbulent flow, Swansea, July 1989.

[2.4] Hey, A.J.G., "Supercomputing with transputers - past, present and future",

[2.5] INMOS Ltd., "Transputer Development System", Prentice-Hall. 1988

[2.6] Hoare, C.A.R. ed., "OCCAM 2 Reference Manual", INMOS Ltd., Prentice-Hall, 1988.

[2.7] Long, L.N., Khan, M.M.S., & Sharp, H.T., "Massively parallel three-dimensional euler/navier-stokes method", A.I.A.A. Journal, Vol. 29, No. 5, May 1991, pp. 657-666.

[2.8] Pountain,D., & May, D.,"A tutorial introduction to occam programming", BSP Professional Books, 1987.

[2.9] Wake, B.E., & Egolf, T.A., "Implementation of a rotary-wing navier-stokes solver on a massively parallel computer", A.I.A.A. Journal, Vol. 29, No. 1, Jan. 1991, pp. 58-67.

[2.10] Amdahl, G., "Validity of a single processor approach to achieving large scale computing capabilities", AFIPS Conference Proceedings, 30, 19767, pp. 483-485.

[2.11] Patton, P.C., "Performance limits for parallel processors", Chapter 1, "Parallel Supercomputing: Methods, algorithms and applications.", ed. Carey, G.F. Wiley, 1990.

[2.12] Gustafson, J., "Re-evaluating Amdahl's law", CACM, 31, 5, pp. 532-533, 1988.

[2.13] Gustafson, J, Montry, G.R., & Benner, R.E., " Development of parallel methods for a 1024-processor hypercube", SIAM J. Sci. Stat. Comput., Vol. 9, No. 4, pp. 609-638, 1989.

CHAPTER THREE

[3.1] Turnock, S.R., "Parallel Implementation of an Explicit Finite Volume Euler Solver on an Array of Transputers ",ICAS Congress 1990, Proceedings Vol 2.,pp. 1557-1568.

[3.2] Ni, R.H., "A Multiple-Grid scheme for Solving the Euler Equations", A.I.A.A. Journal, Vol. 20., No.11, Nov 1982.

[3.3] M.I.T. Aeronautics and Astronautics Department, Graduate Course 16.026, Spring 1987, Assignments 2 and 3, Massachusetts Institute of Technology, Cambridge, U.S.A.

[3.4] Hall, M.G., "Cell-Vertex Multigrid Schemes For Solution of the Euler Equations", Royal Aircraft Establishment TM Aero 2029, H.M.S.O. 1985.

[3.5] Ni, R.H., "Multigrid Convergence Acceleration Techniques for Explicit Euler Solvers and Applications to Navier-Stokes Calculations.", Von Karman Institute lecture notes 1986.

[3.6] Steger, J. and Sorenson, R. "Automatic Mesh-Point clustering Near a Boundary in Grid Generation with Elliptic Partial Differential equations", J. Computational Physics, Vol 33, pp 405-410 (1979).

[3.7] Jameson, A., "Solution of the Euler Equations for Two Dimensional Transonic Flow by a Multigrid Method", MAE Report No.1613, Princeton University, June 1983.

[3.8] Hey, A.J.G., "Supercomputing with transputers - past, present and future",

CHAPTER FOUR

[4.1] Cho. J, & Williams, M., "Propeller-Wing Interaction using a Frequency Domain Panel Method", Journal of Aircraft, Vol. 27, No. 3, March 1990.

[4.2] Kleinstein, G, & Liu, C.H., " Application of airfoil theory for non-uniform streams to wing propeller interaction", Journal of Aircraft, Vol. 9, No. 2, Feb. 1972.

[4.3] Koening, C., " Influence of the propeller on other parts of the airplane structure", Aerodynamic Theory, Vol IV, ed. Durand, W.F., Dover, 1963.

[4.4] Kroo, I., "Propeller-wing integration for minimum induced loss", Journal of Aircraft, Vol. 23, No. 7, July 1986, pp. 561-565.

[4.5] Munk, M., "Minimum induced drag of airfoils", NACA Rept. 121, 1921.

[4.6] Witkowski, D.P, Lee, A.K.H., & Sullivan, J.P, "Aerodynamic interaction between propellers and wings", Journal of Aircraft, Vol. 26, No. 9, Sept. 1989, pp. 829-836.

[4.7] Lorber, P.F., and Egolf, T.A., " An unsteady helicopter rotor-fuselage aerodynamic interaction analysis", pp. 32-42, Journal of the American Helicopter Society, July 1990.

[4.8] Hess, J.L., & Smith, A.M.O.," Calculation of non-lifting potential flow around arbitrary three-dimensional bodies", Journal of Ship Research, Vol. 8, No. 2, Sept. 1964.

[4.9] Molland, A.F., "A method for determining the free-stream characteristics of ship skeg-rudders", International Shipbuilding Progress, Vol 32, June 1985.

[4.10] Kerwin, J.E., & Lee, C-S, "Prediction of steady and un-steady marine propeller performance by numerical lifting surface theory", SNAME Transactions, Vol. 86, 1978, pp. 218-253.

[4.11] Kerwin, J.E., Kinnas, S.A., Lee, J-T, Shih, W-Z, " A surface panel method for the hydrodynamic analysis of ducted propellers", SNAME Transactions, Vol. 95, 1987.

[4.12] Greeley, D.S., & Kerwin, J.E., "Numerical methods for propeller design and analysis in steady flow", SNAME Transactions, Vol. 90, 1982, pp. 415-453.

[4.13] Eckhart, M.K., & Morgan, W.G., "A propeller design method", SNAME Transactions, Vol. 63, 1955.

[4.14] Hess. J.L., "Panel Methods in Computational Fluid Dynamics", Annual review of fluid mechanics. 1990. Vol 22. pp.255-274.

[4.15] Lamb, H.,"Hydrodynamics", Cambridge University Press, sixth edition, 1932.

[4.16] Hess, J.L., "The problem of three-dimensional lifting flow and its solution by means of a surface singularity distribution", Computational Methods Applied Mechanical Engineering, 4:283-319, also 1972, Rep. MDC-J5679, McDonnell Douglas Aircraft Co. Long Beach, Calif.

[4.17] Lee, J-T, "A potential based method for the analysis of marine propellers in steady flow", Ph.D. thesis, M.I.T. Dept. of Ocean Engineering, Aug. 1987.

[4.18] Morino, L., & Kuo, C-C, " Subsonic Potential aerodynamics for Complex Configurations: A general theory", A.I.A.A. Journal, Vol 12., No. 2 Feb. '74.

[4.19] Maskew, B., "Prediction of subsonic aerodynamic characteristics: A case for low-order panel methods", Journal of Aircraft, Vol 19, No. 2, Feb. 1982, pp. 157-163.

[4.20] Margesson, R.J., Kjelgaard, S.O., Sellers, W.L., Morris, C.E.K., Walkley, K.B., & Shields, E.W.," Subsonic panel methods - a comparison of several production codes", Proceedings of AIAA 23rd Aerospace Science Meeting, Reno, Nevada, Jan. 1985.

[4.21] Newman, J.N.,"Distribution of sources and normal dipoles over a quadrilateral panel". Journal of Engineering Mathematics,Vol 20., pp113-126, 1986.

[4.22] Turnock, S.R., "Lifting surface panel method for modelling ship rudders and propellers". Ship Science Report No. 50, University of Southampton. 1992.

[4.23] Maitre, T.A., and Rowe, A.R., "Modelling of flow around a propeller using a potential based method", Journal of Ship Research, Vol 35., No. 2, June 1991, pp.114-126.

[4.24] Kreyszig, E., "Advanced Engineering Mathematics". pp. 781, 5th edition, John Wiley & Sons, 1983.

[4.25] Schlichting, H., " Boundary Layer Theory", McGraw-Hill, New York 1968.

[4.26] Jordan, P.F., "Exact solutions for lifting surfaces", AIAA Journal, Vol 11., No. 8, Aug. 1973, pp. 1123-1129.

CHAPTER FIVE

[5.1] Allwright, J.,"Parallel algorithms for combinatorial optimization on transputer arrays", Ph.D Thesis, Department of Electronics and Computer Science, University of Southampton, 1990.

[5.2] Rua, R., "Parallel computation of eigenvalues and eigenvectors using occam and transputers", Ph.D. Thesis, University of Southampton 90--076060, 1990.

[5.3] Kreyszig, E., "Advanced Engineering Mathematics". pp. 809 & 813, 5th edition, John Wiley & Sons, 1983.

[5.4] Clark, R.W., "A new iterative matrix solution procedure for three-dimensional panel methods", Proceedings of AIAA 23rd Aerospace Science Meeting, Reno, Nevada, Jan. 1985.

[5.5] Kerwin, J.E., & Lee, C-S, "Prediction of steady and un-steady marine propeller performance by numerical lifting surface theory", SNAME Transactions, Vol. 86, 1978, pp. 218-253.

CHAPTER SIX

[6.1] Kracht, A.M.,"Ship-propeller-rudder interaction", presented at the 2nd International symposium on propellers and cavitation, 1st-4th Sept. 1992, Hangzhou, China.

[6.2] Turnock, S.R., "A Test Rig For The Investigation of Ship Propeller/Rudder Interactions", Ship Science Report No. 45, University of Southampton, November 1990.

[6.3] Turnock, S.R.," Computer aided Design and Numerically Controlled Manufacture of a Split mould for a Composite Model Ship Propeller", University of Southampton, Ship Science Report 42, December 1990.

[6.4] Molland, A.F.,& Turnock, S.R.," The Design and Construction of Propeller Blades in Composite Materials for a Wind Tunnel Model". University of Southampton, Ship Science Report 41, October 1990.

[6.5] Molland, A.F.,& Turnock, S.R.,"The design and construction of model ship propeller blades in hybrid composite materials", Composites Manufacturing, Vol. 2, No. 1, March 1991.

[6.6] Molland, A.F., & Turnock, S.R.," The Design, Construction and Calibration of a Thrust and Torque Dynamometer for a Wind Tunnel Propeller Model", University of Southampton, Ship Science Report 44, October 1990.

[6.7] Abbott,I.H. & Von Doenhoff, A.E.," Theory of Wing Sections", Dover Publications. 1958.

[6.8] Molland, A.F." The Design, Construction and Calibration of a five-component strain gauge wind tunnel dynamometer " , University of Southampton, Ship Science Report 1/77, Nov. 1976.

[6.9] Molland, A.F., "The free-stream characteristics of ship skeg-rudders", PhD. thesis, Faculty of Engineering and Applied Science, University of Southampton, 1982.

[6.10] Molland, A.F.," Pressure-distribution investigation of a semi-balanced ship skeg-rudder", University of Southampton, Ship Science report No. 5/81, 1980.

[6.11] Oosterveld, M.W.C., & Oossanen, P. Van, "Further computer analysed data of the Wageningen B-screw series", International Shipbuilding Progress, Vol 22, No 251, July 1975.

[6.12] Barnsley, M.J., & Wellicome, J.F., "Design and Testing of a Horizontal Axis Wind Turbine Model for the Investigation of Stall Regulation Aerodynamics", Proceedings of the European Wind Energy Conference 1989, pp. 724-728, Peter Peregrinus Ltd, 1989.

[6.13] Turnock, S.R., " Two-Dimensional Wind Tunnel Tests of SSS Series 1 Structurally Efficient Aerofoils for Wind Turbines", University of Southampton, Ship Science Report 38, June 1989.

[6.14] English, J.W., & Bain, D.C.," Some manoeuvring devices for use at zero and low ship speed", Trans. NECEIS, Vol. 88, 1971/72.

[6.15] Molland, A.F., & Turnock, S.R.," Wind Tunnel Investigation of the Influence of Propeller Loading on Ship Rudder Performance", University of Southampton, Ship Science Report 46, March 1991.

[6.16] Goodrich, G.J., & Molland, A.F.," Wind Tunnel Investigation of semi-balanced ship skeg rudders", Trans. RINA, Vol. 121, 1979.

[6.17] Molland, A.F., & Turnock, S.R.," Wind Tunnel Test Results for a Model Ship Propeller Based on a Modified Wageningen B4.40 ", University of Southampton, Ship Science Report 46, December 1990.

[6.18] Molland, A.F., & Turnock, S.R.," Further wind tunnel tests on the influence of propeller loading on ship rudder performance", University of Southampton, Ship Science Report No. 52, 1992.

CHAPTER SEVEN

[7.1] Molland, A.F.,"A method for determining the free-stream characteristics of ship skeg-rudders", International Shipbuilding Progress, Vol. 32, No. 370, June 1985.

[7.2] Abbott,I.H. & Von Doenhoff, A.E.," Theory of Wing Sections", Dover Publications. 1958.

[7.3] Maitre, T.A., and Rowe, A.R., "Modelling of flow around a propeller using a potential based method", Journal of Ship Research, Vol 35., No. 2, June 1991, pp.114-126.

[7.4] Lee, J-T, "A potential based method for the analysis of marine propellers in steady flow", Ph.D. thesis, M.I.T. Dept. of Ocean Engineering, Aug. 1987.

[7.5] Molland, A.F.,"The prediction of rudder-propeller interactions using blade element momentum and modified lifting line theory", Ship Science Report No. 54, University of Southampton, 1992.

CHAPTER EIGHT

[8.1] Molland, A.F. & Turnock, S.R.,"The prediction of ship rudder performance characteristics in the presence of a propeller", presented at the 2nd International conference on Manoeuvring and Control of Marine Craft, Southampton, U.K. 15th-17th July 1992.

[8.2] Kose, D, "On a new mathematical model of manoeuvring motions of a ship and its applications". International shipbuilding progress, Vol. 29, 1982.

[8.3] Matsumoto, N., and Suemitsu, K., "Interference effects between the hull, propeller and rudder of a hydrodynamic mathematical model in manoeuvring motion". Naval Architecture and Ocean Engineering, The Society of Naval Architects of Japan, Vol. 22, 1984.

[8.4] Norrbin, N.H., Ship turning circles and manoeuvring criteria. 14th STAR symposium". SNAME, 1989.

[8.5] Stierman, E.J., "The Influence of the Rudder on the Propulsive Performance of Ships - Part I", International Shipbuilding Progress, Vol. 36, No. 407(1989), pp. 303-334.

[8.6] Cho. J, & Williams, M., "Propeller-Wing Interaction using a Frequency Domain Panel Method", Journal of Aircraft, Vol. 27, No. 3, March 1990.

APPENDIX A PARAMETERS WHICH GOVERN RUDDER-PROPELLER INTERACTION

In discussing the parameters which govern the interaction it is convenient to group them into four categories. These groups of parameters can then be used to assess their affect on rudder sideforce (lift C_L) which is of primary importance in ship manoeuvring. Additional dependent variables to be considered are the centre of action of the rudder sideforce (CP_c, CP_s), rudder stall angle and, for ship resistance and propulsion, the total thrust of the rudder-propeller combination (propeller thrust - rudder drag).

The four groups of parameters are defined as follows.

i) Flow variables which control the magnitude of the forces developed. These include the time dependent quantities V (free-stream velocity) and n (propeller rate of revolution) and the properties of the fluid, density (ρ) and dynamic viscosity (μ). Also included is the yaw angle γ between the combination and the free-stream.

ii) Rudder geometric variables which determine how the flow passes over the rudder and hence the force developed. This is controlled by the rudder incidence α, span S, mean chord c, stock position X_1, thickness t, section shape, sweep and twist.

iii) Propeller geometric variables which control how the propeller imparts energy into the flow and generates thrust. This is determined by its diameter D, pitch P, boss diameter, sweep, pitch and thickness distributions, number of blades and blade area ratio.

iv) Relative position and size of the rudder and propeller. The two units can be separated longitudinally (X), laterally (Y) and vertically (Z). The relative size is defined as the coverage ξ and is equal to the proportion of the rudder span in way of the propeller race. This can be expresses as $\xi = \lambda$ D/S where λ is the fraction of propeller diameter impinging on the rudder.

The sideforce (and other dependent variables) can be expressed as a function of the following non-dimensional variables, where the section shape, twist distribution etc. of the rudder and propeller are assumed fixed:

$$C_L = f\{[\ J,\ Rn,\ \gamma\ ],[\ \alpha\ ,\ AR,\ t/c,x1/c\ ],[\ P/D\ ],[\ X/D,\ Y/D,\ Z/D\ \xi\ ]\} \qquad [A.1]$$

From the above it can be seen that for a particular rudder-propeller combination the parameters reduce to:

$$C_L = f\{[\ J,\ Rn,\ \gamma\ ],[\ \alpha\ ],[\ P/D\ ]\} \qquad [A.2]$$

APPENDIX B DESIGN PARAMETERS FOR COMMUNICATIONS HARNESS

The following information is defined as an Occam2 library and is linked to all other libraries, EXE's and PROGRAM's.

The two protocols used in the original harness were:
(1) *message* between the Guest or Host and the Encode or Decode processes in Harness:

```
PROTOCOL message
  CASE
    disp;INT::[]BYTE;BYTE
    OneDRArray;INT;INT::[]REAL32;BYTE
    TwoDRArray;INT;INT::[]REAL32;INT::[]REAL32;BYTE
    ThreeDRArray;INT;INT::[]REAL32;INT::[]REAL32;INT::[]REAL32;BYTE
    OneDR64Array;INT;INT;INT;INT;INT::[]REAL64;BYTE
    inty;INT;BYTE
    realy;REAL32;BYTE
    booly;BOOL;BYTE
:
```

(2) Each of the different tag's data were translated into a single one-dimensional array of BYTEs using the STRINGTOREAL, REALTOSTRING procedures. The first number of the array was an integer which indicated the *message* tag used. Attached to the front of this string was a 25 byte header with the following format, where the # character was used as a spacer between components of the string:

Bytes 1 to 5: Indicate type of message "Whoop" = general message, "Home#" = message for home etc.
Byte 6 = "#"
Byte 7 to 15: Is a padded integer with a value equal to the absolute identity number of the transputer from which the message originates e.g. "9########"
Byte 16 = #
BYTE 17 to 25: Is a padded integer with a value equal to absolute identity number of destination transputer

A single protocol *newmess* was used in the modified harness design, which was simply the original *message* with a new tag zippy added. This tag allows all the other message tags to be combined into a single stream of data. The first integer again indicates the message tag and the first string is the original header string.

```
PROTOCOL newmess
  CASE
    zippy;INT;INT::[]BYTE;INT::[]BYTE;INT::[]REAL32;INT::[]REAL64;INT::[]INT;INT::[]BOOL
    disp;INT::[]BYTE;BYTE
    OneDRArray;INT;INT::[]REAL32;BYTE
    TwoDRArray;INT;INT::[]REAL32;INT::[]REAL32;BYTE
    ThreeDRArray;INT;INT::[]REAL32;INT::[]REAL32;INT::[]REAL32;BYTE
    OneDR64Array;INT;INT;INT;INT;INT::[]REAL64;BYTE
    inty;INT;BYTE
    realy;REAL32;BYTE
    booly;BOOL;BYTE
:
```

The following protocol is used in the live-memory store described in Chapter 5.

```
PROTOCOL pan
  CASE
    finishpan
    setuppan;INT;INT;INT
    getnode;INT;INT
    getpotential;INT
    getpanel;INT
    getcentre;INT
    getTE;INT
```

```
    GetInfo
    GetNPak
    ProduceSurface
    outpotential;REAL32
    outcoord;INT::[]REAL32
    outpanel;INT::[]REAL32
    writeints;INT
    writecoord;INT;INT;INT::[]REAL32
:
```

This protocol is used for the live-memory store Matrix procedure.

```
PROTOCOL mate
  CASE
    finish
    GetPoint;INT;INT
    GetRow;INT
    GetColumn;INT
    FindRowMax;INT
    FindMax
    RowOperate;INT;INT;REAL32;REAL32;INT::[]REAL32
    RowSwop;INT;INT
    ColOperate;INT;INT;REAL32;REAL32;INT::[]REAL32
    ColSwop;INT;INT
    SetZero
    SetIdentity
    OutPoint;REAL32
    OutRC;INT::[]REAL32
    OutMax;INT;INT;REAL32
:

SystemConstants  --  Used to define the maximum size of the various components of the newmess protocol
VAL NH.header.size IS 25:
VAL NH.message.size IS 30:
VAL NH.NoReal32 IS 210: -- 110:  Fundamental constant used to define max. data array size per transputer
VAL NH.NoReal64 IS 50:
VAL NH.NoInt IS 32:
VAL NH.NoBool IS 10:
VAL NH.disp IS 1:
VAL NH.OneDRArray IS 2:
VAL NH.TwoDRArray IS 3:
VAL NH.ThreeDRArray IS 4:
VAL NH.OneDR64Array IS 5:
VAL NH.inty IS 6:
VAL NH.realy IS 7:
VAL NH.booly IS 8:
--}}}
--{{{  constants
VAL up IS 3:
VAL down IS 4:
VAL left IS 1:
VAL right IS 2:
VAL drain IS 10:
VAL MaxDim IS NH.NoReal32: -- 110
VAL MaxDimSqr IS (MaxDim*MaxDim):
```

```
VAL PanelSize IS 50:  -- 30:
VAL SplineSize IS 100:  -- 50:
VAL TotNoReal IS 7*SplineSize:
VAL MaxNoSections IS 30:
VAL MaxHarnMess IS NH.NoReal32:
VAL MaxNoBod IS 10:

--{{{  System Constants used for 2-D Euler solver
VAL g IS 1.4(REAL32):
VAL n IS 110:
VAL m IS (40*n):
VAL nm IS 240:
VAL normal IS 6:
VAL solid IS 1:
VAL inflow IS 2:
VAL outflow IS 3:
VAL free IS 4:
VAL internal IS 5:
--}}}
--}}}
```

APPENDIX C ELLIPTIC GRID GENERATION

Elliptic grid generation process is based on transforming the physical co-ordinates (x,y) of a mesh into values in a computational space $\xi\eta$ and then solving the Laplace equations:

$$\begin{aligned} \xi_{xx} + \xi_{yy} &= 0 \\ \eta_{xx} + \eta_{yy} &= 0 \end{aligned} \qquad \text{(C.1)}$$

with specified boundary conditions, where the subscripted variable denotes differentiation with respect to that variable. Interchanging dependent and independent variables gives

$$\xi_{xx} = \frac{1}{J^2}\left(y_\eta \frac{\partial y_\eta}{\partial \xi} - y_\xi \frac{\partial y_\eta}{\partial \eta} \right) \qquad \text{(C.2)}$$

where

$$J = x_\xi\, y_\eta - y_\xi\, x_\eta$$

and so on for $\xi_{yy}\ \eta_{xx}\ \eta_{yy}$

Expressing in terms of coefficients α, β, and γ

$$\begin{aligned} &\alpha\, x_{\xi\xi} - 2\beta\, x_{\xi\eta} + \gamma\, x_{\eta\eta} \\ &\alpha\, y_{\xi\xi} - 2\beta\, y_{\xi\eta} + \gamma\, y_{\eta\eta} \end{aligned} \qquad \text{(C.3)}$$

where

$$\begin{aligned} \alpha &= x_\eta^{\,2} + y_\eta^{\,2} \\ \beta &= x_\xi\, x_\eta + y_\xi\, y_\eta \\ \gamma &= x_\xi^{\,2} + y_\xi^{\,2} \end{aligned}$$

These analytic expressions are replaced by discrete equations with a uniform grid spacing $\Delta\xi = 1$ and $\Delta\eta = 1$. The coefficients are then replaced by a second order central difference:

$$\alpha = \left(\frac{x_{i,j+1} - x_{i,j-1}}{2} \right)^2 + \left(\frac{y_{i,j+1} - y_{i,j-1}}{2} \right)^2$$

and similar for β *and* γ (C.4)

$$x_{\xi\xi} = \left(x_{i+1,j} - 2\,x_{i,j} + x_{i-1,j} \right)$$

and likewise for other derivatives of x and y.

Substituting these into (C.5) and rearranging to get an expression for a new ordinate x at iterative step n+1 which can be solve using a successive over relaxation (SOR)scheme with acceleration parameter (1<A<2) the final equation for x is (where the detail inside the brackets has been removed for clarity):

$$x_{i,j}^{n+1} = (1 - A)\, x_{i,j}^{n} + \frac{A}{2\alpha^n + 2\gamma^n}\left[\alpha^n(\,) - \frac{\beta}{2}(\,) + \gamma^n(\,) \right] \qquad \text{(C.5)}$$

and similarly for y.

Dirichlet boundary conditions were used, i.e. the values of x and y remained fixed on the boundaries. The S.O.R. scheme is run until a given convergence criteria has been reached and this depends on the degree of grid refinement required.

APPENDIX D PANEL INFLUENCE COEFFICIENTS

(i) Introduction

Newman[4.20] gave the potential due to a distribution of sources or normal dipoles on a flat quadrilateral panel. The derivation given for the normal dipole potential is considered to be superior to that of Hess and Smith[4.8] and is valid for panel's whose panel density is of an arbitrary polynomial form. However, in this work only the constant strength panel case is used.

An efficient scheme for the calculation of the influence coefficient of a panel for an arbitrary field point uses a graded series of far-field approximations to reduce computational time. Newman also gave an arbitrary order multipole expressions for source and normal dipole potential. The scheme used in this work for choosing the relevant expression is that of Newman and is based on the ratio B of the distance between the panel centre and node R to the size of the largest diagonal of the panel. The value of B for the various approximations is:

(1)		B < 2.0	Exact expression.
(2)	2.00 ≤	B < 2.45	4^{th} Order Multipole.
(3)	2.45 ≤	B < 4.0	2^{nd} Order Multipole.
(4)	4.00 ≤	B	Point

In this work expressions are also needed for the source and normal dipole velocity influence of of a flat quadrilateral panels. These are used in the wake adaption scheme and the calculation of the interaction velocity field. The velocity influence were obtained by differentiation of the expressions for the source and dipole potential given by Newman.

(ii) Panel Geometry

Figure D.1 illustrates an arbitrary quadrilateral panel located in an overall cartesian coordinate system (x',y',z').

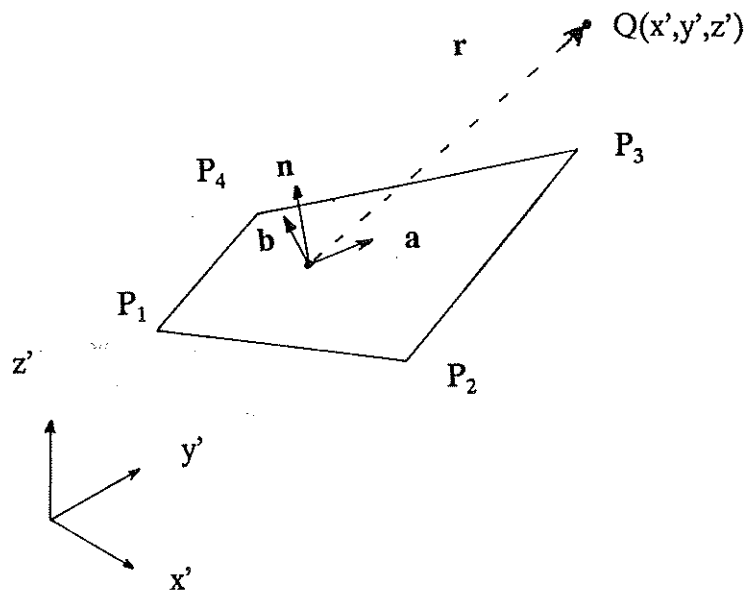


Figure D-1 Panel Geometry Schematic

Following the method of Hess and Smith[4.8], the coordinate centre **C** of the arbitrary panel is defined as the average position of the panel nodes P_i:

$$C = \frac{1}{N}\sum_{i=1}^{N} P_i \qquad (D.1)$$

A panel centroid coordinate system is defined using the panel diagonals $\mathbf{a} = \mathbf{P_3} - \mathbf{P_1}$ and $\mathbf{d} = \mathbf{P_4} - \mathbf{P_2}$. The unit normal **n** is then:

$$n = \frac{a \times d}{|a \times d|} \qquad (D.2)$$

and the panel area $A = \frac{1}{2} |\, \mathbf{a} \times \mathbf{d} \,|$. Once **n** is defined the third vector of the panel orthonormal system is found using the unit vector in the direction of **a**.

$$b = \frac{n \times a}{|a|} \qquad (D.3)$$

To obtain a flat panel, which is necessary for the analysis, the position of each node is adjusted so that they are all coplanar with the panel centre C using:

$$For\ i = 1\ to\ 4, \qquad P'_i = P_i + \left(\left(C - P_i \right) . n \right) n \tag{D.4}$$

The orthonormal system matrix [B] = (a,b,n) is then used to transform the panel nodes into the panel centred coordinates on the plane z=0.

$$P''_i = [B]\left(P_i - C \right) \tag{D.5}$$

A further translation is applied to locate the centre of the panel coordinate system at the centroid of the panel.

Figure D.2 illustrates the planar representation after the transformation to the panel coordinate system, with ξ, η define the position of the panel nodes and s the length between nodes.

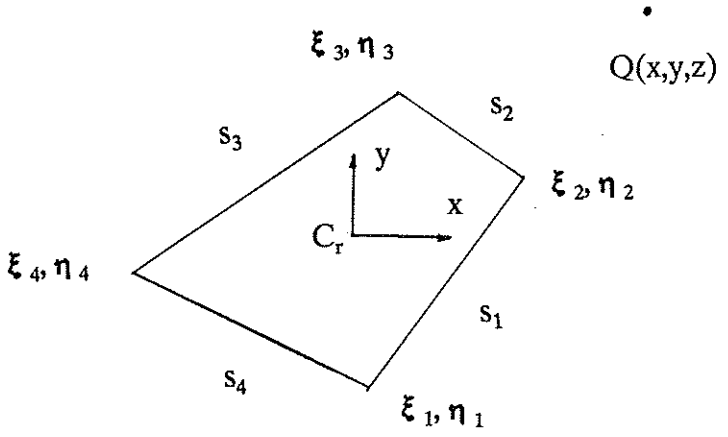


Figure D-2

To find the influence of the at a particular field point Q(x,y,z) the first step is to translate the point into the individual panel coordinate system:

$$Q' = [B]\left(Q - P_{cr} \right) \tag{D.6}$$

where P_{cr} is the panel centroid in the overall coordinate system. The radial distance from the centroid to Q is then simply the magnitude of Q'.

As the expression for velocity influence coefficient V is calculated within the panel coordinate system a final transformation using the transpose of [B] has to be applied to find the velocity influence in the overall system:

$$V^{o} = [\,B\,]^{T}\, V^{p} \tag{D.7}$$

(iii) Exact

Dipole potential

The potential at Q for a normal dipole distribution of constant strength -4π is an integral over the panel surface:

$$\Phi = z \iint [\,(x-\xi)^2 + (y-\eta)^2 + z^2\,]^{-\frac{3}{2}}\, d\xi\, d\eta \tag{D.8}$$

Newman's derived expression is simply the sum of four arctangents for a quadrilateral panel, with coefficients based on the geometric properties of the sides and does not require any numerical integration and is:

$$\Phi = \sum_{n=1}^{4} \tan^{-1}\left(\frac{S_{3n}}{C_{3n}} \right)$$

where $S_3 = S_1C_2 - S_2C_1$ and $C_3 = C_1C_2 + S_1S_2$.

The four geometric coefficients are:

$$S_1 = \delta\eta_n\, [\,(x-\xi_n)^2 + z^2\,] - \delta\xi_n\, (x-\xi_n)\, (y-\eta_n) \tag{D.10}$$

$$S_2 = \delta\eta_n\, [\,(x-\xi_{n+1})^2 + z^2\,] - \delta\xi_n\, (x-\xi_{n+1})\, (y-\eta_{n+1}) \tag{D.11}$$

$$C_1 = R_n\, z\, \delta\xi_n \tag{D.12}$$

$$C_2 = R_{n+1}\, z\, \delta\xi_n \tag{D.13}$$

where R_n is the radial distance between Q and node n, $\delta\eta_n = \eta_{n+1} - \eta_n$, and $\delta\xi_n = \xi_{n+1} - \xi_n$.

Dipole velocity

The velocity influence coefficient of a constant normal dipole distribution of strength -4π at field point Q is:

$$V = -\nabla\Phi \tag{D.14}$$

Differentiating the expression for dipole potential influence gives:

$$V_d = -\sum_{n=1}^{4} \frac{1}{S_3^2 + C_3^2} \left((C_1C_2^2 + S_2^2C_1)\nabla S_1 + (S_1^2S_2 + S_2C_1^2)\nabla C_2 - (C_1^2C_2 + S_1^2C_2)\nabla S_2 - (S_1C_2^2 + S_1S_2^2)\nabla C_1 \right) \quad \text{(D.15)}$$

where

$$\nabla S_1 = \begin{bmatrix} \delta\eta_n (x-\xi_n) - \delta\xi_n (y-\eta_n) \\ -\delta\xi_n (x-\xi_n) \\ \delta\eta_n z \end{bmatrix} \quad \text{(D.16)}$$

$$\nabla S_2 = \begin{bmatrix} \delta\eta_{n+1} (x-\xi_{n+1}) - \delta\xi_n (y-\eta_{n+1}) \\ -\delta\xi_n (x-\xi_{n+1}) \\ \delta\eta_n z \end{bmatrix} \quad \text{(D.17)}$$

$$\nabla C_1 = \begin{bmatrix} \frac{z(x-\xi_n)\delta\xi_n}{R_n} \\ \frac{z(y-\eta_n)\delta\xi_n}{R_n} \\ R_n\delta\xi_n + \frac{z^2\delta\xi_n}{R_n} \end{bmatrix} \quad \text{(D.18)}$$

$$\nabla C_2 = \begin{bmatrix} \frac{z(x-\xi_n)\delta\xi_n}{R_{n+1}} \\ \frac{z(y-\eta_n)\delta\xi_n}{R_{n+1}} \\ R_{n+1}\delta\xi_n + \frac{z^2\delta\xi_n}{R_{n+1}} \end{bmatrix} \quad \text{(D.19)}$$

In the plane z = 0 the expression for V_d reduces to:

$$V_d = -\sum_{n=1}^{4} \begin{bmatrix} 0 \\ 0 \\ \frac{\delta\xi_n (S_1R_{n+1} - S_2R_n)}{S_1S_2} \end{bmatrix} \quad \text{(D.20)}$$

Source potential

The potential at field point Q of a constant source distribution of strength -4π is a surface integral:

$$\Psi = \iint \frac{\delta\xi \, \delta\eta}{r} \tag{D.21}$$

as $\Phi = -d\Psi/dz$ and since Φ and Ψ vanish at infinity partial integration gives that:

$$\Psi = \int_z^\infty \Phi \, dz = -\int_z^\infty z \, d\Phi - z\,\Phi \tag{D.22}$$

Newman evaluated the integral terms to find the expression for Ψ as:

$$\Psi = \sum_{n=1}^{4} v_n Q_n - z\,\Phi \tag{D.23}$$

where

$$v_n = (x - \xi_n)\sin(\theta_n) - (y - \eta_n)\cos(\theta_n) \tag{D.24}$$

$$Q_n = \log\left|\frac{R_n + R_{n+1} + s_n}{R_n + R_{n+1} - s_n}\right| \tag{D.25}$$

and for the angle θ_n

$$\begin{aligned} \sin\theta_n &= \frac{\delta\eta_n}{s_n} \\ \cos\theta_n &= \frac{\delta\xi_n}{s_n} \end{aligned} \tag{D.26}$$

Source velocity

The velocity influence of a constant source distribution of strength -4π at field point Q is:

$$V_s = -\nabla\Psi \tag{D.27}$$

but

$$\begin{aligned} \nabla\Psi &= \nabla\left(\sum_{n=1}^{4} v_n Q_n - z\Phi\right) \\ &= \sum_{n=1}^{4} \nabla v_n Q_n - \Phi\, n - z V_d \end{aligned} \tag{D.28}$$

The terms of Φ_n and zV_d are already known, after manipulation $\nabla v_n Q_n$ can be expressed as:

$$\nabla v_n Q_n = Q_n \begin{bmatrix} \sin\theta_n \\ \cos\theta_n \\ 0 \end{bmatrix} - \nabla\left(R_n + R_{n+1}\right) \frac{2s_n}{\left(R_n + R_{n+1}\right)^2 - s_n^2}\left[(x-\xi_n)\sin\theta_n - (y-\eta_n)\cos\theta_n\right] \tag{D.29}$$

but

$$\nabla\left(R_n + R_{n+1}\right) = \left(\frac{1}{R_n} + \frac{1}{R_{n+1}}\right)\left[\begin{bmatrix} x \\ y \\ z \end{bmatrix} - \begin{bmatrix} \frac{\xi_n}{R_n} + \frac{\xi_{n+1}}{R_{n+1}} \\ \frac{\eta_n}{R_n} + \frac{\eta_{n+1}}{R_{n+1}} \\ 0 \end{bmatrix}\right] \tag{D.30}$$

(iv) Multipole

An appropriate far-field expression for the source Ψ can be found by expanding the integral expression for Ψ as a Taylor series:

$$\Psi = \iint \frac{d\xi\, d\eta}{r} = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{(-)^{m+n}}{m!n!} I_{mn} \frac{\partial^{m+n}}{\partial x^m \partial y^n} \frac{1}{\sqrt{x^2 + y^2 + z^2}} \tag{D.31}$$

where I_{mn} is the moment of the panel about the origin:

$$I_{mn} = \iint \xi^m \eta^n d\xi\, d\eta \tag{D.32}$$

Similarly for a normal dipole distribution:

$$\Phi = \iint \frac{z\, d\xi\, d\eta}{r^3} = \sum_{m=0}^{\infty}\sum_{n=0}^{\infty} \frac{(-)^{m+n}}{m!n!} I_{mn} \frac{\partial^{m+n+1}}{\partial z \partial x^m \partial y^n} \frac{1}{\sqrt{x^2 + y^2 + z^2}} \tag{D.33}$$

The recursive relationships defined in Newman were used to calculate up to the 4^{th} order moments and these 13 values were calculated during the process of setting up a panel.

The various order partial differentials of the radius vector **r** were obtained as were the expressions for calculating the multipole expressions for V_s and V_d:

$$\nabla \frac{\partial^{m+n}}{\partial x^m y^n}\left(\frac{1}{r}\right) \tag{D.34}$$

and

$$\nabla \frac{\partial^{m+n+1}}{\partial z \partial x^m y^n}\left(\frac{1}{r}\right) \tag{D.35}$$

These expressions are not given here as their evaluation is straightforward if tedious.

(v) Point

The zero order multipole expression gives the influence coefficient of the panel as a point with the panel area (I_{00}). That is:

$$\begin{aligned} \Psi &= \frac{A}{r} \\ \Phi &= \frac{zA}{r^3} \end{aligned} \tag{D.36}$$

and

$$V_s = -\nabla\Psi = \frac{A}{r^3} r \tag{D.37}$$

$$V_d = -\nabla\Phi = \frac{3Az}{r^5} r - \frac{A}{r^3}\begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \tag{D.38}$$

(vi) Implementation

Two procedures were written in Occam2:

(1) SetUpSourcePanel, calculates all the geometric parameters and moments for a quadrilateral panel;

(2) NewmanPanel, evaluates the dipole potential and source potential, or the dipole velocity and source velocity influence coefficients.

These procedures were written using 3-component vector processes so that their function is transparent.

APPENDIX E INDEX PROCEDURE FOR PANEL STORE

The co-ordinate data in the Panel store and other live-memory store derivatives is stored in the manner, best illustrated using a series of nested SEQ loops, where index is a variable indicating the position from the zeroth element of a one-dimensional array of REAL32 numbers, and Panel number is the absolute number defining a particular set of four ordered co-ordinates.

```
TotNoPanperT := N / T          -- where N is the total number of panels and T the total number
                               -- of Guest transputers

Mstart := (TotNoPanPerT * ( PNo - 1 ) )        -- where PNo is the absolute guest transputer
                                               --                                    - -
number between 1 and T
Body[0]:= 0

Pstart[0]:= Mstart

SEQ i = 0 FOR (NB + NLB)  --  Number of bodies NB, and number of lifting bodies NLB

        Body[i+1] := 3 * (Nt[i] * Ns [i]) + Body[i]

        Pstart[i+1] := Pstart[i] + ( (Nt[i]-1) * (Ns[i]-1) )

        SEQ j = 0 FOR Nt[i] -- Number of nodes in t direction for body i

                SEQ k = 0 FOR Ns[i] -- Number of nodes in s direction for body i

                        X coordinate stored at index:= Body[i] + 3 * ( (j * Ns[i]) + k )
                        Y coordinate stored at index:= Body[i] + 3 * ( (j * Ns[i]) + k ) + 1
                        Z coordinate stored at index:= Body[i] + 3 * ( (j * Ns[i]) + k ) + 1

                        PanelNo: = Pstart[i] + ( ( j * Ns[i] ) + k )
```

To find the four vertices of a given absolute PanelNo the following process is carried out:

```
RelativePanelNo := PanelNo - Mstart

BodNo := 0
WHILE RelativePanelNo < Pstart[BodNo+1]
        BodNo := BodNo + 1

RelativeBodyPanelNo := RelativePanelNo - Pstart[BodNo]

j value (t direction) := RelativeBodyPanelNo / ( Ns[BodNo] - 1 )
k value (s direction) := RelativeBodyPanelNo - ( j * (Ns[BodNo] - 1))
```

Panel Node **P1** is j,k
Panel Node **P2** is j+1,k
Panel Node **P3** is j+1,k+1
Panel Node **P4** is j,k+1

The outward facing panel normal **n** is defined as:

$$\mathbf{n} = (\mathbf{P3} - \mathbf{P1}) \times (\mathbf{P4} - \mathbf{P2}) / |(\mathbf{P3} - \mathbf{P1})| |(\mathbf{P4} - \mathbf{P2})|$$

APPENDIX F GEOMETRY INPUT FILE

A standard ASCII format text file is used.

The first line defines the number of non-lifting bodies, lifting bodies, and number of free panels. Then, if a plane of symmetry is to be used or axisymmetric body the necessary axes, origins and rates of

rotation are defined.

The velocity field is specified as a uniform distribution of points Nx, Ny, and Nz in the overall cartesian or cylindrical axes system directions. At each point a scaled velocity vector is given relative to a defined uniform inflow velocity.

The definition of the non-lifting and lifting surface bodies is carried out in order. For each body the pivot, offset, scale, and angle of its own body centred axis system is given as described in chapter four. Each body is defined by a variable number of sections, and each section consists of a variable number of points. Also given are the number of spanwise and chordwise panels for the body. Two parameters define the type of panel size distribution on the body. Example distributions allow clustering of panels at body leading and trailing edges using various forms of sinusoidal function.

For the lifting surface bodies additional information is given to describe the trailing edge wake sheet in a manner similar to that of the body itself. An example lifting surface body definition, with comments, is given below for Rudder No. 2. Additional comments denoted (i) are given at end.

```
  0(NB)   1(NLB)   0(NFP)          No. of Bodies,  No. of Lifting Bodies,  Dummy
  0.0   0.0   0.0        Origin of reflection plane   x , y, z
  0.0   0.0   1.0        Unit normal to reflection plane  nx, ny, nz
  0.0   0.0   1.0        dummy nx,ny,nz
  0.0   0.0   0.0        Origin of axis of rotation   x , y, z
  0.0   0.0   0.0        Axis of rotation, magnitude is rate of rotation in rad/sec
  10.0                   scale velocity m/s
   1      1  1           velocity cube no of points in x,y,z direction
   0.0    0.0 0.0        origin of velocity cube  Vxo,Vyo,Vzo
   0.0    0.0 0.0        increments in Vdx, Vdy ,Vdz
(i) 1.0    0.0  0.0      non-dimensional velocity vx,vy,vz, data for (NVx.NVy.NVz) sets
(ii)  26  17  3  -1       Nt, Ns, NS, Image
(iii)  25  10             No of wake panels, No of free panels  { this line is not used for a body}
(iv)  4   0   1  0       t distribution, s distribution, t close, s open
  30.0   0.0   0.0       Pivot Vector, x, y, z
  0.0   0.0   0.0        Offset Vector, x, y, z
  0.00667   0.0667  1.0  Scale Vector, x, y, z
  0.0   0.0  -0.4         Ordered rotation: angle about x, about y, about z axis ( all in degrees)
(v)  35                   NPS No. of points in each section
  100.0  0.0    0.0      start at trailing edge  x, y, z
  95.0   0.134  0.0
  90.0   0.241  0.0
  80.0   0.437  0.0
  ....   .....  ...
  5.0    0.592  0.0
  2.5    0.436  0.0
  1.25   0.316  0.0
  0.0    0.0    0.0      around leading edge
  1.25  -0.316  0.0
  2.5   -0.436  0.0
  5.0   -0.592  0.0
  7.5   -0.7    0.0
  ...    ....   ....
 95.0   -0.134   0.0
 100.0   0.0     0.0      and finish at trailing edge x,y,z
 35                       repeat for next two sections
 100.0   0.0     0.5      start of middle section
 95.0    0.134   0.5
 ....    .....   ....
```

```
 90.0   -0.241    0.5
 95.0   -0.134    0.5
 100.0   0.0      0.5
 35
 100.0   0.0      1.0        start of top section
 95.0    0.134    1.0
 ....    .....    ....
 70.0   -0.611    1.0
 80.0   -0.437    1.0
 90.0   -0.241    1.0
 95.0   -0.134    1.0
 100.0   0.0      1.0
(vi)  4                      no of points in each of three wake sections
  100.0  0.0      0.0
  125.0  0.0      0.0
  150.0  0.0      0.0
  250.0  0.0      0.0
  4
  100.0  0.0      0.5
  125.0  0.0      0.5
  150.0  0.0      0.5
  250.0  0.0      0.5
  4
  100.0  0.0      1.0
  125.0  0.0      1.0
  150.0  0.0      1.0
  250.0  0.0      1.0
```

(i) Velocity field information is ordered in the manner shown below

SEQ i=0 For NVx
SEQ j= 0 For NVy
SEQ k=0 For NVz
$Vx_{i,j,k}$, $Vy_{i,j,k}$, $Vz_{i,j,k}$

and the co-ordinates (using the same indices) as: x:= Vxo + (i*Vdx)
y:= Vyo + (j*Vdy)
z:= Vzo + (k*Vdz)

A cylindrical velocity field is designated by the z component of the index increment having a value greater than 10000. The origin of the axis is taken to be the defined origin, and the unit vector in the axis direction **A** is defined thus: **Ax** := Vdz - 10000
Ay := $Vy_{0,0,0}$
Az := $Vz_{0,0,0}$

Vdx is the increment in the axial direction, and Vdy the increment in radius dr. NVz is always equal to one, and Vy corresponds to the radial velocity Vr (positive outward) and Vz the velocity on the circumferential direction Vw. The direction of Vw is determined by the cross-product **A x (0,0,1)**.

(ii) Nt = No of nodes in t (usually chordwise direction), number of panels is Nt-1,

Ns = No nodes in s (usually spanwise direction), if the total number of transputers is greater than 1 has to be a multiple of the total number of transputers +1

NS = No of sections of data used to define body, limit on TOTAL number of sections to define a body or multiple bodies is 30, a minimum of 3 sections per body.

Image = 0 { no reflection plane} , = -1 { reflection plane located at defined origin and with defined outward facing normal }, = +N { no of degrees of rotational symmetric bodies, 1 is just defined body, >1 is 1 body + N-1 images at a spacing of 360°/N degrees, origin and rotational axis of symmetry are given at start of file }.

(iii) First integer is total number of nodes to be used to panellise wake sheet strip, of which there will be (Ns-1), second integer is the number of wake panels which are allowed to move in wake adaption process.

(iv) First integer is the type of panel size distribution in the t direction and the second integer the distribution in the s direction. At present the allowed mapping function to give the distribution are:

0 Equal sized panels of Δ
1 Sinusoid clustering as parameter tends to unity
2 Sinusoid clustering as parameter tends to zero and unity
3 Sinusoid clustering as parameter tends to zero, half and unity
4 Sinusoid clustering as parameter tends to half
5 Sinusoid clustering as parameter tends to zero
6 Modified sinusoid clustering as parameter tends to zero with minimum size of 0.25Δ
7 Modified sinusoid clustering as parameter tends to zero with minimum size of 0.5Δ

The third integer is set to 1 if the t direction is a closed shape e.g. aerofoil, or 0 if open. The fourth integer states whether the s distribution is open or closed.

(v) The number points defining each of the NS sections is variable between 3 and 100.

(vi) The number of wake sections is equal to the number of lifting body sections, with points defined from the trailing edge direction.

APPENDIX G ERROR ASSOCIATED WITH APPROXIMATE PROPELLER DEFINITION

The exact cylindrical definition of a propeller section lies along a helix of radius r and pitch P, as shown in Figure G.1 (following Dr. J.F. Wellicome).

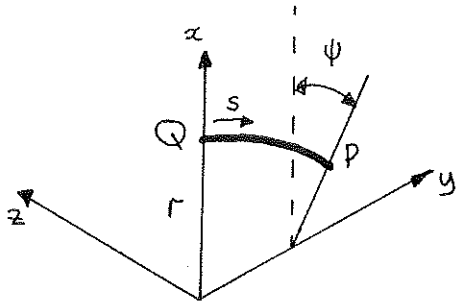


Figure G-1 Helix Definition

For a propeller with no sweep, the x axis is coincident with the propeller generator. A helix of radius r which cuts the generator at y=0 and x=r has the equation:

$$A = \left\{ r\cos\psi \ , \ \frac{P\psi}{2\pi} \ , -r\sin\psi \right\} \qquad [G.1]$$

where **A** is a position along the helix defined in terms of ψ. The arc-length s from the generator is given by:

$$s = \sqrt{r^2 + \left(\frac{P}{2\pi} \right)^2}\, \psi \qquad [G.2]$$

The expanded section lies on a cylindrical surface laid along the helix with the thickness perpendicular to this and defined in terms of a unit vector thus:

$$\mathbf{t} = \frac{1}{\sqrt{1 + \left(\frac{2\pi r}{P} \right)}} \left\{ \sin\psi \;,\; \frac{2\pi r}{P} \;,\; \cos\psi \right\} \qquad [G.3]$$

If the substitution that:

$$\tan\phi = \frac{2\pi r}{P} \qquad [G.4]$$

is made then the location of a point on the propeller surface of thickness t in cartesian right-handed set is given by:

$$\begin{bmatrix} r\cos\psi + t\cos\phi\ \sin\psi \\ \frac{P\psi}{2\pi} + t\sin\phi \\ -r\sin\psi + t\cos\phi\ \cos\psi \end{bmatrix} \qquad [G.5]$$

The approximate expression given in Chapter 6 locates the point at:

$$\begin{bmatrix} r\cos\phi \\ r\sin\theta\ \cos\phi + t\sin\phi \\ -r\sin\theta\ \sin\phi + t\cos\phi \end{bmatrix} \qquad [G.6]$$

The error in position definition $\boldsymbol{\Delta}$ is then found by subtracting [G.6] from [G.5].

$$\begin{bmatrix} r(\cos\phi - \cos\psi) + t\cos\phi\ \sin\psi \\ r\cos\phi\,(\theta - \sin\theta) \\ r(\sin\theta\ \sin\phi - \sin\psi) + t\cos\phi\ \cos\psi \end{bmatrix} \qquad [G.7]$$

where the identity:

$$\theta\ \sin\phi = \psi \quad as \quad s = r\theta \quad and \quad s = r\psi \sqrt{1 + \left(\frac{P}{2\pi} \right)^2} \qquad [G.8]$$

is used, so that:

$$\frac{P\psi}{2\pi} = \frac{P\theta \sin\phi}{2\pi} = \frac{P\theta\ r \sin\phi}{2\pi\ r} = r\theta \cos\phi \qquad [G.9]$$

It can be seen that Δ reduces to zero on the generator. The error in axial position is proportional to θ -sinθ which is small for even moderate values of θ. The net effect is a slight change in effective pitch at every radius much the same to operating a propeller off-design pitch. Overall, this results in the approximate propeller shape having a different pitch from the nominal design pitch ratio setting. The propeller used in this work was designed with a pitch ratio setting of 1.0 and has an actual value of 0.95. This conclusion is confirmed by the good comparison between the Wageningen data and the propeller used (Fig. 6.12).