CONTROL SOFTWARE FOR TWO DIMENSIONAL AIRFOIL TESTS USING A SELF-STREAKLINING FLEXIBLE WALLED TRANSONIC TEST SECTION

S. W. D. Wolf

UNIVERSITY OF SOUTHAMPTON
Southampton, England

Grant NSG-7172
July 1982

LIBRARY COPY
JUL 1 2 1982

LANGLEY RESEARCH CENTER
LIBRARY, NASA
HAMPTON, VIRGINIA

NASA
National Aeronautics and Space Administration
Langley Research Center
Hampton, Virginia 23665
CONTENTS

1. Introduction

2. Control System Hardware Outline

3. Control Software
   3.1. Program TSWT
   3.2. Subroutine DATA
   3.3. Subroutine REDUCE
   3.4. Subroutine WAS
   3.5. Subroutine STAR
   3.6. Subroutine SUME
   3.7. Subroutine FORCE
   3.8. Subroutine SET
   3.9. Subroutine WALL
   3.10. Program REAN
   3.11. Program RUN

4. Symbols

5. References

Figures

Appendix A
INTRODUCTION

The operating procedure of a self-streamlining wind tunnel has already been discussed and is summarised in the flow diagram shown on Figure 1. The iterative nature of the streamlining process, requiring numerous measurements and calculations coupled with the need for a rapid and continual exchange of data between wind tunnel and computer, makes mandatory the use of a computer.

The current operation of the Transonic Self-Streamlining Wind Tunnel (TSWT) involves on-line data acquisition with automatic wall adjustment. A tunnel run consists of streamlining the walls from known starting contours in iterative steps and acquiring model data. Each run performs what is described as a streamlining cycle. The associated control software is presented here.

Development of the on-line control system for TSWT continues. It is anticipated that modifications to the existing software package will consist of minor changes to some of the subroutines.

The introduction of further subroutines for

(a) Calculation of model forces from the 'wall data' (wall pressures and positions).

(b) Solution of mixed imaginary flowfields with a shock present.

(c) Prediction of wall shapes, to minimise boundary interference on a three dimensional model in a two dimensional test section.

(d) Assessment of residual boundary interferences on a three dimensional model in a two dimensional test section.

is planned as they become available. The modular architecture of the software will allow these additions to be made easily.

The software package has been developed for simple application to other self-streamlining wind tunnels. The modular architecture allows individual control system subroutines to be utilised with existing users software. Also the software package is written in a general manner to minimise new application modification.

During TSWT development numerous programmes have been written to check sections of the control software. A number of these remain in use to assist with TSWT operation as follows:
i) Set both walls to known contours together or individually.

ii) Allow operator modification of known wall contours.

iii) Display current position of both walls.

iv) Display and/or load contents of any specified data file record.

In addition programs have been written to command a Tektronix 4662 plotter to display model pressure distributions, flexible wall Mach number distributions and wall shapes.
CONTROL SYSTEM HARDWARE OUTLINE

An overview of the current control system is shown in Figure 2. Macroscopically, the system involves the interaction of the tunnel operator with the wind tunnel and computer to generate the required test data.

The on-line control system has two distinct functions:

1) to streamline the flexible walls which includes assessment of residual boundary interferences.

2) to acquire and reduce test data from a model.

These functions are achieved by two control loops between wind tunnel and computer linked to processing software for data manipulation. The control loops are for wall shape control and pressure data acquisition (Scanivalve control). Data acquisition is automatic since the Scanivalve is stepped by computer commands and analogue pressure data is fed direct to the computer.

The control system hardware has evolved about these two control loops using both analogue and digital data transfer. Microscopically, the control system becomes complex as shown in Figure 3.

The system hardware consists of the computer and its peripherals communicating with a series of control and signal conditioning sub-systems housed in a control cabinet by the wind tunnel. This cabinet is then connected to the test section wall position sensors (linear potentiometers) and wall jack stepper motors.

The hardware will perform four functions:

1) wall movement.

2) wall and model pressure measurements.

3) wall position sensing.

4) system monitoring (not yet existing).

The wall movement function involves the loading of forty motor latch boards with direction information (stop, forward-go or reverse-go). Then the sending of a 'go' pulse to the Pulse Sequence Generator (PSG), starts actual wall movement. The PSG generates control pulses at a fixed frequency to the forty Motor Drive Boards (MDBs). The MDB, using signals from the motor latch board and the power
supply, controls the sequence of power pulses transmitted to the 3-phase stepper motor powering each wall jack. After a pre-determined and variable time interval the PSG is switched off and wall movement ceases. The wall movement control sequence is then repeated until satisfactory wall contours are achieved.

The wall and model pressure measurements involve the driving of the scanivalve system by step pulses, to a required sampling port. Then the analogue signals from the pressure transducers are sampled by the computer after a suitable settling time (i.e. 50 milli-seconds).

The wall position sensing function involves the sampling of forty analogue signals from the linear potentiometers attached to each wall jack mechanism. These signals are continuously available at the computer peripherals, but they have been found to be susceptible to electronic interference when the jack motors are switched on. Hence the wall position is only sampled with the walls stationary.

The system monitor is a necessary part of a practical digital control system. When this hardware becomes available it will provide information on the status of hardware components, to allow quick error diagnostics during tunnel operations.

The conditioned analogue signals from the test section (within the range ±5 volt) are fed via the control cabinet to a DEC AD11-K Module for 12-bit analogue to digital conversion. This module is combined with a DEC AM11-K Multiplexer to make 64 channels available for input signals.

The digital signals, at 0 and 5 volts, are transmitted to the wind tunnel in a code described in Appendix A. Device selection is by means of a 'telephone exchange' called the "address decoder". In fact, commands to the test section are sent to all devices but only one device is enabled to read the information, by address decoder selection. The versatility of a digital control system is well known and the reduction in interface wiring compared with an analogue system is significant. The address decoder has the capability of addressing 64 devices.

The control system hardware layout in Figure 3 has been simplified for clarity. In practice there are numerous synchronisation signal paths between devices, to ensure correct operation sequencing and to prevent 'race' problems.

The tunnel operator monitors the control system from a computer VDU consol and inputs test parameters. The consol allows the real time display of test section and model data which is stored on the computer mass storage device. This data can subsequently be drawn in graphical form on an XY Plotter (Tektronix 4662).
Hard copy of summarised run data is printed in real time on a line printer. The control of a model wake traverse mechanism has been incorporated in the system, utilising the wall movement technique already described.

The control system hardware has excess capability with 11 spare digital address slots and 15 spare analogue input channels. This spare capacity may be used for the control of tunnel Mach number and model attitude in the future.
3. **CONTROL SOFTWARE**

Computer software has been developed for the on-line TSWT control system using a versatile modular architecture. Hence the program has been reduced to a collection of manageable subprograms which can be combined to control the wind tunnel and output real time results or provide more detailed off-line reanalysis of previously acquired data.

An overview of the control software package is shown below.

<table>
<thead>
<tr>
<th>File Type &amp; Name</th>
<th>File Storage Name</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Main Program (TSWT)</td>
<td>OFLEX</td>
<td>i) Control and sequence subroutine calls.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) Read test parameters from the operator.</td>
</tr>
<tr>
<td>Subroutine 1 (DATA)</td>
<td>OAD</td>
<td>Acquire pressure data from the wind tunnel.</td>
</tr>
<tr>
<td>Subroutine 2 (REDUCE)</td>
<td>ODR</td>
<td>Read tunnel data from disc storage and reduce raw pressure data from the wind tunnel.</td>
</tr>
<tr>
<td>Subroutine 3 (WAS)</td>
<td>OJUDD</td>
<td>Perform wall setting calculations.</td>
</tr>
<tr>
<td>Subroutine 4 (STAR)</td>
<td>ODST</td>
<td>Calculate local boundary layer displacement thickness and Mach number along each wall.</td>
</tr>
<tr>
<td>Subroutine 5 (SUME)</td>
<td>OERR</td>
<td>Assess wall induced interferences at the model.</td>
</tr>
<tr>
<td>Subroutine 6 (FORCE)</td>
<td>{ OURING</td>
<td>Calculation of model forces for NACA 0012-64 and NPL 9510 sections respectively.</td>
</tr>
<tr>
<td></td>
<td>{ ONPL</td>
<td></td>
</tr>
<tr>
<td>Subroutine 7 (SET)</td>
<td>OUT</td>
<td>i) Store run data on disc</td>
</tr>
<tr>
<td></td>
<td></td>
<td>ii) Output data to the terminal and/or the plotter.</td>
</tr>
<tr>
<td>Subroutine 8 (WALL)</td>
<td>OADJ</td>
<td>Move the walls to new contours</td>
</tr>
</tbody>
</table>
This breakdown of the software into modules has been extremely useful, particularly for storage, editing and debugging purposes.

The software written in FORTRAN IV language is run on a DEC 11/34 with a DEC RT-11 V4 operating system. The software is linked to a system library and a FORTRAN library to access functions and system subroutines and a Real Time System Library (RTSL) to access peripheral control subroutines. The complete compiled and linked program requires over 100 blocks (25.6K words) of memory spare.

Current 16-bit computer processors are only capable of addressing 32K words (64K bytes) of real memory space. But of this, only 22K words is available for a user's program, depending on the size of the operating system. Therefore to run the TSWT control software on a 16-bit machine a technique of overlaying has to be used, so that only part of the software is stored in the real memory at any instant during execution.

Each subroutine is a self contained program communicating with the main program via common data blocks. So in theory only one subroutine is required in the real memory at any one time for execution. In practice, the subroutines have been grouped together to minimise the number of overlays, thereby reducing the time required for overlaying. The overlaying structure of the control software is shown below

<table>
<thead>
<tr>
<th>SEGMENT 1</th>
<th>SEGMENT 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>OVERLAY Subroutine 1</td>
<td>Subroutine 5</td>
</tr>
<tr>
<td>Subroutine 2</td>
<td>Subroutine 6</td>
</tr>
<tr>
<td>Subroutine 3</td>
<td>Subroutine 7</td>
</tr>
<tr>
<td>7762 words</td>
<td>Subroutine 8</td>
</tr>
<tr>
<td>7762 words</td>
<td>6558 words</td>
</tr>
<tr>
<td>ROOT SEGMENT Main Program + System Library</td>
<td>FORTRAN Library RTSL Library</td>
</tr>
</tbody>
</table>

This program structure is implemented at 'link' time during the program generation cycle as below
These commands generated a runable program called OFLEX. The control software memory requirement was reduced from 27k words to 17.3k words.

At 'run' time the program OFLEX requires four data files to exist on the computer mass storage device. Data file ADC.DAT receives the raw analogue-to-digital counts of the 'wall and model data' for each streamlining iteration; PAD.DAT provides and receives sets of wall contours and the associated external imaginary wall velocity distributions; NPL.DAT or WING.DAT receives Cps from the NPL 9510 and NACA 0012-64 models respectively, for each streamlining iteration; TWST.DAT holds all fixed tunnel data, i.e. jack positions, potentiometer calibrations, scaling and coupling factors, matrix coefficients for camber interference assessment and boundary layer information. RUN.DAT holds run data, i.e. ambient temperature and pressure, run number, iteration record number, and number of model tappings. The data files ADC.DAT, PAD.DAT and NPL.DAT/WING.DAT each hold 50 records: Records 1 to 3 in PAD.DAT hold data on the three aerodynamically straight contours (i.e. position and boundary layer thickness). Records 1 and 2 of NPL.DAT and WING.DAT files are used to store model tapping X and Y coordinates relative to the model leading edge. Records 4 to 50 are available to store data from each streamlining iteration. Hence iteration record numbers range from 4 to 50. When the iteration record number equals 50, ADC.DAT, PAD.DAT and NPL.DAT/WING.DAT must be copied to a data bank since the original data is then overwritten by subsequent iterations, starting with record 4. The upper limit on the iteration record number has been chosen to keep the data files in manageable proportions (i.e. 25.6k words maximum size). The total storage requirement of data files to run OFLEX in 61.7k words.

The data file RUN.DAT must be loaded with current run data before the control software is activated. This operation is performed by running a program OSTART, the software of which, called RUN, is described in section 3.11.

A complete listing of the control software is described in the following sections. Where possible standard FORTRAN has been used but peripheral control commands are peculiar to the DEC system used. These subroutine calls can be grouped into Analogue to Digital sampling commands (ADC and RTS), and programmable clock commands (SETR and LWAIT). In addition there are calls to the system library routines (IPEEK and IPOKE) for digital input and output operations.
An example of the minimal printout from OFLEX associated with a typical streamlining cycle, involving two wall adjustments, is shown on Figure 5. The walls were initially set to contours stored in record 30 of file PAD.DAT and the run finished with wall contours as stored in record 32. The model had no pressure tappings and therefore no wing performance data was presented in the print-out. Note that the print-out from OFLEX is sent to a line printer (logical Unit 7) while operator information is sent to a VDU Consol (Logical Unit 5). The operator information consists of prompts to indicate the stages reached in the streamlining cycle and error warnings.

The versatility of the software has allowed simple generation of programs for particular tasks such as tunnel data re-analysis. Using the existing subroutines as building blocks, a new program has been made up of a series of these subroutines linked to a new main program. For example, data re-analysis is achieved by running the program ORLEX. The main program REAN is a modification of TSWT with different subroutine calls and an extended print-out demanded as described in section 3.10. The program structure is very similar to that of the control software and is generated using the following link command with a memory requirement of 17.1k words.

```
R LINK
*ORLEX = OREF, FORLIB, RTSL/C
*OAD, ODR, OJUDD/O:1/C
*OERR, ODST, OUT/O:1/C
```

where OREF is the file storage name of program REAN.

An example of the extended printout is shown on Figure 6 for a typical re-analysis of raw TSWT data for one iteration of run 389.

Should any new analysis technique become available, then a new subroutine could replace or supplement the existing subroutines, and be incorporated in OFLEX and/or ORLEX by minor adjustments to the main program and the LINK commands.
3.1. Program TSWT

The main control program listed on Figure 4.1 reads tests parameters from the tunnel operator and sequences subroutine calls:

Lines 0002 - 0009

Define all common data blocks used thus:

Block 1

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NJ</td>
<td>Effective number of jacks per wall (real plus dummy).</td>
</tr>
<tr>
<td>MT</td>
<td>Number of model pressure tappings.</td>
</tr>
<tr>
<td>NR</td>
<td>Number of tunnel reference samples during a scanivalve scan.</td>
</tr>
<tr>
<td>CL1</td>
<td>Printout control value.</td>
</tr>
<tr>
<td>BL</td>
<td>Prandtl-Glauert scaling factor = (\sqrt{1-M^2})</td>
</tr>
<tr>
<td>PR1</td>
<td>For development only.</td>
</tr>
<tr>
<td>AK1</td>
<td>Aerodynamic coupling factor.</td>
</tr>
<tr>
<td>AK3</td>
<td>Wall movement scaling factor.</td>
</tr>
<tr>
<td>AN</td>
<td>Model angle of attack (degrees).</td>
</tr>
<tr>
<td>R3</td>
<td>Chord Reynolds number.</td>
</tr>
<tr>
<td>PP2</td>
<td>Dynamic pressure allowing for compressibility.</td>
</tr>
<tr>
<td>ITRN</td>
<td>Run number.</td>
</tr>
</tbody>
</table>

Block 2

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RN</td>
<td>Bottom wall potentiometer outputs, volts.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Top wall potentiometer outputs, volts.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Bottom wall imaginary wall velocities.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Top wall imaginary wall velocities.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Top wall real wall static pressures.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Bottom wall real wall static pressures.</td>
</tr>
<tr>
<td>&quot;</td>
<td>Development data.</td>
</tr>
</tbody>
</table>
Block 3 Array  D = Measuring point co-ordinates.
    "   WTY = Top wall movement from the straight wall contours (Inches).
    "   WBH = Bottom wall movement from the straight wall contours.
    "   WLY = Effective panel length per wall tapping.
    "   B = Model Cps.
    "   PD = Transducer Cps.

Block 4 Array  E = Calculated imaginary external top wall velocities.
    "   H = Calculated imaginary external bottom wall velocities.
    "   Y = Predicted top wall movements required (Inches).
    "   G = Predicted bottom wall movements required (Inches).
    "   WI = Real top wall velocities squared \((V/U_\infty)^2\)
    "   XI = Real bottom wall velocities squared \((V/U_\infty)^2\)

Block 5 Array  U = Difference between real and imaginary top wall velocities \((u/U_\infty)\)
    "   V = Difference between real and imaginary bottom wall velocities \((u/U_\infty)\)
    "   DS = 'Straight' wall local \(\delta^*\) values.
    "   DET = Local \(\Delta\delta^*\) values.
    "   CS = Matrix coefficients for camber interference assessment.
Block 7 Array PC = Potentiometer calibrations (Volts/Inch)

Block 8 Array TAB = Ambient Temperature (Deg. C)
AMP = Ambient Pressure (Cm Hg)
IFN = Iteration Record No:
IAM = Automatic In-file selection trigger (0-off; 1-on)

Block 9 Array IX = Pressure channel zeroes.

Line 0010 Initialise Arrays X & DET.

Line 0012 Set streamlining trigger (PRl) to zero (i.e. walls not streamlined).

Line 0013 Set automatic in-file selection trigger (IAM) to zero (i.e. Non automatic selection).

Lines 0014 - 0015 Define File 5 as data file RUN.DAT.

Lines 0016 - 0022 Read run data from RUN.DAT and load ITRN with the run number, IFN with iteration record number, TAB with ambient temperature, AMP with ambient pressure and MT with number of mode 1 taps.

Line 0023 Set extended printout trigger (CLl) to zero (i.e. minimal print-out required).

Line 0024 NR = Number of tunnel reference samples in a pressure scan.

Line 0025 NJ = Number of measuring points per wall (See Figure 7).

Line 0026 - 0048 Input test parameter:

Angle of attack (AN)

Lines 0049 - 0059 Subroutine call sequence.

Line 0053 If no model tappings (i.e. MT=0) do not call Subroutine FORCE.

Line 0057 If walls streamlined (i.e. PRL=1) do not call Subroutine WALL
When the Iteration Record number is 50 and jump out of streamlining loop.

Increment the Iteration Record number

If walls are un-steamlined (i.e. PR1=0) continue the streamlining cycle.

Load run data array TUN and load file RUN.DAT with new values of TUN.

3.2. Subroutine DATA (Incorporating Subroutines GAD and STEP)

This subroutine primarily acquires all the tunnel pressures either from the tunnel via an automatically stepped scanivalve or from disc storage. Some data is then reduced.

The option to acquire old data stored on disc allows re-analysis of previous runs. All analogue to digital conversions associated with the pressure data are handled by software in Subroutine GAD. Automatic control of the Scanivalve is achieved by software in Subroutine STEP. The listing on Figure 4.2 can be broken down thus

Dimension of data blocks DA, DB, DC and IDATA.

Define only common data blocks required in Subroutine DATA.

Initialise the first row of 2D- array IDATA.

CHD = Model chord = 4 inches (NACA 0012-64)
      = 6 inches (NPL 9510)

NJ1 = number of wall jacks.

NR = number of tunnel reference samples in a pressure scan.

IPL = Approximate low value of scanivalve position encoder output at pressure scan start condition.

Transducer calibration valves (cm Hg per A-D count).
If a re-analysis (i.e. Cl = 500), open data file 4, select an existing record (IR) and fill IDATA with raw transducer A-D counts from a previous TSWT run.

Load pressure channel zeroes into IDATA

Identify all passes of the data acquisition cycle other than the first and print a prompt.

Initiate DO loop to acquire tunnel pressures using the scanivalve system.

If the first data acquisition cycle (i.e. X(30) = 0) read the four pressure transducer zero signals and halt the software until a 'CR' is received from the terminal - transmitted when the tunnel flow is stabilised by the operator.

Wait for a change in the encoder output generated by a scanivalve step using Subroutine STEP. If there is a step error halt program.

Sample four transducer outputs using Subroutine GAD and load array IDATA.

Load pressure channel zeroes into array IX

Open a data file 4 and fill a selected record (IOT) with 192 raw A-D count values from the iteration, stored in IDATA.

Convert raw data into pressures relative to tunnel reference pressure (cm Hg), and load into array PD.

Load arrays P and Q with wall jack centreline pressures with dummy measuring points included (See Figure 7).

Load arrays P and Q with mid-jack centreline pressures.

RD (NR + 1) = test section total pressure reading.

RD (NR + 2) = Ambient pressure (cm Hg).

RD (NR + 3) = Ambient temperature (deg C).

RL = Stagnation pressure (cm Hg)
Calculate tunnel reference Mach numbers at regular intervals through the pressure scan, and look for excessive fluctuations (i.e. $\Delta M > .01$).

Calculate selected tunnel pressures for possible calibration checks.

Store average tunnel reference static pressure in Bl.

Subroutine GAD

Line 0001 Subroutine label with data transfer of value N and array IDATA.

Line 0002 Dimension of IDATA (Average A-D counts) and IDT (Raw A-D counts) arrays.

Line 0003 $NS = \text{Number of samples per channel}$

Lines 0004 - 0006 Time delay of 50 milli-seconds using the programmable clock (Subroutine SETR).

Lines 0007 - 0011 Take NS samples of four analogue input channels at 1kHz and store digital results in IDT.

Lines 0012 - 0020 Convert all raw A-D counts from packed integer format to real number format (Subroutine CVSWG) then take the average of the NS samples per channel and store in IDATA.

Line 0021 Stop the programmable clock.

Subroutine STEP

Line 0002 POKE digital output register with the command number 16384 (See Appendix A). Note no scanivalve address is required with the present hardware configuration.

Lines 0003 - 0005 Time delay of 20 milli-seconds using the programmable clock.

Line 0006 POKE digital output register with a zero value to clear.
3.3. Subroutine REDUCE

This subroutine performs data reduction operations on tunnel pressure information and acquires further tunnel data from data files held on the mass storage device. The software listed on Figure 4.3. is described thus:

Lines 0004 - 0010 Define only common data blocks required in subroutine REDUCE.

Lines 0011 - 0015 Set variables, as previously described for subroutine DATA.

Lines 0016 - 0025 Calculate the average freestream Mach number of the run.

Line 0026 \( PP2 = \frac{qI}{q_c} \) compressibility correction to tunnel q.

Lines 0027 - 0041 Calculate chord Reynold's number (R3).

Lines 0042 - 0054 Convert tunnel pressure data into Cp using the tunnel reference pressure associated with each group of six readings.

Lines 0055 - 0080 Load model pressure coefficients into array B from transducer channels 1 and 3 data. Note this software is model dependent and configured for use of the NPL 9510 section.

Lines 0083 - 0092 Specify IWF corresponding to 'Straight Wall' base data for Mach number panels \( M < .725 \), \( .725 < M < .825 \) or \( M > .825 \) for re-analysis case only. Otherwise a dummy data base is used (i.e. IWF=1).

Lines 0094 - 0096 Load arrays DA, DB and PC with data from File 2 (TUN.DAT).

Lines 0097 - 0098 Define File 3 as data file PAD.DAT.

Lines 0099 - 0109 Define wall contour record number automatically (i.e. IAM=1) or by operator input.

Line 0110 Load array DC with 'wall data' from File 3.
Load arrays WTY, WBY and DS with data from array DA.

Load arrays CS and D with data from array DB

Read coupling factor (AK1) and scaling factor (AK3) from File 2, record 10.

Calculate effective wall lengths between mid-jack wall points and store results in array WL.

Load arrays RS, RN, X, W with data from array DC (See section 3.1 for array descriptions).

Convert the wall Y co-ordinates into movement from straight contours in inches.

Close data files

Signify the completion of the data acquisition cycle by setting X(30) = 1.

3.4. Subroutine WAS

The wall adjustment strategy\(^1,2,3\) (WAS) manipulates the calculated imbalance between real and imaginary wall velocities to generate a wall movement which will give zero wall loading or vorticity.

The strategy requires interpolation of real wall velocities at regular intervals along each wall. To ensure accurate interpolation at the wall ends using curve fitting to the wall velocity distribution, two straight dummy wall extensions are added to each flexible wall, 15.25 cm (6 inches) and 22.86 cm (9 inches) in length respectively. The dummy wall measuring points introduced are points 1 and 2 upstream where the wall velocity is assumed free-stream and points 22, 23 and 24 downstream, which have a wall velocity equal to that of measuring point 21 or jack 19. Measuring point 1 is also assumed to be the origin of the wall boundary layer. Measuring points 3 to 21 are real and correspond to the position of wall jacks. The software representation of each flexible wall, in terms of measuring point is shown in Figure 7. The software listed on Figure 4.4. is described below

Define only common data required in subroutine.
Dimension working arrays.

Message to tunnel operator.

Equalising top and bottom wall coupling and scaling factors.

Compute external velocities for next wall shape where

\[ Q_l = \text{compressible dynamic pressure} \]

\[ \text{TEMP}1 \& \text{TEP}1 = \text{incompressible top and bottom wall Cps.} \]

\[ U \& V = \text{inbalance between real and imaginary wall velocities (Top and bottom walls).} \]

\[ E \& H = \text{external velocity perturbations} \]

\[ \text{XI} \& \text{WI} = \text{real wall velocities squared } (v/u_\infty)^2 \]

Load Z with interjack streamwise spacing co-ordinates.

Set up DO loop for both walls.

Load A & XB with D & U or V data in sets of four values for top or bottom wall.

Cubic spline fit to each set of data to obtain wall velocity imbalance between jacks.

Summation of velocity induced by the vorticity distributed along a wall at each jacking point. The result is stored in arrays S for the top wall and T for the bottom wall.

End of wall velocity analysis repeated for each wall.

Numerical integration of top and bottom wall normal velocity components to generate jack movement demands.

Raw movements are stored in TT and R. Arrays Y & G store the effective jack movements after scaling.

Likewise E and H (the external velocity perturbations) are modified by scaling in lines 0133 & 0134. In lines 0135 & 0136 values in arrays Y and G are converted to compressible values using \( B1 = \sqrt{1-M^2} \).
3.5. Subroutine STAR

The function of STAR is to calculate Mach number and boundary layer displacement thickness values for each wall jacking point necessary only for re-analysis information. The boundary layer calculations use a numerical solution of the Von Karman Momentum Integral equation for a turbulent boundary layer. The software listing is shown on Figure 4.5

Line 0002  
Dimension of array DELTA for $\delta^*$ storage.

Lines 0003 - 0007  
Define only required common data blocks.

Lines 0008 - 0011  
Calculation of isentropic flow relationships.

Lines 0012 - 0013  
Calculation of air density (kg/ft$^3$).

Lines 0014 - 0015  
Calculation of air temperature (K).

Lines 0016 - 0017  
Calculation of air viscosity (lb/ft$^2$sec$^{-1}$).

Line 0018  
CLI value selects extended printout when equal to 500.

Lines 0027 - 0145  
Overall Do loop performs calculations for each jacking on both walls using sets of three jacks, labelled 0, 1 and 2.

Lines 0044 - 0045  
Load X1 and X2 with jack 1 and 2 co-ordinates relative to jack 0.

Lines 0046 - 0051  
Ensure that the correct tunnel reference pressure is used with each wall pressure.

Lines 0053 - 0062  
Load SP1, SP2 and SP3 with top or bottom wall velocity perturbations.

Lines 0063 - 0068  
Calculate wall velocities U0, U1 and U2 (ft/sec).

Lines 0069 - 0077  
Calculate local wall mach number and store in P (top wall) or Q (bottom wall).

Lines 0081 - 0087  
For the second measuring point assume a turbulent boundary layer growth according to $\delta^* = \frac{0.0213x}{R_x^{0.1428}}$.

Lines 0088 - 0092  
Calculate the velocity gradient at jack 1 and store value in D2.
Guess the size of $\delta^*$ and store as Pl.

Calculate components of M.I. equation.

Check if the measuring point is a special case (points 3 or 24 along each wall) and calculate $S_1$ with appropriate velocity gradient ($S_1 = \text{rate of change of } \delta^* \text{ with streamwise distance}$).

Calculate $\delta^*$ from $d\delta^*/dx$ and store value in P4.

Home-in on correct value of Pl (If P4 > Pl increase Pl by 25 E-6).

Load array DELTA with $\delta^*$ values (inches).

Calculate $\Delta\delta^*$ from 'straight wall' to contoured wall.

Output results to a terminal if $CL_1=500$.

Isolation of special cases requiring P4, ST & D2 results to be stored in new addresses.

Load array DELTA with $\delta^*$ values (inches).

Calculate wall Mach numbers between jacks and store results in arrays P and Q, corresponding to top and bottom wall data.

3.6. Subroutine SUME

SUME assesses the residual interferences, due to any residual wall loading, along the tunnel centre-line and the model chord line. The three components of the interferences are related to estimated changes in $C_L$. The quality of streamlining is then assessed from these interferences and a measure of the average pressure imbalance across the walls. A listing is shown on Figure 4.6.

Define only required Common Data blocks.

Define model in use $IWT = 0$ for NACA 0012-64

$IWT = 1$ for NPL 9510

CHD = Model chord (inches)

OR = X co-ordinate of model pivot point which is at ¼ chord and nominally mid way between the straight walls (Inches).
Lines 0014 - 0016  Calculate model chord line X and Y increments and store in AX and AH respectively.

Lines 0017 - 0018  IPP = Position of model pivot point as a number of chords.

Line 0020  OR = X co-ordinate of the model ¼ chord point - the origin.

Line 0021  Subroutine print trigger (OT) set to zero for print-out.

Line 0023  Eliminate dummy jacks by reducing number of jacks per wall by four.

Line 0025  Y3 = nominal tunnel semi-height (Inches).

Lines 0030 - 0037  Calculate imaginary wall velocity squared and sum the absolute value of the load difference between real and imaginary velocities squared for each jack. Store the results in EE (top wall) and F (bottom wall).

Lines 0038 - 0043  Calculate the average Cp difference between real and imaginary flows for each wall (called E) and store as ET and CEB, and print-out.

Line 0046  If minimal print-out is specified (i.e. CLI = 0) jump to line 0083

Line 0050  For 37 points along the tunnel centreline summate the effect of wall vorticity, at the model and along the tunnel centreline.

Line 0051  X1 = X co-ordinate of measuring point.

Lines 0054 - 0074  Sum the velocity components of the wall vorticity at co-ordinate X1 on the tunnel centreline and store results in arrays UT and VT.

Line 0055  X2 - Horizontal displacement between jack and analysis point (Inches).

Line 0057  Y2 = Displacement of bottom wall downwards from the centreline (Inches).
Calculate the horizontal induced velocity perturbation component due to vorticity at measuring points \( n \). (See Figure 7)

\[
U_l = \frac{Z_n}{2\pi} \times \sum_{\text{Top}} \frac{Y_n}{X_n^2 + Y_n^2} \times \Gamma_n
\]

where \( \Gamma_n \) = Vortex strength

= Local imbalance of wall real and imaginary velocities.

\( X_n \) = horizontal separation of analysis point and a vortex assumed at a measuring point \( n \).

\( Y_n \) = vertical separation of same.

and \( Z_n \) = panel length = distance between mid-jack points spanning measuring point \( n \).

In software notation

\[
U_l = \frac{WL}{TOP}\times((U \times R2) + (V \times R3))
\]

where WL, U and V are arrays.

Calculate the vertical component of induced velocity perturbation \( V_l \) due to vorticity at measuring points \( n \)

\[
V_l = \frac{Z_n}{2\pi} \times \sum_{\text{Top}} \frac{X_n}{X_n^2 + Y_n^2} \times \Gamma_n
\]

If \( OT = 0 \) print-out results.

Calculate velocity perturbations for nine equidistant points along the model chord line.

Calculate induced flow angles along the model chord line (AA2).
Calculate induced $C_p$ (CP1).

Store $\frac{1}{4}$ chord point $C_p$ in address CP.

Load $A_1$ and $A_2$ with leading edge and trailing edge induced flow angles.

$SP = $ Contribution to induced camber.

Differentiate between odd and even numbered measuring points.

Load accumulators for Simpsons rule numerical integration technique.

$CPE = $ Average $C_p$ errors induced along model chord.

$P_3 = $ Induced camber in terms of $\Delta C_L$.

$A_3 = $ Induced camber (degrees).

$CL = $ Induced Angle of Attack in terms of $\Delta C_L$.

$A_1 = $ Induced Angle of attack (degrees).

If a re-analysis print-out all residual interference information (i.e. $CL_1 = 500$).

Print abbreviated version of residual interference information if $CL_1 = 0$.

Increment ISC for each quality of streamlining satisfied.

If all the qualities of streamlining are satisfied (i.e.$ISC = 5$) $PR_1 = 1$, the walls are streamlined.

If the walls are streamlined print messages to the operator and the line printer.

3.7. Subroutine FORCE (Based on a NASA numerical technique)

FORCE integrates the model pressure distribution to give coefficients $C_L$, $C_D$ and $C_M$. 

- 23 -
3.7.1. NACA 0012-64 version - listed on Figure 4.7a.

Lines 0002 - 0006 Define required Common Data blocks.

Lines 0007 Dimension working arrays.

Line 0010 \( MT = \) Number of model tappings +2 for dummy leading edge tappings.

Lines 0011 - 0012 Open data file 2 (WING.DAT)

Line 0013 Load array AG with X co-ordinates of model tappings (upper surface followed by lower surface).

Line 0014 Load AJ with Y co-ordinates as for the X co-ordinates.

Line 0015 \( IHT = \) Number of model tappings per surface.

Lines 0016 - 0023 Titling and test parameter input from the terminal (\( \text{AN} = \) angle of attack - degrees).

Lines 0024 - 0029 Calculate an effective vertical component for each model tapping, and store in array AH.

Lines 0031 - 0040 Initiate a DO loop with special processing for first and last tappings on each surface and different analysis for each surface.

Line 0042 \( WF = \) Weighting Factor, in this case for \( C_N \).

Line 0043 \( L6 = \) Local components of \( C_N \) for upper surface.

Line 0044 \( L5 = \) Sum of \( C_N \) components.

Line 0045 Store negative L6 in array A.

Lines 0047 - 0050 Calculate the upper surface \( C_N \) component for the dummy leading edge tap.

Lines 0051 - 0056 Calculate the \( C_N \) component for the last downstream tap on the upper surface.

Line 0055 \( C_N = \) Upper surface \( C_N \).

Lines 0057 - 0061 Calculate the lower surface \( C_N \) component for the dummy leading edge tap.

Lines 0062 - 0067 Calculate \( C_N \) component for the last downstream tap on the lower surface.
Calculate $C_N$ components for lower surface tappings.

Do loop to calculate $C_L$ components for each tapping.

$C_5 =$ Upper surface $C_c$ component.

Store $C_5$ in array AH.

$C_3 =$ Sum of $C_c$ components.

$CCP =$ Super surface $C_c$.

$C_5 =$ Lower surface $C_c$ component.

$St =$ Lowers surface model $C_c$.

$C_3 =$ Total $C_c$.

Do loop to calculate $C_m$ components about the model leading edge.

$BB =$ Upper surface $C_m$ components.

$C_9 =$ Sum of upper surface $C_m$ components.

$TIT =$ Tapping position (% chord).

$BB =$ Lower surface $C_m$ components.

$C_8 =$ Sum of lower surface $C_m$ components.

$STl =$ Total $C_m$.

Output forces $C_N$, $C_c$, $C_m$.

Calculate coefficients $C_L$ and $C_D$ from $C_N$, $C_c$ and angle of attack.

Output wing performance coefficients.

Close data file.

3.7.2. NPL 9510 version - listed on Figure 4.7b

This version of FORCE is the same as the NACA 0012-64 version in all respects, except for the number of model tappings (50 in total, with a split 32:18 on upper and lower model surfaces respectively) and the labelling associated with the print-
out. Also notice that print-out control commands have been inserted at lines
0015, 0092, 0112, 0120 and 0126. Setting CLI = 0 suppresses all print-
out except data associated with the label 'WING PERFORMANCE'. NPL 9510 model
Cps are stored in NPL.DAT for subsequent plotting (See next section).

3.8. Subroutine SET

SET controls the output of tunnel data to the VDU console and the line printer
and the output of results to disc storage. This software is configured for test-
ing the NPL 9510 section. A listing is shown on Figure 4.8.

Line 0002  Dimension data array DA.
Line 0003 – 0009  Define required Common Data blocks.
Line 0010  Initialise some array elements for print-out purposes.
Lines 0011 – 0012  Open wall output file 3 with 50 records each 512
words in length (PAD.DAT).
Lines 0013 – 0014  Open wing output file 2 with 50 records each 128 words
in length (NPL.DAT).
Line 0015  NJI = Number of wall jacks.
Line 0016  Define model used: IWT = 0 for NACA 0012-64;
IWT = 1 for NPL 9510.
Line 0017  CLI = 0 for minimal print-out, so jump to line 0055
Lines 0019 – 0035  Output of pressure transducer Cps, external wall
velocity for each jack and the predicted jack move-
ment for zero wall interference.
Lines 0036 – 0039  Print-out top wall labels.
Lines 0042 – 0043  Define File 6 as WALM.DAT
Lines 0044 – 0053  Output wall Mach number distributions stored in
arrays P and θ into an operator set (IMR) record of
File 6 for subsequent plotting.
Lines 0055 – 0080  Store potentiometer values for new wall contours in
array DA. If CLI = 500 print-out information on jack
X & Y co-ordinates, local Mach number, current poten-
tiometer volt values, predicted potentiometer volt
values for the next iteration.
Line 0060  WM = Effective position of top wall contour allowing for boundary layer growth along the wall.

Lines 0061 - 0069  If movement demands for the downstream jacks (No. 16-20) exceed mechanical limits in terms of pot volts, set to safe minimum values.

Line 0078  If CLI = 500 print out the local wall Mach Number between jacks.

Lines 0081 - 0115  Repeat analysis and print-out for the bottom wall.

Lines 0105 - 0114  Calculation of wall Mach number standard deviation for top and bottom walls for use in special re-analysis of tests with an empty test section.

Line 0116  Set KT to equal the total number of jacks.

Lines 0117 - 0118  Store top wall external velocity perturbations in array DA.

Lines 0122 - 0126  Print prompts on wall and model output records.

Line 0128  Write array DA into record IFN of File 3 on disc.

Lines 0129 - 0132  Define the number of model tappings and store model Cps in record IFN of File 2 on disc.

Lines 0133 - 0134  Close data files 2 and 3.

3.9. Subroutine WALL (incorporating Subroutines INIT, START, MOVE and DIO)

This subroutine controls the test section wall adjustments. Both walls are moved simultaneously in variable increments of movement. Each jack is commanded to move away from the model in its last adjustment to ensure the walls are rigid to air pressure loads. Numerous safety checks are included to guard against jacks jamming or jacks out of control.

Subroutine INIT initialises the control system. Subroutine START switches the jacks on and off for a time interval proportional to the average position error of the wall adjacent to the model pressure surface. Subroutine MOVE loads each jack with direction information and checks the data is loaded. Subroutine DIO handles all the Digital input and output operations between wind tunnel and computer.

The software is dependent on the tunnel hardware and is listed on Figure 4.9.
Define Common data required.

NWJ = total number of wall jacks.

Define arrays thus

PC = potentiometer calibration (volts per inch)
IM = stores jack movement demands for checks.
ADC = stores wall pot. volts for movement checks.
PV = wall contour pot. volts.
PL = minimum safe pot. volt readings for downstream jacks.

Load array PL with values

Define File 3 as PAD.DAT and load array PV with a set of wall contour pot. volts from record IFN of File 3.

IF = 1 for the first pass through the wall adjustment cycle

IP = anti-backlash overshoot increment.

Initialise the control system.

Load variables
IW = Wall number (O:top/l:bottom)
ICNT = No. of attempts to move one jack.
IS = No. of jacks correctly positioned.
ITOL = Wall setting tolerance band in pot. volts.

Calculate movement required for each jack from its current location and load each jack with direction information.

Read the potentiometer output for jack (L), convert the result to pot. volts, and store in IPV.

IMOVIe = Actual jack movement from previous position. Ignored on first pass.

Store current jack pot. volts (IPV) in array ADC.

If IPV > 980 (i.e. jack approaching mechanical limit) then stop.
Lines 0029 - 0036  Single out downstream jacks (i.e. No. 16-20 and 36-40) and ensure that the demanded pot. volts are greater or equal to the minimum safety values.

Line 0038  IPD = Demanded jack pot. volts adjusted to overshoot the wall position, towards the model, by an increment IP.

Line 0039  IMV = Required jack movement in pot. volts. (+ve towards the model and -ve away from the model).

Line 0041  If the wall is being adjusted from its overshoot position (i.e. IP = 0) only allow the wall to move away from the model (i.e. IMV < 0). Set IMV = 0 if the jack tries to move towards the model. This technique ensures the wall is rigid to air pressure loads by eliminating jack mechanism backlash in the correct sense.

Line 0043  Change the sign of all bottom wall movement demands for hardware compatibility.

Lines 0045 - 0049  Change the sign of top wall movement demands for all odd numbered jacks to accommodate the hardware configuration for the top wall control.

Line 0051  Jump to line 0063 if this is the first pass (i.e. IF = 1).

Line 0053  For jacks 1 and 21 do not check for jack jamming due to the small movement demands on these jacks positioned close to the wall anchor point (See figure 7).

Line 0055  If the demanded jack movement (IML(L)) is more than 50 pot. volts and the jack potentiometer reading shows only a change of 10 or less pot. volts then warn the operator and abort the wall adjustment.

Line 0057  If only one jack has yet to be correctly adjusted (IS = 39) increment ICNT for each pass through the adjustment cycle.
If IS = 39 and ICNT = 6 then warn the operator and abort the wall adjustment. One jack may have moved out of its potentiometer measuring range.

If the change in jack pot. volts between position samples is greater than 50 warn the operator and abort the wall adjustment. A jack may be moved the wrong way.

Load array IM with values of required jack movement.

If the required movement is less than or equal to ITOL go to statement 40 (line 0067).

Increment IS by one and set IMV = 0.

Load jack (L) with direction information using Subroutine MOVE.

Set IF = 0 to indicate the first pass is complete.

If IS = 40 the wall is correctly adjusted.

Move the walls using Subroutine START.

If IS less than 40 repeat the wall adjustment cycle

If the wall has been correctly adjusted to the demanded contour (IP = 0) finish. If the wall has been correctly adjusted to the overshoot contour (IP = 5) repeat the wall adjustment cycle with IP = 0.

Set IW = 0 for top wall condition.

For all downstream jacks (Nos. 16-20 and 36-40) warn the operator when their pot. limit has been reached.

Subroutine INIT

NWJ = 40 = Number of wall jacks.

Set data operation as write before read IDI = 128.
Line 0004 - 0010
Generate command numbers ICOM for a motor stop directive to each jack and send them individually to the wind tunnel using Subroutine DIO.

Line 0007
Check the data has been correctly loaded.

Subroutine START

Line 0001
Subroutine statement with data transfer of the demanded jack pot. volts (PV) and the required movements or position errors (IM).

Line 0004 - 0006
Determine the average position error of the bottom wall in pot. volts and store the result in IAM.

Line 0007
RCT = number of clock counts between motor power on and off.

Line 0008
The upper limit of RCT is 600. Set by wall damage considerations.

Line 0010
The lower limit on RCT is 25. Set by resolution of the wall position measurements and speed considerations.

Line 0012 - 0014
If RCT > 50 load jacks 1 and 21 with a motor stop because they will overshoot by too large an amount.

Line 0016 - 0019
Generate a command number ICOM to start the motors (See Appendix A) and send it to the wind tunnel using Subroutine D10.

Line 0020 - 0022
Variable time delay set by number of clock counts (RCT) at 100 Hz.

Line 0023
Stop all motors

Subroutine MOVE

Line 0001
Subroutine statement with data transfer of LJ (Jack number) and IMV (required jack movement).

Line 0002
Determine whether IMV is negative, positive or zero.
Load the jack direction data into IDR corresponding to either stop, forward-go or reverse-go.

All data operations are write before read (IDI = 128).

Generate a command number ICOM.

Send ICOM to jack IJ using Subroutine D10.

Subroutine D10

This subroutine performs the handshaking between computer and wind tunnel to transfer data on the digital I/O lines.

Subroutine statement with data transfer ICOM(Command number),IDI (Data operation) and INPUT (Data check)

Load Digital output buffer (address 167774) with command number 45 to ensure all motors are switched off.

Send command number ICOM to the wind tunnel.

Halt program until an external data accept signal is received on the digital I/O status register (handshaking between wind tunnel & computer).

Reset the Digital I/O status register to zero.

If IDI = 0 (Write only operation) finish.

Halt the program until an external data ready pulse is received on the digital I/O status register.

Store the contents of the Digital input register in INPUT.

Initialise the status register.

3.10 Program REAN

The main re-analysis program listed on Figure 4.10 reads test parameters and sequences subroutine calls. The program is very similar to TSWT, the main differences are:
1. The printout trigger is set to give an extended print-out (i.e. 
   CL1 = 500).
2. All test parameters and run details are provided by the operator.
3. The program contains no loops and will run through only once per 
   activation.
4. Subroutine WALL is not included.
5. Subroutine STAR is included to provide additional information during 
   the re-analysis.

3.11 Program RUN

This program has to be activated before each session of TSWT runs, to ensure 
the data file RUN.DAT contains the current ambient conditions, run and file 
numbers.

The software is shown on Figure 4.11 and discussed below.

Line 0002        Define array TUN to hold run data.

Lines 0003 - 0004 Define File 2 as data file RUN.DAT.

Line 0005        Load array TUN with run data stored in RUN.DAT, record 

Lines 0007 - 0009 Indicate if only an increment of the record number is 
   required (Yes: INC = 1).

Lines 0010 - 0011 Read thermocouple A-P channel and load TUN (3) with 
   the current ambient temperature.

Line 0012        If INC = 1 jump to line 0025.

Line 0014 - 0021 Input ambient pressure (cm Hg), the run number and the 
   file number as requested.

Lines 0022 - 0023 Load TUN (1) & TUN (2) with new values.

Line 0025        Increment the record number by one.

Line 0026        Write the modified contents of array TUN to RUN.DAT 
   on disc.

Lines 0027 - 0028 Print the contents of array TUN as a check.
SYMBOLES

\[ V = \text{wall velocity perturbation} \quad (V = U - U_\infty). \]
\[ U = \text{local wall velocity}. \]
\[ U_\infty = \text{freestream velocity}. \]
\[ \delta^* = \text{boundary layer displacement thickness}. \]
\[ x = \text{horizontal distance from the boundary layer origin (Inches)}. \]
\[ R_x = \text{Reynolds number based on boundary layer length (x)}. \]
\[ Y = \text{vertical distance from the tunnel centreline (positive upwards)}. \]
\[ C_L = \text{lift coefficient}. \]
\[ C_D = \text{pressure drag coefficient}. \]
\[ C_m = \text{pitching moment coefficient about the airfoil leading edge}. \]
\[ C_N = \text{normal force coefficient}. \]
\[ C_C = \text{chordwise force coefficient}. \]
\[ M = \text{Freestream Mach number}. \]
\[ X = \text{horizontal distance from leading edge (Inches)}. \]
\[ Y = \text{vertical distance from the chord line (Inches)}. \]


<table>
<thead>
<tr>
<th></th>
<th>REFERENCES</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>M.J. Goodyer and S.W.D. Wolf</td>
</tr>
<tr>
<td>2.</td>
<td>S.W.D. Wolf and M.J. Goodyer</td>
</tr>
<tr>
<td>3.</td>
<td>M. Judd, S.W.D. Wolf and M.J. Goodyer</td>
</tr>
</tbody>
</table>
FIG. 1 SELF-STREAMLINING OPERATING PROCEDURE
FIG. 2  TSWT CONTROL SYSTEM OUTLINE
FIG. 3  T.S.W.T. CONTROL SYSTEM HARDWARE
Figure 4 Listings of TSWT Control Software
SUBROUTINE WAS

COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/N(30),RS(30),W(30),P(30),D(30),RD(20)
COMMON/THREE/D(60),WTY(30),WL(30),B(50),PD(40,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)

DIMENSION A(4),XB(4),C(30,4),Z(30),S(30),T(30)

REAL N

WRITE(7,5)
5 FORMAT(1115X,' WAS COMPUTING NOW !!')

AK2 = AK1
AK4 = AK3
NJ1 = NJ-4

MM = 1
M = NJ

DO 100 I = 1,NJ
10 IF(MM.EQ.NR) GO TO 10

NT = (I-5)/6
IF(I-5).EQ.NT MM = MM +1
Q1 = (RD(MM) - RD(NR+1))/PP2
TEMP = -P(I)/Q1
TEMP1 = B1*TEMP
TEMP = SORT(1-TEMP1)
U(I) = TEMP-X(I)
E(I) = (AK3*U(I)/2)+X(I)
TEP = -Q(I)/Q1
TEP1 = B1*TEP
TEMP = SORT(1-TEMP1)
V(I) = W(I)-TEP
H(I) = W(I)-(AK4*V(I)/2)
XI(I) = (TEMP+1)*(TEMP+1)
CONTINUE

L = M-2

DO 110 I = 1,L
110 Z(I) = (D(I)+D(I+1))/2

DO 175 NN = 1,2
NC = NN-1
120 J = 1,4
KI = I+J
A(J) = D(I+J)
IF(NC.EQ.0) GO TO 25
XB(J) = V(I+J)
GO TO 120

VO = (XB(3)-XB(2))/(A(3)-A(2))
V1 = XB(2)-VO*A(2)
V2 = 1/(A(4)-A(1))
V3 = (XB(4)-VO*A(4)-V1)/(A(4)-A(2))*(A(3)-A(4))
CALL REDUCE
CALL WAS
CALL SUME
IF(MT.EQ.0) GO TO 100
CALL FORCE
100 CALL SET
IF(FRI.EQ.1.) GO TO 200
CALL WALL
200 IAM=1
IF(IFN.EQ.50) WRITE(7,400)
400 FORMAT(/60('**')/" REMEMBER TO COPY DATA FILES BEFORE NEXT RUN'
C /60('**'))
IF(IFN.EQ.50) GO TO 250
IFN=IFN+1
IF(FRI.EQ.0.) GO TO 500
250 TUN(1)=ITRN+1
TUN(2)=IFN
IFN=IFN+1
IFN.EQ.50) TUN(1)=ITRN
IFN.EQ.50) TUN(2)=4.
510 CALL ASSIGN(4,'RUN.DAT',0,'OLD',)
DEFINE FILE 4 (6r256,U,IJR)
WRITE(4,'(TUN(N),N=1,6)'
CALL CLOSE(4)
END
TSUT
SUBROUTINE DATA

AUTO SCAN

DIMENSION DA(100),DB(100),DC(100),SD(4)
DIMENSION IDATA(49,4)
COMMON/ONE/NJ,HT,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),WT(30),X(30),P(40),Q(40),RD(20)
COMMON/THREE/D(60),WTY(30),WBY(30),WL(50),B(50),PB(48,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),W(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
COMMON/SEVEN/PC(50)
COMMON/EIGHT/TAB,AMP,IFN,IAM
COMMON/NINE/I(4)
DATA IDATA(1,1),IDATA(1,2),IDATA(1,3),IDATA(1,4)/0,0,0,0/
DATA ACQUISITION

CHD = 4.
NJ1 = NJ-4
M = NJ
NR = 8
IF1 = 2050
S0(1) = .0331668
S0(2) = .0142486
S0(3) = .0134165
S0(4) = .0141669
IP0 = 0
IF(CL1.EQ.0) GO TO 36
WRITE(5,31)
31 FORMAT(’/’’DATA INPUT FILE = ’)
CALL ASSIGN (4,-1,’OLD’,)
DEFINE FILE 4 (50,512,U,IJR)
WRITE(7,32) IFN
32 FORMAT(40(’*’)/’ ITERATION RECORD NO. = ’,I4/40(’*’))
READ(4,IFN)((IDATA(N,J),N=1,49),J=1,4)
CALL CLOSE(4)
GO TO 44
IDATA(1,1)=IX(1)
IDATA(1,2)=IX(2)
IDATA(1,3)=IX(3)
IDATA(1,4)=IX(4)
IF(CL1.EQ.0.AND.,X(30).EQ.1)WRITE(5,800)
800 FORMAT(’ SCANIVALVE GO’/13(’=’))
DO 25 N=1,49
25 IF(N.GT.1) GO TO 40
IF(X(30).EQ.1,) GO TO 25
CALL GAP(N,IDATA)
WRITE(5,15)
15 FORMAT(’/’’TURN ON WIND’)
READ(5,35) ISTOP
FORMATT(12)
GO TO 25
FORMATT(13)

Figure 4.2.1.
CALL STEP
IENC=CVSWG(IADC(20,1),1)
IF2=IABS(IENC-IP1)
IF1 = IENC
IF(IF2.GT.200) GO TO 30
WRITE(7,43)
FORMAT(//'STEP ERROR')
GO TO 180
CALL GADeN,IDATA)
CONTINUE
IX(1)=IDATA(1,1)
IX(2)=IDATA(1,2)
IX(3)=IDATA(1,3)
IX(4)=IDATA(1,4)
CALL ASSIGN(4,'ADC.DAT',0,'OLD',)
DEFINE FILE 4(50,512,U,IJR)
WRITE(7,42) IFN
DATA REDUCTION
DO 45 J = 1,4
DO 5 N = 2,49
IDATA(N,J)= IDATA(1,J)-IDATA(N,J)
CONTINUE
CONTINUE
DO 110 N = 22,24
P(N) = PD(N,2)
Q(N) = PD(N,4)
CONTINUE
DO 115 N = 3,21
IA = N-2
P(N) = PD(IA,2)
Q(N) = PD(IA,4)
CONTINUE
DO 120 N = 25,33
IT=IT+1
IF(N.EQ.28) IT=IT+1
P(N)=PD(IT,2)
Q(N)=PD(IT,4)
CONTINUE
RD(NR+1) = PD(48,1)
RD(NR+2) = AMP
RD(NR+3) = TAB
AB1 = 0.
Figure 4.2.2.
0106    R1 = RD(NR+2) - RD(NR+1)
0107    DO 55 J = 1,NR
0108    ITR = 1 + ((J-1)*6)
0109    RD(J) = PD(ITR,1)
0110    AD1 = AD1 + RD(J)
0111    R2 = RD(NR+2) - RD(J)
0112    R3 = 0.28571*ALOG(R1/R2)
0113    PP1 = 5.0 * (EXP(R3)-1)
0114    TFM = SORT(PP1)
0115    IF(J.EQ.1) TFM1=TFM
0117    TR = RD(J)/2.54
0118    IF(J.EQ.1) GO TO 55
0120    IF(ABS(TFM-TFM1).GT.0.01) WRITE(7,56)
0122    56 FORMAT(/' MACH NO. ERROR'/20('*'))
0123    55 CONTINUE
0124    H1 = PD(1,2) + RD(1)
0125    H2 = PD(43,2) + RD(6)
0126    H3 = PD(20,2) + RD(4)
0127    H4 = PD(1,4) + RD(1)
0128    H5 = PD(43,4) + RD(6)
0129    H6 = PD(20,4) + RD(4)
0130    H7 = PD(2,1)
0131    H1 = H1/2.54
0132    H2 = H2/2.54
0133    H3 = H3/2.54
0134    H4 = H4/2.54
0135    H5 = H5/2.54
0136    H6 = H6/2.54
0137    H7 = H7/2.54
0138    AS = AD1/NR
0139    TR = AS/2.54
0140    SP = RD(NR+1)/2.54
0141    BI=AS
0142    180 RETURN
0143    END

DATA

Figure 4.2.3.
SUBROUTINE GAP (N, IDATA)
DIMENSION IDATA(49,4), IDT(500)
NS=15
IEND = 0
CALL SETR(4,0,50., IEND)
CALL LWAIT(0, IEND)
IEND = 0
IRTS=0
CALL RTS(IDT, 60,, NS, 16, 4,, 2, IRTS, IDUM)
CALL SETR(4, 9, 1,, IEND)
CALL LWAIT(0, IRTS)
DO 5 J = 1, 4
DO 70 NN = 1, NS
IS = ((NN-1)*4)+J
IDT(IS) = CVSGLG(IDT(IS), 1)
SUM = SUM + IDT(IS)
CONTINUE
70 IDATA(N, J) = SUM/NS
CONTINUE
5 CALL SETR(-1,,)
RETURN
END

SUBROUTINE STEP
CALL IPOKE("167774, 16384")
IEND=0
CALL SETR(4, 0, 20,, IEND)
CALL LWAIT(0, IEND)
CALL IPOKE("167774, 0")
RETURN
END

Figure 4.2.4.
SUBROUTINE REDUCE

DIMENSION DA(100), DB(100), DC(100), SD(4)

DIMENSION IDATA(49, 4)

COMMON/ONE/NJ, NT, NR, CL1, B1, PR1, AK1, AK3, AN, R3, PP2, ITRN

COMMON/TWO/RN(30), W(30), X(30), P(40), R(40), RD(20)

COMMON/THREE/D(60), WTY(30), WBY(30), UL(30), B(50), PB(48, 4)

COMMON/FOUR/E(30), H(30), Y(30), G(30), WI(30), XI(30)

COMMON/FIVE/U(30), V(30), DS(60), DET(60), CS(20)

COMMON/SEVEN/PC(50)

COMMON/EIGHT/TAB, AMP, IFN, IAM

DATA ACQUISITION

AS = B1
CHD = 4.
NJ1 = NJ-4
M = NJ
NR = 8
MM = 1
R1 = RD(NR+2) - RD(NR+1)
R2 = RD(NR+2) - AS
R3 = 0.28571 * ALOG(R1/R2)
PP1 = 5.0 * (EXP(R3) - 1)
AH1 = SQRT(PP1)
B1 = SQRT(1 - PP1)
WRITE(5, 90) AH1
WRITE(7, 90) AH1
FORMA(//' HACH NO. = ', F8.4)
PP2 = 1.0 + (0.25 * PP1) + (0.025 * PP1 * PP1)
CC1 = 1 + (0.2 * PP1)
CC2 = 2.5 * ALOG(CC1)
CC3 = EXP(CC2)
D1 = RD(NR+2) * 8.998E-3 / (273.15 + RD(NR+3))
D1 :: D1/CC3
AT = RD(NR+3) + 273.15
AT = AT / CC1
CC4 = 0.92 * ALOG(AT/273.15)
V1 = 11.52 * EXP(CC4)
PP3 = (AS - RD(NR+1)) / PP2
U0 = SQRT(56.369 * PP3 / D1)
R2 = U0 * D1 * 3218E4 / V1
R3 = R2 * CHD / 12.
WRITE(7, 95) R3

FORMAT(//' REYNOLDS NO. = ', E12.6)
DO 60 J = 2, 4
MM = 1
DO 65 N = 1, 48
IF(MM.EQ.NR) GO TO 170
NT = (N-5) / 6
DT = NT * 6
IF((N-5).EQ.NT) MM = MM + 1
Q1 = (RD(MM) - RD(NR+1)) / PP2
PD(N, J) = -PD(N, J) / Q1
CONTINUE
CONTINUE

DO 500 J=16,24
IJ=J+8
JB=J+29
IJJB=J+22
B(IJ)=PD(J,3)
IF(J.GT.21) GO TO 500
B(JB)=PD(IJJB,3)
CONTINUE

DO 600 K=2,42
NT=(K-5)/6
NJ=(K-7)/6
IF((K-5).EQ.NT) MM=MM+1
IF((K-7).EQ.NJ) GO TO 600
Q1=(RD(MM)-RD(NR+1))/PP2
B(NP)=(RD(MM)-PD(K,1))/Q1
NP=NP+1
IF(K.EQ.25) NP=33
CONTINUE

CALL ASSIGN(2,'TUN.DAT',0,'OLD',)
DEFINE FILE 2 C12,256,U,IJR)
IF(CL1.EQ.0.) IWF=1
IF(CL1.EQ.0.) GO TO 77
WRITE(7,75)
FORMAT('FILE NO. ?'/',1 FOR M<.725'/',2 FOR .725<M>.825')
WRITE(7,76)
FORMAT('3 FOR M>.825 ANS = ')!
READ(7,35) IWF
READ(2'12) (DA(J),J=1,80)
READ(2'11) (DB(J),J=1,33)
NJ2 = NJ1 + NJ1
READ(2'12) (PC(J),J=1,NJ2)
CALL ASSIGN(3,'PAD.DAT',0,'OLD',)
DEFINE FILE 3 (50,512,U,INR)
IF(IAM.EQ.1) GO TO 20
WRITE(5,140)
FORMAT('S WALL RECORD? = ')!
READ(5,35) IR
IF(CCL1.EQ.0.) GO TO 41
GO TO 40
IR=IFN-1
WRITE(7,30)
FORMAT('WALL CONTOURS RECORD =',14)
READ(3'IR) (DC(J),J=1,88)
DO 150 K = 1,80
112 IF(K.LE.20) UY(K) = DA(K)
114 IF(K.LE.40.AND.K.GT.20) WY(K-20) = DA(K)
116 IF(K.GT.40) BS(K-40) = DA(K)
150 CONTINUE

Figure 4.3.2.
0119    DO 155 J = 1,33
0120    IF(J.LE.9) CS(J) = DB(J)
0122    IF(J.GT.9) D(J-9) = DB(J)
0124  155 CONTINUE
0125    READ(2,10) AK1,AK3
0126    DO 156 J = 1,NJ1
0127    NJJ = J + 2
0128    WL(J) = (D(NJJ+1)-D(NJJ-1))/2
0129  156 CONTINUE
0130    DO 160 K = 1,88
0131    IF(K.LE.20) RS(K) = DC(K)
0133    IF(K.GT.20.AND.K.LE.40) RN(K-20) = DC(K)
0135    IF(K.GT.40.AND.K.LE.64) X(K-40) = DC(K)
0137    IF(K.GT.64) W(K-64) = DC(K)
0139  160 CONTINUE
0140    DO 175 J = 1,NJ1
0141    WTY(J)=WTY(J)-RS(J)
0142    WTY(J)=WTY(J)/PC(J)
0143    WBY(J)=RN(J)-WBY(J)
0144    WBY(J)=WBY(J)/PC(J+NJ1)
0145  175 CONTINUE
0146    CALL CLOSE(3)
0147    CALL CLOSE(2)
0148    X(30)=1.
0149    RETURN
0150    END
REDUCE

Figure 4.3.3.
SUBROUTINE WAS
COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(30),Q(30),RD(20)
COMMON/THREE/D(60),WTY(30),WBY(30),WL(30),B(50),PD(48,4)
COMMON/FOUR/E(30),U(30),V(30),DS(60),DET(60),CS(20)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
DIMENSION A(4),XB(4),C(30,4),ZC(30),S(30),T(30)
REAL N
WRITE(7,5)
FORMAT('WAS COMPUTING NOW!!')
AK2 = AK1
AK4 = AK3
NJ1 = N-4
36
MM = 1
M = NJ
DO 100 I = 1,NJ
IF(MM.EQ.NR) GO TO 10
NT = (I-5)/6
IF(CC-5).EQ.NT) MM = MM + 1
Q1 = (-RD(MM) + RD(NR + 1))/PP2
TEMP = -P(I)/Q1
TEMP1 = B1*TEMP
TEMP = SQRT(1-TEMP1)
TEMP = TEMP-1
U(I) = TEMP-X(I)
E(I) = (AK3*U(I)/2) + X(I)
TEP = -Q(I)/Q1
TEP1 = B1*TEP
TEP = SQRT(1-TEP1)
TEP = TEP-1
V(I) = W(I)-TEP
H(I) = W(I) - (AK4*V(I)/2)
W(I) = (TEP+1)*(TEP+1)
XI(I) = (TEMP+1)*(TEMP+1)
CONTINUE
L = M-2
do 110 I = 1,L
Z(I) = (D(I)+D(I+1))/2
do 175 NN = 1,2
NC = NN-1
65
I = 0
do 120 J = 1,4
KI = I+J
A(J) = D(I+J)
15
IF (NC.EQ.0) GO TO 25
XB(J) = V(I+J)
go to 120
XB(J) = U(I+J)
CONTINUE
V0 = (XB(3)-XB(2))/(A(3)-A(2))
V1 = XB(2)-V0*A(2)
V2 = 1/(A(4)-A(1))
V3 = (XB(4)-V0*A(4)-V1)/(A(4)-A(2))*A(3)-A(4))

Figure 4.4.1.

V6 = V2*(V3-V4)

V5 = V4-V6*A1

I = I+1

P1 = A(2) + A(3)

C(I,1) = V1-A(2)*A(3)*V5

C(I,2) = V0+V5*P1-V6*A(2)*A(3)

C(I,3) = V6*P1-V5

C(I,4) = -V6

IF (I.LT.(N-3)) GO TO 35

LI = M-2

DO 130 J = 2,LI

GO TO 130

Z0 = Z(J)

Z02 = Z0*Z0

Z03 = Z02*Z0

SS = 0

K = M-3

DO 140 I = 1,K

S(J) = S(J)/6.28319

GO TO 130

TC1) = SS/6.28319

CONTINUE

R = 0

TT = 0

S(I) = 0.0

T(I) = 0.0

DO 150 I =1,NJ1

GO TO 130

T(J) = SS/6.28319

CONTINUE

175 CONTINUE

75 R = 0

101 TT = 0

102 S(I) = 0.0

103 T(I) = 0.0

104 DO 150 I =1,NJ1

105 TO = S(I)

106 R0 = T(I)

107 T1 = Z(I)

108 I1 = I+1

109 T2 = D(I1)

110 T3 = S(I1)

111 R3 = T(I1)

112 T4 = Z(I1)

113 I2 = I+2

Figure 4.4.2.
0114 \quad T5 = D(I2)
0115 \quad T6 = S(I2)
0116 \quad R6 = T(I2)
0117 \quad T7 = Z(I2)
0118 \quad FS1 = (T6-T3)/(T7-T4)
0119 \quad FS2 = (R6-R3)/(T7-T4)
0120 \quad T2SQ = T2*T2
0121 \quad T5SQ = T5*T5
0122 \quad T8 = (FS1-(T0-T3)/(T1-T4))/(T7-T1)
0123 \quad T9 = FS1 - T8 \times T7
0124 \quad P2 = (T3-T9*T4)*(T5-T2)+T8*((T5SQ*T5)-(T2SQ*T2))/3
0125 \quad T7 = TT+P2+(T9-T8*T4)*((T5SQ)-(T2SQ))/2
0126 \quad R8 = (FS2-(R0-R3)/(T1-T4))/(T7-T1)
0127 \quad R9 = FS2 - R8 \times T7
0128 \quad P3 = (R3-R9*T4)*T5-T2)+R8*((T5SQ*T5)-(T2SQ*T2))/3
0129 \quad F = E(I2)
0130 \quad R = R+P3+(R9-R8*T4)*((T5SQ)-(T2SQ))/2
0131 \quad Y(I2) = (AK3*TT)+(AK2*AK4*R)
0132 \quad G(I2) = (AK4*R)+(AK1*AK3*TT)
0133 \quad E(I2) = E(I2) + ((H(I2)-W(I2)+V(I2))\times AK2)
0134 \quad H(I2) = H(I2)+((F-U(I2)-X(I2))\times AK1)
0135 \quad Y(I2) = B1 \times Y(I2)
0136 \quad G(I2) = B1 \times G(I2)
0137 \quad 150 \quad \text{CONTINUE}
0138 \quad \text{RETURN}
0139 \quad \text{END}
SUBROUTINE STAR
DIMENSION DELTA(60)
COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(30),Q(30),RD(30)
COMMON/THREE/D(60),WTY(30),WL(30),B(50)
COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
PP1 = 1 - (B1*B1)
CC1 = 1 + (.2*PP1)
CC2 = 2.5 * ALOG(CC1)
CC3 = EXP(CC2)
D1 = RD(NR+2)*8.998E-3/(273.15+RD(NR+3))
D1 = D1/CC3
AT = RD(NR+3)+273.15
AT = AT/CC1
CC4 = .92*ALOG(AT/273.15)
V1 = 11.52*EXP(CC4)
IF(CL1.EQ.0.) GO TO 56
WRITE(7,S)
FORMAT///SX, ' DELTA STAR CALCS.'///'
WRITE(7,10)
TOP WALL')
FORMAT(2X, 'TAP NO.', 2X, 'DU/DX', 5X, 'MACH NO.', 6X, 'D*', 9X 'DD*')
NJ2 = NJ * 2
RR1 = RD(NR+2) - RD(NR+1)
MM = 1
DO 2 N = 2,NJ2
NN = N
IF(N.GT.NJ) NN = N-NJ
IF(NN.GT.(NJ-2)) GO TO 2
IF(CL1.EQ.500 .AND.N.EQ.(NJ+1)) WRITE(7,11)
FORMAT///' BOTTOM WALL'/
IF(CL1.EQ.500 .AND.N.EQ.(NJ+1)) WRITE(7,10)
MM = 1
IF(N.EQ.(NJ+1)) GO TO 2
IF(N.EQ.NJ.OR.N.EQ.NJ2) GO TO 12
X1 = X(NN) - X(NN-1)
X2 = X(NN+1) - X(NN-1)
IF(MM.EQ.NR) GO TO 110
SP1 = Q(N)
SP2 = Q(N+1)
SP3 = Q(N-1)
GO TO 100
SP1 = P(N)
SP2 = P(N+1)
SP3 = P(N-1)
GO TO 110
NN = N-NJ
SP1 = Q(NN)
SP2 = Q(NN+1)
SP3 = Q(NN-1)
PP3 = (SP1+RD(MM)-RD(NR+1))/PP2
PP4 = (SP2+RD(MM)-RD(NR+1))/PP2
PP5 = (SP3+RD(MM)-RD(NR+1))/PP2
Figure 4.5.1.
U1 = SQRT(56.369*PP3/D1)
U2 = SQRT(56.369*PP4/D1)
U0 = SQRT(56.369*PP5/D1)

LOCAL MACH NO. CALCS

R2 = RD(NR+2) - (SP1+RD(MM))
R3 = 0.28571*ALOG(RR1/R2)
PP1 = 5.0*(EXP(R3)-1)
AM1 = SQRT(PP1)
IF(N.GT.NJ) Q(NN-1)=AM1
IF(N.GT.NJ) GO TO 23
P(NN-1)=AM1

23 IF (N.EQ.2.0R.N.EQ.(NJ+2)) GO TO 22
GO TO 33

R2 = U0*D1*32.18E6/V1
R1 = (R2*X1)/12
C1 = 0.142857*ALOG(R1)
C2 = EXP(C1)
C3 = 0.00127 * X1
P6 = C3/C2
GO TO 133

Y1 = U1 - U0
Y2 = U2 - U0
A1 = (Y2-((X2*Y1)/X1))/((X2*X2)-(X1*X2))
B2 = (Y1-(A1*(X1*X1)))/X1
D2 = (2*A1*X1) + B2

DO 12 MI = 1,1000
12 D0 3 MI = 1,1000

P1 = (25 + M1*100)/1E6
NODE = 1

122 P2 = (U1*P1*D1*32.18E6)/V1
C4 = 0.25 * ALOG(P2)
C5 = EXP(C4)
P3 = 0.0128/C5
IF (N.EQ.NJ.OR.N.EQ.NJ2) GO TO 66
S1 = P3-(3.4*P1*D2*12/U1)
IF (N.EQ.3.0R.N.EQ.(NJ+3)) GO TO 77
GO TO 88

66 S1 = P3-(3.4*P1*D3*12/U1)

P4 = P5+(0.5*(S1+S2)*X1/12)
GO TO 99

77 P4 = (S1*X1/12) + P6
GO TO 99

88 P4 = P5+(0.5*(S1+S2)*X1/12)

IF (NODE.EQ.2) GO TO 155
IF (P4.LT.P1) GO TO 111
CONTINUE

155 IF (P4.GT.P1) GO TO 144

111 P1 = P1 - 25E-6
NODE = 2
GO TO 122

144 P1 = P1 + 25E-6

DELTA(N) = 16.8 *P1
N2 = N-2
Figure 4.5.3.
SUBROUTINE SUME
COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),X(30),P(40),O(40),RD(20)
COMMON/THREE/D(60),UTY(30),WBY(30),WL(30),B(50),PD(48,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),WI(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
DIMENSION UT(37),VT(37)
IWT=1
TOPI = 6.283185
CHD = 4.
IF(IWT.EQ.1) CHD=6.0
OR=24.56
AN1 = AN*.01745
AX = (CHD*COS(AN1)/8.0
AH = (CHD*SIN(AN1))/8.0
IPP = 7
IF(IWT.EQ.1) IPP = 4.33
OR = OR - ((IPP-3)*AX)
DT = 0
TOT = 0.0
NJ1 = NJ-4
Y4 = 0.0
Y3 = 3.0
MNU = 0.0
EE = 0.0
F = 0.0
IJ = NJ-2
DO 160 I = 3,IJ
VI = W(I)+1
V2 = X(I)+1
WI = V1 * V1
X(I) = V2 * V2
EE = EE + ABS(XI(I)-X(I))
F = F + ABS(WI(I)-W(I))
CONTINUE
CET = EE/NJ1
CEB = F/NJ1
IF(CL1.EQ.0.)WRITE(5,170) CET,CEB
WRITE(7,170) CET,CEB
FORMAT(I0X,'WALL CP ERROR'/5X,'TOP - ',F8.4,
      5X,'BOTTOM - ',F8.4)
DO 161 I = 1,37
X1 = OR + ((I-19)*1.0)
SUMU = 0.0
SUMV = 0.0
DO 162 J = 1,NJ1
X2 = X1 - (D(J)+2)
Y1 = Y4+Y3-WTY(J)-DET(J)
Y2 = Y4-Y3-WBY(J)+DET(J)
Figure 4.6.1.
\[ P_1 = X_2 \times X_2 \]
\[ P_2 = P_1 + (Y_1 \times Y_1) \]
\[ P_3 = P_1 + (Y_2 \times Y_2) \]
\[ R_1 = X_2 / P_2 \]
\[ R_2 = Y_1 / P_2 \]
\[ R_3 = Y_2 / P_3 \]
\[ U_1 = (U(J+2) \times R_2) + (V(J+2) \times R_3) \]
\[ U_1 = (U_1 \times W(J)) / TOT \]
\[ V_1 = (U(J+2) \times V(J+2)) \times R_1 \]
\[ V_1 = (V_1 \times W(J)) / TOT \]
\[ U_1 = (U(J+2) \times V(J+2)) \times R_3 \]
\[ U_1 = (V_1 \times W(J)) / TOT \]
\[ \text{SUMU} = \text{SUMU} + U_1 \]
\[ \text{SUMV} = \text{SUMV} + V_1 \]

0070 162 CONTINUE
0071 IF (MN.GT.0) GO TO 98
0073 UT(I) = SUMU
0074 VT(I) = SUMV
0075 X3 = (I-19) * 1.0
0076 IF (OT.GT.0) GO TO 161
0078 WRITE(7,15) X3,UT(I),VT(I)
0079 15 FORMAT(3F12.4)
0080 161 CONTINUE
0081 WRITE (7,20)
0082 20 FORMAT(/5X,'MODEL ERRORS',25X,'CP'/)
0083 95 SM = 0.0
0084 SE = 0.0
0085 SG = 0.0
0086 DO 164 JJ = 1,9
0087 X1 = OR - ((3-JJ)*AX)
0088 Y4 = -(IPP-JJ)*AH
0089 MN = 1
0090 GO TO 56
0091 98 AA2 = ATAN(SUMV/(1+SUMU))
0092 CP1 = 1.0 - ((1+SUMU)*(1+SUMU))
0093 IF (JJ.EQ.3) CP = CP1
0095 TOT = TOT + CP1
0096 IF (OT.GT.0) GO TO 104
0098 X1 = X1 - OR
0099 IF(CL1.EQ.0.) GO TO 104
0101 WRITE(7,25) X1, SUMU, SUMV, CP1
0102 25 FORMAT(4F12.4)
0103 104 IF (JJ.EQ.1) A1 = AA2
0105 IF (JJ.EQ.9) A2 = AA2
0107 SP = AA2 * (1-CS(JJ))
0108 IF (JJ.EQ.1 OR JJ.EQ.9) GO TO 102
0110 MB = JJ/2
0111 MB = MB*2
0112 IF (MB.EQ.JJ) GO TO 101
0114 SO = SO + SP
0115 GO TO 164
0116 101 SE = SE + SP
0117 GO TO 164
0118 102 SM = SM + SP
0119 164 CONTINUE
0120 CPE = TOT/9.0

Figure 4.6.2.


0121   P1 = SHF(2*50)+(4*SE)
0122   P2 = (AX/3)*P1
0123   P3 = 2*P2
0124   A3 = (A1-A2) * 59.29578
0125   IF(CL1.NE.0) WRITE(7,35)
0127  35  FORMAT(/10X,' EFFECT',25X,' DELTA CL')
0128   CL = TOP1 * A1
0129   A1 = A1 * 59.29578
0130   IF(CL1.EQ.0) GO TO 58
0132   WRITE(7,30) A1,CL
0133  30  FORMAT(/5X,'ALPHA ERROR = ',F8.4,' DEGREES',5X,F8.4)
0134   WRITE(7,40) A3,P3
0135  40  FORMAT(/5X,'INDUCED CAMBER = ',F8.4,' DEGREES',2X,F8.4)
0136   WRITE(7,45) CP
0137  45  FORMAT(/5X,'VEL. ERROR CP = ',F8.4)
0138   WRITE (7,55) CPE,-CP
0139  55  FORMAT(5X,' AVERAGE',6X,='F8.4,13X,F8.4///)
0140  58  IF(CL1.EQ.0) WRITE(5,57) A1,A3,CPE
0142  57  IF(CL1.EQ.0) WRITE (7,57) A1,A3,CPE
0144   ISC=0
0146   IF(CET.LE.01)ISC=ISC+1
0148   IF(CEB.LE.01)ISC=ISC+1
0150   IF(ABS(A1).LE.015)ISC=ISC+1
0152   IF(ABS(A3).LE.07)ISC=ISC+1
0154   IF(ABS(CPE).LE.007)ISC=ISC+1
0156   IF(ISC.EQ.5)PR1=1.
0158   IF(CL1.EQ.0.AND.PR1.EQ.1.) WRITE(5,800)
0160   IF(PR1.EQ.1.) WRITE (7,810)
0162  800  FORMAT(/6('*'),'TURN OFF WIND',6('*'))
0163  810  FORMAT(/9X,23('*')/9X,'# WALLS STREAMLINED #',9X,23('*'))
0164   RETURN
0165   END

SUME

Figure 4.6.3.
SUBROUTINE FORCE
COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/RN(30),RS(30),W(30),P(30),Q(30),RD(20)
COMMON/THREE/D(60),WTY(30),WBY(30),WL(30),B(50),PD(48,4)
COMMON/FOUR/E(30),H(30),G(30),WI(30),XI(30)
COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
DIMENSION A(50),BB(50),AJ(50),AG(50),AH(50)
REAL LS,L6,L7,J
INTEGER S
MT = MT+2
CALL ASSIGN(2,'WING.DAT',0,'OLD',,)
DEFINE FILE 2
READ(2'1) (AG(L),L=1,MT)
READ(2'2) (AJ(L),L=1,MT)
IHT = HT/2
WRITE(7,30)
30 FORMAT(/'  NACA SECTION ANALYSIS'/20X,' 0012-64'/
C  /
   ' RUN NO. = ')
WRITE(7,7) ITRN
WRITE(7,9)
9 FORMAT(/' $   ALPHA = ')
WRITE(7,12) AN
12 FORMAT('+',F6.2)
7 FORMAT('+',IS)
DO 54 K = 2,(MT-1)
54 AH(K) = (AJ(K+1)-AJ(K-1))/2
AH(1) = AJ(2)/2
AH(IHT+1) = AJ(IHT+2)/2
AH(IHT) = (.0012 - AJ(IHT-1))/2
AH(MT) = (.0012 - AJ(MT-1))/2
L5 =0
DO 3 IQ = 1,MT
3 IF (IQ.EQ.1) GO TO 540
IF (IQ.EQ.IHT) GO TO 580
IF (IQ.EQ.(IHT+1)) GO TO 630
IF (IQ.EQ.MT) GO TO 530
IF (IQ.GT.(IHT+1)) GO TO 780
WF = (AG(IQ+1)-AG(IQ-1))/2
L6 = WF*B(IQ)
L5= L5-L6
A(IQ) = -L6
GOTO 3
540 WF = AG(IQ+1)/2
L5 = -WF*B(IQ)
A(IQ) = L5
GOTO 3
580 WF = (1.0 - AG(IHT-1))/2
L6=WF*B(IQ)
L5=L5-L6
A(IQ)=-L6
CN = L5
GOTO 3
630 WF = AG(IQ+1)/2
L6 = WF*B(IQ)
Figure 4.7a.1.
L5 = L5 + L6
A(IO) = L6
GO TO 3
530 WF = (1.0 - AG(MT-1))/2
L6 = WF*B(MT)
L5 = L5 + L6
A(MT) = L6
CNL = L5 - CN
GO TO 3
780 WF = (AG(IO+1) - AG(IO-1))/2
L6 = WF*B(IO)
L5 = L5 + L6
A(IO) = L6
3 CONTINUE
890 C3 = 0
DO 40 IY = 1, MT
IF (IY.GT.(IHT+1)) GO TO 980
C5 = (B(IY)*AH(IY))
C5 = C5 + C5
GO TO 40
970 CCP = C3
GO TO 40
980 ST = C3 - CCP
GO TO 196
WRITE (7,165)
WRITE (7,194)
194 FORMAT(2X,'CHORD','3X,' CP LOCAL','3X,' CN LOCAL','3X,' CC LOCAL','3X,' CM LOCAL')
C9 = 0.
C8 = 0.
DO 5 S = 1, MT
IF (S.EQ.(IHT+1)) GO TO 1240
IF (S.GT.(IHT+1)) GO TO 1270
BB(S) = (-A(S)*AG(S) + AH(S)*AJ(S))
C9 = C9 + BB(S)
TIT = AG(S)*100
IF (CL1.EQ.0.) GO TO 5
WRITE (7,185) TIT, B(S), A(S), AH(S), BB(S)
GO TO 5
1240 IF (CL1.EQ.0.) GO TO 1270
WRITE (7,195)
195 FORMAT(//10X,'LOWER SURFACE')
WRITE (7,194)
1270 BB(S) = (A(S)*AG(S) + AH(S)*AJ(S))
C8 = C8 - BB(S)
TIT = AG(S)*100
IF (CL1.EQ.0.) GO TO 5

Figure 4.7a.2.
Figure 4.7a.3.
SUBROUTINE FORCE

COMMON/ONE/NJ,MT,NR,CL1,B1,PRI,AK1,AK3,AN,R3,PP2,ITRN
COMMON/TWO/R6(30),R6(30),W(30),X(30),P(30),G(30),D(20)
COMMON/THREE/B(60),WBY(30),WBY(30),W(30),V(30),PD(48,4)
COMMON/FOUR/E(30),H(30),Y(30),G(30),R(30),X(30)
COMMON/FIVE/U(30),V(30),D(60),DET(60),CS(20)

DIMENSION A(50),B(50),AJ(50),AG(50),AH(50)
REAL L5,L6,L7,J
INTEGER S

CALL ASSIGN(2,'NPL.DAT',0,'OLD',,
DEFINE FILE 2 (50,128,U,INR)
READ(2,1) (AG(L),L=1,MT)
READ(2,2) (AJ(L),L=1,MT)
IHT = 32
IF(CL1.EQ.0.) GO TO 31
WRITE(7,30)
FORMAT(11115X,'NPL SECTION ANALYSIS'/20X,'RUN NO. = ',9510'/
C /'
WRITE(7,7) ITRN
WRITE(7,9)
WRITE(7,12) AN
WRITE(7,14) B(31)=(B(30)+B(32))/2.
DO 54 K = 2,(MT-1)
AH(K) = (AJ(K+1)-AJ(K-1))/2.
AH(1) = AJ(2)/2
AH(IHT) = A(0.0024)
AH(IHT) = (.0024)
L5 =0
DO 3 IQ = 1,MT
IF (IQ.EQ.1) GO TO 540
IF (IQ.EQ.IHT) GO TO 580
IF (IQ.EQ.(IHT-1)) GO TO 630
IF (IQ.EQ.MT) GO TO 530
IF (IQ.GT.(IHT+1)) GO TO 780
WF = (AG(IQ+1)-AG(IQ-1))/2
L6 = WF*B(IQ)
L5 = L5-L6
A(IQ) = L6
GO TO 3
L6 = WF*B(IQ)
L5 = L5-L6
A(IQ) = L5
GO TO 3
L6 = WF*B(IQ)
L5 = L5-L6
A(IQ) = L5
CN = L5
GO TO 3
L6 = WF*B(IQ)
L5 = L5-L6
A(IQ) = L5
GO TO 3
L6 = WF*B(IQ)
L5 = L5-L6
A(IQ) = L5
CN = L5
GO TO 3
L6 = WF*B(IO)
L5 = L5 + L6
A(IO) = L6
GO TO 3

WF = .5-(AG(MT-1)/2)
L6 = WF*B(MT)
L5 = L5 + L6
A(MT) = L6
CNL = L5 - CN
GO TO 3

WF = (AG(IO+1)-AG(IO-1))/2
L6 = WF*B(IO)
L5 = L5 + L6
A(IO) = L6
GO TO 3

CONTINUE

WF = .5-(AG(MT-1)/2)
L6 = WF*B(MT)
L5 = L5 + L6
A(MT) = L6
GO TO 3

CONTINUE

C3 = 0
DO 40 IY = 1,MT
IF (IY.GT.(IHT+1)) GO TO 980
C5 = (B(IY)*AH(IY))
AH(IY) = C5
C3 = C3 + C5
GO TO 40

CCP = C3
GO TO 40

C5 = (B(IY)*AH(IY))
AH(IY) = C5
C3 = C3 + C5
CONTINUE

C9 = 0.
C8 = 0.
DO 5 S = 1, MT
IF (S.EQ.(IHT+1)) GO TO 1240
IF (S.GT.(IHT+1)) GO TO 1270
BB(S) = (-A(S)*AG(S)+(AH(S)*AJ(S))
C9 = C9 + BB(S)
TIT = AG(S)*100
IF (CL1.EQ.0.) GO TO 5
WRITE(7,185) TIT, B(S), A(S), AH(S), BB(S)
GO TO 5

IF (CL1.EQ.0.) GO TO 1270
WRITE(7,195)

FORMAT(/ // // , 10X,’ LOWER SURFACE’)
WRITE (7,194)

1270 BB(S) = (A(S)*AG(S)+(AH(S)*AJ(S))
C8 = C8 - BB(S)

TIT = AG(S)*100

Figure 4.7b.2.
0120   IF(CL1.EQ.0.) GO TO 5
0122   WRITE (7,185) TIT,B(S),A(S),AH(S),-BB(S)
0123  185  FORMAT(4X,F4.1,4F12.4)
0124   5   CONTINUE
0125   ST1 = C8+C9
0126   IF(CL1.EQ.0) GO TO 209
0128   WRITE(7,205)
0129  205  FORMAT(/18X,'PRESSURE',2X,'SUCTION',2X,'TOTAL'/)
0130   WRITE(7,206)CN,CNL,L5
0131  206  FORMAT(12X,'CN',2X,3F9.4)
0132   WRITE(7,207)CCP,ST,C3
0133  207  FORMAT(12X,'CC',2X,3F9.4)
0134   WRITE(7,208)C9,CS,ST1
0135  208  FORMAT(12X,'CM',2X,3F9.4)
0136  209  AT = AN * 0.017453
0137   CS1 = COS(AT)
0138   SN = SIN(AT)
0139   CL = (L5*CS1)-(C3*SN)
0140   CD = (C3*CS1) + (L5*SN)
0141   WRITE (7,225)
0142  225  FORMAT(22X,'WING PERFORMANCE'/18X,'CL',10X,'CD',10X,'CM')
0143   WRITE (7,235) CL,CD,ST1
0144  235  FORMAT(10X,3F12.4)
0145   CALL CLOSE(2)
0146   END

FORCE

Figure 4.7b.3.
SUBROUTINE SET

DIMENSION DA(100)

COMMON/DONE/NJ, HT, NR, CL1, B1, PR1, AK1, AK3, A3, R3, PP2, ITRN

COMMON/TWO/KN(30), RS(30), U(30), X(30), P(40), Q(40), RD(20)

COMMON/THREE/R(60), WT(30), WB(30), WL(30), B(50), PD(48, 4)

COMMON/FOUR/E(30), H(30), Y(30), G(30), WC(30), X(30), PC(40), QC(40), RD(20)

COMMON/FIVE/U(30), V(30), DSC(60), DET(60), CS(20)

COMMON/SEVEN/PC(50)

COMMON/EIGHT/TAB, AMP, IFN, IAM

DATA Y(1), Y(2), G(1), G(2), Y(23), G(23), G(24)/0, 0, 0, 0, 0 /

CALL ASSIGN(3, 'PAD.DAT', 0, 'OLD', )

DEFINE FILE 3 (50, 512, U, INR)

CALL ASSIGN(2, 'NPL.DAT', 0, 'OLD', )

DEFINE FILE 2 (50, 128, U, INR)

DATA OUTPUT

NJ1 = NJ-4

IWT=1

IF(CL1.EQ.0.) GO TO 12

WRITE(7, 65)

FORMAT('/' TRANSDUCER OUTPUT'/' CP VALUES CHANNELS 2-4'/

7X', '1', 7X', '2', 7X', '3', 7X', '4')

WRITE(7, 70) (N, PD(N, J), J=1, 4), N=1, 48)

WRITE(7, 41) ITRN

WRITE(7, 41) ITRN

FORMAT('/' RUN', 'I4', ' OUTPUT'/12X,' EXT VEL.', '7X', ' MOVEMENT',

8X', 'Y CO-ORD')

DO 45 J = 1, NJ

IF(J.LT.3 OR J.GT.22) GO TO 63

WMT = -WTY(J-2)

WMB = -WB(30, 31)

WRITE(7, 50) D(J), E(J), H(J), Y(J), G(J), WMT, WMB

CONTINUE

WRITE(7, 300) ITRN

FORMAT('/' RUN', 'I4', ' DATA'/15X, ' TOP WALL')

WRITE(7, 320)

FORMAT('/'5X, 'JACK', '5X, 'MACH NO.' )

THN=0

BMN=0

CALL ASSIGN(6, 'WALM.DAT', 0, 'OLD', )

DEFINE FILE 6 (20, 256, U, INR)

WRITE(7, 21)

WRITE(7, 22) ITRN

READ(7, 22) ITRN

FORMAT(12X)

DO 23 J=2, 21

X(J)=P(J)

X(J+21)=Q(J)

Figure 4.8.1.
CONTINUE
WRITE(6,IMR)(X(J),J=1,42)
CALL CLOSE(6)
GMN=0
DO 170 J = 1,NJ1
   IS1 = RS(J) + ((Y(J+2))*PC(J))
   DA(J)=IS1
   IP = RS(J)
   YM=P(J+1)
   WM = -WTY(J)-DET(J)
   IF (DA(16).LT.123.) DA(16)=123.
   IF (DA(17).LT.178.) DA(17)=178.
   IF (DA(18).LT.161.) DA(18)=161.
   IF (DA(19).LT.140.) DA(19)=140.
   IF (DA(20).LT.216.) DA(20)=216.
   IF(CL1.EQ.0.)GO TO 170
WRITE(7,840)J,WM,YM,Y(J+2),IP,IS1
FORMAT(7X,I2,11X,3F11.4,2(7X,I3))
IF(J.LT.5.OR.J.GT.13)GO TO 170
IBJ=J+20
WRITE(7,810)P(IBJ)
FORMAT(13X,F11.4)
CONTINUE
IF(CL1.GT.0.) WRITE(7,20)
FORMAT(11115X,'BOTTOM WALL')
DO 180 J = 1,NJ1
   IS1 = RN(J) - ((G(J+2))*PC(J+NJ1))
   DA(J+NJ1)=IS1
   IP = RN(J)
   YM=-WBY(J)+DET(J+NJ1)
   NIJ=J+16
   YM=Q(J+1)
   IF (DA(36).LT.209.) DA(36)=209.
   IF (DA(37).LT.143.) DA(37)=143.
   IF (DA(38).LT.136.) DA(38)=136.
   IF (DA(39).LT.183.) DA(39)=183.
   IF (DA(40).LT.270.) DA(40)=270.
   IF(CL1.EQ.0.)GO TO 180
WRITE(7,840)J,WM,YM,G(J+2),IP,IS1
GO TO 180
IF(J.GT.19)GO TO 180
AMN=(P(J+1)-Q(J+1))/2
GMN=GMN+(AMN*AMN)
TMN=TMN+(P(J+1)*P(J+1))
BMN=BMN+(Q(J+1)*Q(J+1))
IF(J.LT.5.OR.J.GT.13)GO TO 180
IBJ=J+20
WRITE(7,810)Q(IBJ)
CONTINUE
KT ::= 2*NJ1
DO 190 J = 1,NJ
   DA(J+KT) = E(J)
   KS = KT + NJ
DO 200 J = 1,NJ

Figure 4.8.2.
0121   200   DA(J+KS) = H(J)
0122   IF(CL1.EQ.0.)WRITE(5,30) IFN
0124   WRITE(7,30) IFN
0125   30   FORMAT(/' WALL & MODEL OUTPUT RECORD NO. = ',14//50(=',')//)
0126   35   FORMAT(I2)
0127   ND = 2 * (NJ+NJ1)
0128   WRITE(3'IFN) (DA(J),J=1,ND)
0129   MT2 = MT + 2
0130   IF(IWT.EQ.1) MT2=MT
0132   WRITE(2'IFN) (B(J),J=1,MT2)
0133   CALL CLOSE(2)
0134   CALL CLOSE(3)
0135   RETURN
0136   END

SET
SUBROUTINE WALL

ONLINE WALL EXERCISER

COMMON/EIGHT/TAB,AMP,IFN,IAM
NWJ=40
DIMENSION PC(50),IM(50),ADC(50),PV(50)
DIMENSION PL(10)
DATA PL/130.,178.,161.,140.,216.,209.,143.,136.,183.,270. /
CALL ASSIGNC3,'PAD.DAT',0,'OLD',)
DEFINE FILE 3 (50,512,U,INR)
READ(3,IFN) (PV(J),J=1,NWJ)
CALL CLOSE(3)

CALCULATE MOVEMENT

IF=1
IP=5
CALL INIT
50
IU=0
ICNT=0
IS=0
ITOL=3
DO 25 L=1,NWJ
IAI=L+23
IJP=CVSWG(IADC(IAI))
IPV=IJP*24414
IMOVE=ADC(L)-IPV
ADC(L)=IPV
IF(IPV.GE.980) WRITE(5,200)
200 FORMAT(/' POT LIMIT REACHED ON JACK ',14)
IF(IPV.GE.980) STOP
IF(L.LE.15) GO TO 60
IF(L.GT.20.AND.L.LE.35) GO TO 60
IF(L.EQ.36) IW=15
NDJ=L-(15+IW)
IF(PV(L).LT.PL(NDJ)) PV(L)=PL(NDJ)
IPD=PV(L)-IP
IMV=IPV-IPD
IF(IS.GE.39.AND.ICNT.EQ.6) GO TO 500
IF(IABS(IMV).GT.10) GO TO 500
IM(L)=IMV
IF(IABS(IMV).LE.ITOL) GO TO 40
25 CONTINUE
STOP
Figure 4.9.1.
GO TO 35

IS=IS+1

IMV=0

CALL MOVE(IJ,IMV)

CONTINUE

IF=0

IF(IS.EQ.40) GO TO 70

CALL START(PV,IM)

IF(IS.LT.40) GO TO 50

IF(IP.EQ.0) GO TO 100

IF=0

GO TO 50

100 IW=0

DO 85 J=16,40

IF(J.GT.20.AND.J.LT.36) GO TO 85

IF(J.EQ.36)IW=15

ILIM=J-(15+IW)

IF(IABS(IK).LE.ITOL) WRITE(5,80) J

80 FORMAT(' LOWER POT LIMIT REACHED ON JACK ',I4)

CONTINUE

GO TO 250

WRITE(5,150) L

150 FORMAT(/' JACK ',I4,' JAMMED!!!!!!!!!'

GO TO 250

WRITE(5,160) L

160 FORMAT(//' WALL OUT OF CONTROL AT JACK ',I4/40('*'

RETURN

END

SUBROUTINE INIT

INITIALISE TSWT CONTROL SYSTEM

NWJ=40

IDI=128

go 10 j=1,NWJ

ICOM=J+IDI+0

CALL DIO(ICOM,IDI,INPUT)

IF(INPUT.NE.0) WRITE(5,30) J

30 FORMAT(/' JACK ',I4,' I/O ERROR'

CONTINUE

RETURN

END

INIT

Figure 4.9.2.
SUBROUTINE START(PV,IM)
DIMENSION PV(50),IM(50)
ISUM=0
DO 10 J=21,40
10 ISUM=ISUM+ABS(IM(J))
IAM=ISUM/20
RCT=(IAM/2)*25.
IF(RCT.GT.600.) RCT=600.
IF(RCT.LT.25.) RCT=25.
IF(RCT.GT.50.) CALL DIO(129,128,INPUT)
IF(RCT.GT.50.) CALL DIO(149,128,INPUT)
START
ALL
MOTORS
IPSA=45
IDI=0
ICOM=IPSA+IDI+4096
CALL DIO(ICOM,IDI,INPUT)
WAIT
FOR
WALL
TO
MOVE
IEND=0
CALL SETR(5,0,RCT,IEND)
CALL LWAIT(0,IEND)
STOP
ALL
MOTORS
CALL DIO(45,IDI,INPUT)
RETURN
END

SUBROUTINE MOVE(IJ,IMV)
IF(IMV.GT.0) 15,10,5
IF(IDR.GT.3072) GO TO 20
IDR=0
GO TO 20
IDR=2560
IDI=128
ICOM=IJ+IDI+IDR
LOAD
COMMAND
CALL DIO(ICOM,IDI,INPUT)
RETURN
END

SUBROUTINE DIOCICOM,IDI,INPUT)
CALL IPOKEC"167774,45)
CALL IPOKEC"167774,ICOM)
IF(IPEEKC"167770).GE.0) GO TO 10
CALL IPOKEC"167770,0)
IF(IDI.EQ.0) GO TO 20
IF(IPEEKC"167770).NE.128) GO TO 30
INPUT=IPEEKC"167772)
CALL IPOKEC"167770,0)
RETURN
END

Figure 4.9.3.
0001 C PROGRAM REAN
0002 C MAIN CONTROL PROGRAM
0003 COMMON/ONE/NJ,MT,NR,CL1,B1,PR1,AK1,AK3,AN,R3,PP2,ITRN
0004 COMMON/TWO/RN(30),RS(30),W(30),X(30),P(40),Q(40),RD(20)
0005 COMMON/THREE/B(60),WBY(30),WY(30),WL(30),B(50),PD(48,4)
0006 COMMON/FOUR/E(30),Y(30),G(30),WI(30),XI(30)
0007 COMMON/FIVE/U(30),V(30),DS(60),DET(60),CS(20)
0008 COMMON/SEVEN/PC(50)
0009 COMMON/EIGHT/TAB,AMP,IFN,IAM
0010 CL1 = 500.
0011 NR = 8
0012 WRITE(7,5)
0013 FORMAT(16X,' UNIVERSITY OF SOUTHAMPTON'/5X,
0014 C ' TRANSONIC SELF-STREAMLINING WIND TUNNEL'/20X,8('**'))
0015 WRITE(7,10)
0016 FORMAT(17X,' DATA REANALYSIS'/17X,15('*'))
0017 CALL IDATE(I,J,K)
0018 WRIT(7,15) J,I,K
0019 FORMAT(19X,I2,2('-',I2))
0020 WRITE(7,40)
0021 FORMAT(10X,' RUN NO. = ')  
0022 READ(7,45) ITRN
0023 FORMAT(14)
0024 WRITE(7,110)
0025 FORMAT(10X,' FILE NO. = ')  
0026 READ(7,45) IFN
0027 FORMAT(50)
0028 WRITE(7,50)
0029 FORMAT(15X,' MODEL ALPHA (DEG) = ')  
0030 READ(7,60) AN
0031 FORMAT(10X,' NO. OF MODEL TAPS ? ')  
0032 READ(7,65) MT
0033 WRITE(7,80)
0034 FORMAT(10X,' INPUT 1 FOR AUTO IN-FILE SELECTION -')  
0035 READ(7,45) IAM
0036 WRITE(7,85)
0037 FORMAT(10X,' INPUT AMBIENT CONDITIONS'/10X,' TEMP (DEG.C) = ')  
0038 READ(7,25) TAB
0039 FORMAT(10X,' PRES. (CM HG) = ')  
0040 READ(7,25) AMP
0041 FORMAT(10X,' CALL DATA
0042 CALL REDUCE
0043 CALL WAS
0044 CALL STAR
0045 CALL SUME
0046 IF(MT.EQ.0) GO TO 100
0047 CALL FORCE
0048 CALL SET
0049 END

Figure 4.10.1.
0001 PROGRAM RUN
0002 DIMENSION TUN(5)
0003 CALL ASSIGN(2, 'RUN.DAT', 0, 'OLD', )
0004 DEFINE FILE 2 (6, 256, U, IJR)
0005 READ(2, 1)(TUN(J), J=1, 5)
0006 WRITE(5, 25)
0007 25 FORMAT('INPUT 1 FOR RECORD NO. INCREMENT ')
0008 READ(5, 35) INC
0009 35 FORMAT(I6)
0010 IAD = CVSWG(IADC(21))
0011 TUN(3) = (IAD - 2045) / 8.15
0012 IF(INC, EQ, 1) GO TO 20
0014 WRITE(5, 5)
0015 5 FORMAT('GOOD MORNING'/'INPUT PRES.,')
0016 READ(5, 45) TUN(4)
0017 45 FORMAT(F8.4)
0018 WRITE(5, 15)
0019 15 FORMAT('INPUT NEXT RUN NO. & FILE NO.,')
0020 READ(5, 10) IR, IF
0021 10 FORMAT(2I6)
0022 TUN(1) = IR
0023 TUN(2) = IF
0024 GO TO 30
0025 20 TUN(2) = TUN(2) + 1
0026 30 WRITE(2, 1)(TUN(J), J=1, 5)
0027 WRITE(5, 55)(TUN(J), J=1, 5)
0028 55 FORMAT('RUN DATA STORED '/5F8.2)
0029 CALL CLOSE(2)
0030 END

RUN

Figure 4.11.1.
Figure 5 A sample output of the control software OFLEX
UNIVERSITY OF SOUTHAMPTON
TRANSONIC SELF-STREAMLINING WIND TUNNEL
*********

AUTO MODE
22-1-82

RUN NO. = 407

MODEL ALPHA (DEG) = 4.0

AMBIENT CONDITIONS
TEMP = 22.25   PRES(CM HG) = 76.65

ITERATION RECORD NO. = 31

MACH NO. = 0.5945

REYNOLDS NO. = 0.118538E+07

RECORD = 30

WALL CP ERROR
TOP = 0.0061   BOTTOM = -0.0055
RESIDUALS = 0.0287 -0.0042 0.0014

WALL & MODEL OUTPUT RECORD NO. = 31

=================================================================

ITERATION RECORD NO. = 32

MACH NO. = 0.5948

REYNOLDS NO. = 0.118590E+07

RECORD = 31

WALL CP ERROR
TOP = 0.0026   BOTTOM = 0.0038
RESIDUALS = -0.0163 0.0051 0.0000

WALL & MODEL OUTPUT RECORD NO. = 32

=================================================================

Figure 5.1.
ITERATION RECORD NO. = 33

MACH NO. = 0.5966

REYNOLDS NO. = 0.118846E+07

RECORD = 32

WALL CP ERROR
TOP = 0.0029
BOTTOM = 0.0028
RESIDUALS = -0.0066 0.0101 -0.0017

*****************************************************************************
* WALLS STREAMLINED *
*****************************************************************************

WALL & MODEL OUTPUT RECORD NO. = 33

==============

Figure 5.2.
Figure 6. A sample output of the re-analysis software ORLEX
RUN ORLEX
UNIVERSITY OF SOUTHAMPTON
TRANSONIC SELF-STREAMLINING WIND TUNNEL

*********

DATA REANALYSIS
***************
11-2-82

RUN NO. = 389
FILE NO. = 21
MODEL ALPHA (DEG) = 3.0
NO. OF MODEL TAPS ? 50
INPUT 1 FOR AUTO IN-FILE SELECTION -1

INPUT AMBIENT CONDITIONS
TEMP (DEG.C) = 22.0
PRES. (CM HG) = 76.32

DATA INPUT FILE = *ADC.DAT
*****************************************************************************
ITERATION RECORD NO. = 21
*****************************************************************************

MACH NO. = 0.8038

REYNOLDS NO. = 0.143046E+07

FILE NO. ?
1 FOR M=<.725
2 FOR .725<M>.825
3 FOR M>=.825 ANS = 2

WALL CONTOURS RECORD = 20

WAS COMPUTING NOW!!
DELTA STAR CALCS.

TOP WALL

<table>
<thead>
<tr>
<th>TAP NO.</th>
<th>DU/DX</th>
<th>MACH NO.</th>
<th>D*</th>
<th>DD*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.3870</td>
<td>0.8238</td>
<td>0.0155</td>
<td>-0.0005</td>
</tr>
<tr>
<td>2</td>
<td>-3.0340</td>
<td>0.8057</td>
<td>0.0227</td>
<td>-0.0004</td>
</tr>
<tr>
<td>3</td>
<td>0.6563</td>
<td>0.7998</td>
<td>0.0290</td>
<td>-0.0000</td>
</tr>
<tr>
<td>4</td>
<td>0.1942</td>
<td>0.8092</td>
<td>0.0344</td>
<td>-0.0009</td>
</tr>
<tr>
<td>5</td>
<td>1.9132</td>
<td>0.8012</td>
<td>0.0395</td>
<td>-0.0012</td>
</tr>
<tr>
<td>6</td>
<td>14.1533</td>
<td>0.8127</td>
<td>0.0403</td>
<td>-0.0042</td>
</tr>
<tr>
<td>7</td>
<td>32.5273</td>
<td>0.8366</td>
<td>0.0386</td>
<td>-0.0076</td>
</tr>
<tr>
<td>8</td>
<td>54.5744</td>
<td>0.8975</td>
<td>0.0349</td>
<td>-0.0130</td>
</tr>
<tr>
<td>9</td>
<td>55.2920</td>
<td>0.9884</td>
<td>0.0302</td>
<td>-0.0189</td>
</tr>
<tr>
<td>10</td>
<td>-2.2115</td>
<td>1.0633</td>
<td>0.0294</td>
<td>-0.0218</td>
</tr>
<tr>
<td>11</td>
<td>-53.4752</td>
<td>0.9817</td>
<td>0.0344</td>
<td>-0.0189</td>
</tr>
<tr>
<td>12</td>
<td>-37.5962</td>
<td>0.9032</td>
<td>0.0424</td>
<td>-0.0130</td>
</tr>
<tr>
<td>13</td>
<td>-17.6038</td>
<td>0.8758</td>
<td>0.0483</td>
<td>-0.0088</td>
</tr>
<tr>
<td>14</td>
<td>-10.6314</td>
<td>0.8436</td>
<td>0.0571</td>
<td>-0.0038</td>
</tr>
<tr>
<td>15</td>
<td>-4.2687</td>
<td>0.8159</td>
<td>0.0672</td>
<td>0.0013</td>
</tr>
<tr>
<td>16</td>
<td>-0.3833</td>
<td>0.8144</td>
<td>0.0735</td>
<td>0.0029</td>
</tr>
<tr>
<td>17</td>
<td>0.1915</td>
<td>0.8130</td>
<td>0.0781</td>
<td>0.0033</td>
</tr>
<tr>
<td>18</td>
<td>0.5746</td>
<td>0.8159</td>
<td>0.0823</td>
<td>0.0029</td>
</tr>
<tr>
<td>19</td>
<td>0.1913</td>
<td>0.8173</td>
<td>0.0861</td>
<td>0.0021</td>
</tr>
<tr>
<td>20</td>
<td>0.0000</td>
<td>0.8173</td>
<td>0.0907</td>
<td>0.0021</td>
</tr>
</tbody>
</table>

BOTTOM WALL

<table>
<thead>
<tr>
<th>TAP NO.</th>
<th>DU/DX</th>
<th>MACH NO.</th>
<th>D*</th>
<th>DD*</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.4216</td>
<td>0.8077</td>
<td>0.0155</td>
<td>-0.0009</td>
</tr>
<tr>
<td>2</td>
<td>-1.1125</td>
<td>0.8060</td>
<td>0.0223</td>
<td>-0.0004</td>
</tr>
<tr>
<td>3</td>
<td>-1.5025</td>
<td>0.7981</td>
<td>0.0290</td>
<td>-0.0000</td>
</tr>
<tr>
<td>4</td>
<td>-3.5534</td>
<td>0.7935</td>
<td>0.0353</td>
<td>0.0004</td>
</tr>
<tr>
<td>5</td>
<td>-7.0914</td>
<td>0.7721</td>
<td>0.0437</td>
<td>0.0025</td>
</tr>
<tr>
<td>6</td>
<td>-12.0694</td>
<td>0.7533</td>
<td>0.0508</td>
<td>0.0063</td>
</tr>
<tr>
<td>7</td>
<td>-3.1566</td>
<td>0.7367</td>
<td>0.0542</td>
<td>0.0084</td>
</tr>
<tr>
<td>8</td>
<td>8.0421</td>
<td>0.7459</td>
<td>0.0550</td>
<td>0.0075</td>
</tr>
<tr>
<td>9</td>
<td>7.1233</td>
<td>0.7563</td>
<td>0.0550</td>
<td>0.0059</td>
</tr>
<tr>
<td>10</td>
<td>6.3412</td>
<td>0.7634</td>
<td>0.0554</td>
<td>0.0046</td>
</tr>
<tr>
<td>11</td>
<td>-3.0033</td>
<td>0.7714</td>
<td>0.0567</td>
<td>0.0042</td>
</tr>
<tr>
<td>12</td>
<td>-8.1537</td>
<td>0.7563</td>
<td>0.0596</td>
<td>0.0058</td>
</tr>
<tr>
<td>13</td>
<td>0.6964</td>
<td>0.7520</td>
<td>0.0622</td>
<td>0.0063</td>
</tr>
<tr>
<td>14</td>
<td>8.0705</td>
<td>0.7742</td>
<td>0.0630</td>
<td>0.0025</td>
</tr>
<tr>
<td>15</td>
<td>3.0736</td>
<td>0.8001</td>
<td>0.0634</td>
<td>-0.0017</td>
</tr>
<tr>
<td>16</td>
<td>1.9875</td>
<td>0.8001</td>
<td>0.0664</td>
<td>-0.0037</td>
</tr>
<tr>
<td>17</td>
<td>1.6074</td>
<td>0.8149</td>
<td>0.0697</td>
<td>-0.0055</td>
</tr>
<tr>
<td>18</td>
<td>-2.0260</td>
<td>0.8121</td>
<td>0.0748</td>
<td>-0.0054</td>
</tr>
<tr>
<td>19</td>
<td>-1.6459</td>
<td>0.7998</td>
<td>0.0806</td>
<td>-0.0051</td>
</tr>
<tr>
<td>20</td>
<td>0.0000</td>
<td>0.7998</td>
<td>0.0861</td>
<td>-0.0046</td>
</tr>
</tbody>
</table>

UNIT REYNOLDS NO. = 357276.9 D* FPG = 0.0088
### WALL CP ERROR

**TOP** - 0.0072  
**BOTTOM** - 0.0064

#### RESIDUAL ERRORS

<table>
<thead>
<tr>
<th>X</th>
<th>U/UFS</th>
<th>V/UFS</th>
</tr>
</thead>
<tbody>
<tr>
<td>-18.0000</td>
<td>0.0009</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-17.0000</td>
<td>0.0009</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-16.0000</td>
<td>0.0008</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-15.0000</td>
<td>0.0007</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-14.0000</td>
<td>0.0006</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-13.0000</td>
<td>0.0004</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-12.0000</td>
<td>0.0003</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-11.0000</td>
<td>0.0002</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-10.0000</td>
<td>0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-9.0000</td>
<td>0.0000</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-8.0000</td>
<td>-0.0001</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-7.0000</td>
<td>-0.0002</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-6.0000</td>
<td>-0.0003</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-5.0000</td>
<td>-0.0004</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-4.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-3.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-2.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>-1.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>1.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>2.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>3.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>4.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>5.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>6.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>7.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>8.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>9.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>10.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>11.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>12.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>13.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>14.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>15.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>16.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>17.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
<tr>
<td>18.0000</td>
<td>-0.0005</td>
<td>-0.0000</td>
</tr>
</tbody>
</table>

#### MODEL ERRORS

<table>
<thead>
<tr>
<th>CP</th>
<th>EFFECT</th>
<th>DELTA CL</th>
</tr>
</thead>
<tbody>
<tr>
<td>-1.4979</td>
<td>0.0017</td>
<td>-0.0034</td>
</tr>
<tr>
<td>-0.7490</td>
<td>0.0022</td>
<td>-0.0044</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0024</td>
<td>-0.0048</td>
</tr>
<tr>
<td>0.7490</td>
<td>0.0022</td>
<td>-0.0045</td>
</tr>
<tr>
<td>1.4979</td>
<td>0.0018</td>
<td>-0.0036</td>
</tr>
<tr>
<td>2.2469</td>
<td>0.0013</td>
<td>-0.0025</td>
</tr>
<tr>
<td>2.9959</td>
<td>0.0008</td>
<td>-0.0016</td>
</tr>
<tr>
<td>3.7449</td>
<td>0.0005</td>
<td>-0.0010</td>
</tr>
<tr>
<td>4.4938</td>
<td>0.0004</td>
<td>-0.0007</td>
</tr>
</tbody>
</table>

**EFFECT**

**ALPHA ERROR** = -0.0045 DEGREES  
**INDUCED CAMBER** = 0.0007 DEGREES  
**VEL. ERROR CP** = -0.0048  
**AVERAGE** = -0.0029

---

Figure 6.3.
### NPL SECTION ANALYSIS

**9510**

**RUN NO. = 389**

**ALPHA = 3.00**

#### UPPER SURFACE

<table>
<thead>
<tr>
<th>ZCHORD</th>
<th>CP LOCAL</th>
<th>CN LOCAL</th>
<th>CC LOCAL</th>
<th>CM LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0227</td>
<td>-0.0038</td>
<td>0.0079</td>
<td>0.0000</td>
</tr>
<tr>
<td>0.7</td>
<td>-0.7242</td>
<td>0.0036</td>
<td>-0.0062</td>
<td>-0.0001</td>
</tr>
<tr>
<td>1.0</td>
<td>-1.0257</td>
<td>0.0046</td>
<td>-0.0024</td>
<td>-0.0001</td>
</tr>
<tr>
<td>1.6</td>
<td>-1.2861</td>
<td>0.0064</td>
<td>-0.0027</td>
<td>-0.0002</td>
</tr>
<tr>
<td>2.0</td>
<td>-1.2979</td>
<td>0.0066</td>
<td>-0.0019</td>
<td>-0.0002</td>
</tr>
<tr>
<td>2.6</td>
<td>-1.3171</td>
<td>0.0070</td>
<td>-0.0016</td>
<td>-0.0002</td>
</tr>
<tr>
<td>3.1</td>
<td>-1.3824</td>
<td>0.0068</td>
<td>-0.0016</td>
<td>-0.0002</td>
</tr>
<tr>
<td>3.6</td>
<td>-1.3838</td>
<td>0.0146</td>
<td>-0.0026</td>
<td>-0.0006</td>
</tr>
<tr>
<td>5.2</td>
<td>-1.3318</td>
<td>0.0270</td>
<td>-0.0041</td>
<td>-0.0015</td>
</tr>
<tr>
<td>7.7</td>
<td>-1.2563</td>
<td>0.0314</td>
<td>-0.0038</td>
<td>-0.0025</td>
</tr>
<tr>
<td>10.2</td>
<td>-1.1720</td>
<td>0.0439</td>
<td>-0.0040</td>
<td>-0.0046</td>
</tr>
<tr>
<td>15.2</td>
<td>-1.1809</td>
<td>0.0591</td>
<td>-0.0044</td>
<td>-0.0091</td>
</tr>
<tr>
<td>20.2</td>
<td>-1.1601</td>
<td>0.0580</td>
<td>-0.0035</td>
<td>-0.0119</td>
</tr>
<tr>
<td>25.2</td>
<td>-1.1279</td>
<td>0.0563</td>
<td>-0.0028</td>
<td>-0.0143</td>
</tr>
<tr>
<td>30.2</td>
<td>-1.1249</td>
<td>0.0844</td>
<td>-0.0031</td>
<td>-0.0256</td>
</tr>
<tr>
<td>40.2</td>
<td>-1.0392</td>
<td>0.0779</td>
<td>-0.0021</td>
<td>-0.0314</td>
</tr>
<tr>
<td>45.2</td>
<td>-0.7681</td>
<td>0.0383</td>
<td>-0.0004</td>
<td>-0.0173</td>
</tr>
<tr>
<td>50.2</td>
<td>-0.5044</td>
<td>0.0189</td>
<td>-0.0001</td>
<td>-0.0095</td>
</tr>
<tr>
<td>52.7</td>
<td>-0.4348</td>
<td>0.0109</td>
<td>0.0000</td>
<td>-0.0057</td>
</tr>
<tr>
<td>55.2</td>
<td>-0.3941</td>
<td>0.0099</td>
<td>0.0001</td>
<td>-0.0054</td>
</tr>
<tr>
<td>57.7</td>
<td>-0.3548</td>
<td>0.0089</td>
<td>0.0001</td>
<td>-0.0051</td>
</tr>
<tr>
<td>60.2</td>
<td>-0.3606</td>
<td>0.0090</td>
<td>0.0002</td>
<td>-0.0054</td>
</tr>
<tr>
<td>62.7</td>
<td>-0.3504</td>
<td>0.0087</td>
<td>0.0003</td>
<td>-0.0055</td>
</tr>
<tr>
<td>65.2</td>
<td>-0.2664</td>
<td>0.0067</td>
<td>0.0003</td>
<td>-0.0043</td>
</tr>
<tr>
<td>67.7</td>
<td>-0.2778</td>
<td>0.0069</td>
<td>0.0003</td>
<td>-0.0047</td>
</tr>
<tr>
<td>70.2</td>
<td>-0.2736</td>
<td>0.0102</td>
<td>0.0007</td>
<td>-0.0071</td>
</tr>
<tr>
<td>75.2</td>
<td>-0.2361</td>
<td>0.0119</td>
<td>0.0010</td>
<td>-0.0088</td>
</tr>
<tr>
<td>80.2</td>
<td>-0.1737</td>
<td>0.0088</td>
<td>0.0010</td>
<td>-0.0040</td>
</tr>
<tr>
<td>85.3</td>
<td>-0.0976</td>
<td>0.0049</td>
<td>0.0007</td>
<td>-0.0041</td>
</tr>
<tr>
<td>90.1</td>
<td>-0.0143</td>
<td>0.0007</td>
<td>0.0001</td>
<td>-0.0006</td>
</tr>
<tr>
<td>95.2</td>
<td>0.0626</td>
<td>-0.0031</td>
<td>-0.0007</td>
<td>0.0029</td>
</tr>
<tr>
<td>****</td>
<td>0.1394</td>
<td>-0.0034</td>
<td>0.0003</td>
<td>0.0034</td>
</tr>
</tbody>
</table>

#### LOWER SURFACE

<table>
<thead>
<tr>
<th>ZCHORD</th>
<th>CP LOCAL</th>
<th>CN LOCAL</th>
<th>CC LOCAL</th>
<th>CM LOCAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>1.0684</td>
<td>0.0053</td>
<td>0.0098</td>
<td>0.0000</td>
</tr>
<tr>
<td>1.0</td>
<td>0.8331</td>
<td>0.0074</td>
<td>0.0098</td>
<td>-0.0003</td>
</tr>
<tr>
<td>1.8</td>
<td>0.5949</td>
<td>0.0127</td>
<td>0.0059</td>
<td>-0.0004</td>
</tr>
<tr>
<td>5.3</td>
<td>0.1998</td>
<td>0.0085</td>
<td>0.0026</td>
<td>-0.0005</td>
</tr>
<tr>
<td>10.2</td>
<td>-0.0488</td>
<td>-0.0024</td>
<td>-0.0005</td>
<td>0.0003</td>
</tr>
<tr>
<td>15.3</td>
<td>-0.1493</td>
<td>-0.0074</td>
<td>-0.0009</td>
<td>0.0012</td>
</tr>
<tr>
<td>20.2</td>
<td>-0.2291</td>
<td>-0.0142</td>
<td>-0.0009</td>
<td>0.0029</td>
</tr>
<tr>
<td>27.7</td>
<td>-0.3163</td>
<td>-0.0275</td>
<td>-0.0000</td>
<td>0.0076</td>
</tr>
<tr>
<td>37.6</td>
<td>-0.3310</td>
<td>-0.0288</td>
<td>0.0017</td>
<td>0.0107</td>
</tr>
<tr>
<td>45.1</td>
<td>-0.2138</td>
<td>-0.0162</td>
<td>0.0020</td>
<td>0.0072</td>
</tr>
<tr>
<td>52.6</td>
<td>-0.0488</td>
<td>-0.0037</td>
<td>0.0006</td>
<td>0.0019</td>
</tr>
<tr>
<td>60.1</td>
<td>0.0988</td>
<td>0.0067</td>
<td>-0.0011</td>
<td>-0.0040</td>
</tr>
<tr>
<td>67.5</td>
<td>0.3073</td>
<td>0.0179</td>
<td>-0.0030</td>
<td>-0.0121</td>
</tr>
<tr>
<td>71.7</td>
<td>0.3629</td>
<td>0.0318</td>
<td>-0.0039</td>
<td>-0.0228</td>
</tr>
<tr>
<td>85.1</td>
<td>0.4490</td>
<td>0.0412</td>
<td>-0.0036</td>
<td>-0.0351</td>
</tr>
<tr>
<td>90.1</td>
<td>0.4430</td>
<td>0.0221</td>
<td>-0.0000</td>
<td>-0.0200</td>
</tr>
<tr>
<td>95.1</td>
<td>0.3825</td>
<td>0.0190</td>
<td>0.0005</td>
<td>-0.0180</td>
</tr>
<tr>
<td>****</td>
<td>0.1311</td>
<td>0.0032</td>
<td>0.0003</td>
<td>-0.0032</td>
</tr>
</tbody>
</table>

#### PRESSURE SUCTION TOTAL

| CN | 0.6319 | 0.0755 | 0.7075 |
| CC | -0.0350 | 0.0193 | -0.0157 |
| CH | -0.1870 | -0.0844 | -0.2715 |

**WING PERFORMANCE**

<table>
<thead>
<tr>
<th>CL</th>
<th>CD</th>
<th>CM</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7073</td>
<td>0.0214</td>
<td>-0.2715</td>
</tr>
</tbody>
</table>

**Figure 6.4.**
<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>26.5666</td>
<td>-0.0463</td>
<td>0.0891</td>
<td>-0.0107</td>
</tr>
<tr>
<td>2</td>
<td>3.6152</td>
<td>-0.0063</td>
<td>0.1375</td>
<td>-0.0069</td>
</tr>
<tr>
<td>3</td>
<td>42.8183</td>
<td>0.0038</td>
<td>0.1788</td>
<td>0.0075</td>
</tr>
<tr>
<td>4</td>
<td>49.5844</td>
<td>-0.0171</td>
<td>0.1154</td>
<td>0.0176</td>
</tr>
<tr>
<td>5</td>
<td>55.2891</td>
<td>-0.0006</td>
<td>0.1331</td>
<td>-0.0649</td>
</tr>
<tr>
<td>6</td>
<td>55.5544</td>
<td>-0.0248</td>
<td>0.1037</td>
<td>0.1066</td>
</tr>
<tr>
<td>7</td>
<td>26.5003</td>
<td>-0.0777</td>
<td>0.1504</td>
<td>0.1431</td>
</tr>
<tr>
<td>8</td>
<td>55.9856</td>
<td>-0.2113</td>
<td>0.1702</td>
<td>0.1230</td>
</tr>
<tr>
<td>9</td>
<td>57.4449</td>
<td>-0.4048</td>
<td>0.1594</td>
<td>0.1015</td>
</tr>
<tr>
<td>10</td>
<td>57.4781</td>
<td>-0.5595</td>
<td>0.8655</td>
<td>0.0858</td>
</tr>
<tr>
<td>11</td>
<td>56.3836</td>
<td>-0.3903</td>
<td>0.0904</td>
<td>0.0680</td>
</tr>
<tr>
<td>12</td>
<td>54.6921</td>
<td>-0.2219</td>
<td>0.0778</td>
<td>0.1014</td>
</tr>
<tr>
<td>13</td>
<td>26.5334</td>
<td>-0.1621</td>
<td>0.1490</td>
<td>0.1108</td>
</tr>
<tr>
<td>14</td>
<td>52.8015</td>
<td>-0.0915</td>
<td>0.1700</td>
<td>0.0617</td>
</tr>
<tr>
<td>15</td>
<td>53.0005</td>
<td>-0.0229</td>
<td>0.1365</td>
<td>0.0120</td>
</tr>
<tr>
<td>16</td>
<td>52.5362</td>
<td>-0.0197</td>
<td>-0.2664</td>
<td>0.0120</td>
</tr>
<tr>
<td>17</td>
<td>52.1382</td>
<td>-0.0164</td>
<td>-0.2778</td>
<td>-0.0207</td>
</tr>
<tr>
<td>18</td>
<td>52.0719</td>
<td>-0.0227</td>
<td>-0.2736</td>
<td>-0.0144</td>
</tr>
<tr>
<td>19</td>
<td>26.6993</td>
<td>-0.0259</td>
<td>-0.2361</td>
<td>0.0125</td>
</tr>
<tr>
<td>20</td>
<td>50.1150</td>
<td>0.0341</td>
<td>-0.1737</td>
<td>0.1953</td>
</tr>
<tr>
<td>21</td>
<td>43.9792</td>
<td>1.0152</td>
<td>-0.0976</td>
<td>-0.0094</td>
</tr>
<tr>
<td>22</td>
<td>38.0755</td>
<td>0.0032</td>
<td>-0.0143</td>
<td>0.0507</td>
</tr>
<tr>
<td>23</td>
<td>36.9146</td>
<td>-0.0306</td>
<td>1.1970</td>
<td>0.1238</td>
</tr>
<tr>
<td>24</td>
<td>35.9860</td>
<td>-0.1368</td>
<td>0.1394</td>
<td>0.1330</td>
</tr>
<tr>
<td>25</td>
<td>26.9978</td>
<td>1.0057</td>
<td>0.0876</td>
<td>0.0514</td>
</tr>
<tr>
<td>26</td>
<td>35.0905</td>
<td>-0.2630</td>
<td>0.1088</td>
<td>0.1102</td>
</tr>
<tr>
<td>27</td>
<td>35.2231</td>
<td>-0.5160</td>
<td>0.1518</td>
<td>0.0619</td>
</tr>
<tr>
<td>28</td>
<td>34.9910</td>
<td>-0.7459</td>
<td>0.1370</td>
<td>1.0440</td>
</tr>
<tr>
<td>29</td>
<td>2.5587</td>
<td>-0.2625</td>
<td>0.1574</td>
<td>0.0932</td>
</tr>
<tr>
<td>30</td>
<td>7.8605</td>
<td>-0.2378</td>
<td>0.1383</td>
<td>0.0945</td>
</tr>
<tr>
<td>31</td>
<td>26.5534</td>
<td>-0.2352</td>
<td>0.1550</td>
<td>0.0680</td>
</tr>
<tr>
<td>32</td>
<td>13.2004</td>
<td>-0.3198</td>
<td>0.1407</td>
<td>0.0623</td>
</tr>
<tr>
<td>33</td>
<td>22.0539</td>
<td>-0.1557</td>
<td>0.1401</td>
<td>0.1152</td>
</tr>
<tr>
<td>34</td>
<td>27.6279</td>
<td>-0.1036</td>
<td>0.1227</td>
<td>0.0355</td>
</tr>
<tr>
<td>35</td>
<td>29.9165</td>
<td>-0.0356</td>
<td>0.1471</td>
<td>0.0107</td>
</tr>
<tr>
<td>36</td>
<td>31.7075</td>
<td>-0.0254</td>
<td>0.1513</td>
<td>0.1195</td>
</tr>
<tr>
<td>37</td>
<td>26.5666</td>
<td>-0.0146</td>
<td>0.1070</td>
<td>-0.0069</td>
</tr>
<tr>
<td>38</td>
<td>33.6643</td>
<td>-0.0140</td>
<td>0.3073</td>
<td>-0.0075</td>
</tr>
<tr>
<td>39</td>
<td>33.9960</td>
<td>-0.0317</td>
<td>0.3629</td>
<td>0.0000</td>
</tr>
<tr>
<td>40</td>
<td>31.4090</td>
<td>0.0343</td>
<td>0.4490</td>
<td>-0.0289</td>
</tr>
<tr>
<td>41</td>
<td>27.6279</td>
<td>-0.0153</td>
<td>0.4430</td>
<td>0.0964</td>
</tr>
<tr>
<td>42</td>
<td>24.5434</td>
<td>-0.2015</td>
<td>0.3825</td>
<td>0.1291</td>
</tr>
<tr>
<td>43</td>
<td>26.5334</td>
<td>-0.6707</td>
<td>0.1311</td>
<td>0.0768</td>
</tr>
<tr>
<td>44</td>
<td>11.3099</td>
<td>-0.2734</td>
<td>0.1413</td>
<td>0.0611</td>
</tr>
<tr>
<td>45</td>
<td>1.6252</td>
<td>0.0057</td>
<td>0.1131</td>
<td>0.0504</td>
</tr>
<tr>
<td>46</td>
<td>9.7842</td>
<td>-0.0922</td>
<td>0.0778</td>
<td>0.1121</td>
</tr>
<tr>
<td>47</td>
<td>1.6252</td>
<td>-0.0165</td>
<td>0.0892</td>
<td>0.0787</td>
</tr>
<tr>
<td>48</td>
<td>0.2653</td>
<td>-0.0540</td>
<td>-0.0742</td>
<td>0.0964</td>
</tr>
</tbody>
</table>

Figure 6.5.
<table>
<thead>
<tr>
<th>Time (s)</th>
<th>EXT VEL</th>
<th>MOVEMENT</th>
<th>Y CO-ORD</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>3.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
</tr>
<tr>
<td>6.0000</td>
<td>0.0117</td>
<td>0.0013</td>
<td>0.0007</td>
</tr>
<tr>
<td>9.0000</td>
<td>0.0008</td>
<td>0.0011</td>
<td>-0.0005</td>
</tr>
<tr>
<td>12.0000</td>
<td>-0.0007</td>
<td>-0.0021</td>
<td>-0.0001</td>
</tr>
<tr>
<td>15.0000</td>
<td>0.0048</td>
<td>-0.0056</td>
<td>0.0000</td>
</tr>
<tr>
<td>18.0000</td>
<td>0.0018</td>
<td>-0.0170</td>
<td>-0.0003</td>
</tr>
<tr>
<td>20.0000</td>
<td>0.0119</td>
<td>-0.0279</td>
<td>-0.0004</td>
</tr>
<tr>
<td>21.0000</td>
<td>0.0255</td>
<td>-0.0408</td>
<td>0.0004</td>
</tr>
<tr>
<td>22.0000</td>
<td>0.0602</td>
<td>-0.0386</td>
<td>0.0014</td>
</tr>
<tr>
<td>23.0000</td>
<td>0.1059</td>
<td>-0.0394</td>
<td>0.0029</td>
</tr>
<tr>
<td>24.0000</td>
<td>0.1359</td>
<td>-0.0456</td>
<td>0.0041</td>
</tr>
<tr>
<td>25.0000</td>
<td>0.1034</td>
<td>-0.0247</td>
<td>0.0034</td>
</tr>
<tr>
<td>26.0000</td>
<td>0.0681</td>
<td>-0.0262</td>
<td>0.0014</td>
</tr>
<tr>
<td>27.0000</td>
<td>0.0506</td>
<td>-0.0301</td>
<td>0.0006</td>
</tr>
<tr>
<td>29.0000</td>
<td>0.0267</td>
<td>-0.0189</td>
<td>0.0004</td>
</tr>
<tr>
<td>32.0000</td>
<td>0.0067</td>
<td>-0.0033</td>
<td>0.0005</td>
</tr>
<tr>
<td>35.0000</td>
<td>0.0069</td>
<td>-0.0025</td>
<td>0.0001</td>
</tr>
<tr>
<td>38.0000</td>
<td>0.0042</td>
<td>0.0057</td>
<td>0.0004</td>
</tr>
<tr>
<td>41.0000</td>
<td>0.0071</td>
<td>0.0045</td>
<td>0.0002</td>
</tr>
<tr>
<td>44.0000</td>
<td>0.0070</td>
<td>-0.0039</td>
<td>0.0007</td>
</tr>
<tr>
<td>47.0000</td>
<td>0.0070</td>
<td>-0.0039</td>
<td>0.0007</td>
</tr>
<tr>
<td>50.0000</td>
<td>0.0074</td>
<td>-0.0023</td>
<td>0.0000</td>
</tr>
<tr>
<td>53.0000</td>
<td>0.0074</td>
<td>-0.0023</td>
<td>0.0000</td>
</tr>
</tbody>
</table>
RUN 389 DATA

TOP WALL

JACK MACH NO.

MACH NO. RECORD (1-20) = 20

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>RECORD</th>
<th>WALL1</th>
<th>WALL2</th>
<th>WALL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0158</td>
<td>0.7634</td>
<td>0.0007</td>
<td>467</td>
</tr>
<tr>
<td>2</td>
<td>0.0183</td>
<td>0.7714</td>
<td>0.0005</td>
<td>566</td>
</tr>
<tr>
<td>3</td>
<td>0.0546</td>
<td>0.7563</td>
<td>-0.0001</td>
<td>551</td>
</tr>
<tr>
<td>4</td>
<td>0.0875</td>
<td>0.7520</td>
<td>-0.0000</td>
<td>611</td>
</tr>
<tr>
<td>5</td>
<td>0.1456</td>
<td>0.7742</td>
<td>-0.0003</td>
<td>639</td>
</tr>
<tr>
<td>6</td>
<td>0.8101</td>
<td>0.8001</td>
<td>-0.0004</td>
<td>692</td>
</tr>
<tr>
<td>7</td>
<td>0.8256</td>
<td>0.8001</td>
<td>0.0004</td>
<td>746</td>
</tr>
<tr>
<td>8</td>
<td>0.8748</td>
<td>0.8149</td>
<td>0.0014</td>
<td>890</td>
</tr>
<tr>
<td>9</td>
<td>0.9339</td>
<td>0.8121</td>
<td>0.0029</td>
<td>922</td>
</tr>
<tr>
<td>10</td>
<td>1.0569</td>
<td>0.7998</td>
<td>0.0041</td>
<td>941</td>
</tr>
<tr>
<td>11</td>
<td>1.1771</td>
<td>0.7998</td>
<td>0.0034</td>
<td>933</td>
</tr>
<tr>
<td>12</td>
<td>0.9200</td>
<td>0.9032</td>
<td>0.0014</td>
<td>896</td>
</tr>
<tr>
<td>13</td>
<td>0.9189</td>
<td>0.8758</td>
<td>0.0006</td>
<td>869</td>
</tr>
<tr>
<td>14</td>
<td>0.3407</td>
<td>0.8436</td>
<td>0.0006</td>
<td>870</td>
</tr>
<tr>
<td>15</td>
<td>0.2582</td>
<td>0.8159</td>
<td>0.0005</td>
<td>775</td>
</tr>
<tr>
<td>16</td>
<td>0.2401</td>
<td>0.8144</td>
<td>0.0001</td>
<td>318</td>
</tr>
<tr>
<td>17</td>
<td>0.2027</td>
<td>0.8130</td>
<td>0.0004</td>
<td>362</td>
</tr>
<tr>
<td>18</td>
<td>0.1696</td>
<td>0.8159</td>
<td>0.0002</td>
<td>339</td>
</tr>
<tr>
<td>19</td>
<td>0.1368</td>
<td>0.8173</td>
<td>0.0007</td>
<td>300</td>
</tr>
<tr>
<td>20</td>
<td>0.0842</td>
<td>0.8173</td>
<td>0.0007</td>
<td>365</td>
</tr>
</tbody>
</table>

BOTTOM WALL

<table>
<thead>
<tr>
<th>MACH NO</th>
<th>RECORD</th>
<th>WALL1</th>
<th>WALL2</th>
<th>WALL3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.0345</td>
<td>0.8077</td>
<td>-0.0005</td>
<td>583</td>
</tr>
<tr>
<td>2</td>
<td>0.0467</td>
<td>0.8060</td>
<td>-0.0003</td>
<td>502</td>
</tr>
<tr>
<td>3</td>
<td>0.0881</td>
<td>0.7981</td>
<td>0.0000</td>
<td>523</td>
</tr>
<tr>
<td>4</td>
<td>0.1119</td>
<td>0.7935</td>
<td>0.0004</td>
<td>531</td>
</tr>
<tr>
<td>5</td>
<td>0.1569</td>
<td>0.7721</td>
<td>0.0012</td>
<td>495</td>
</tr>
<tr>
<td>6</td>
<td>0.1523</td>
<td>0.7533</td>
<td>0.0008</td>
<td>382</td>
</tr>
<tr>
<td>7</td>
<td>0.1141</td>
<td>0.7367</td>
<td>-0.0000</td>
<td>474</td>
</tr>
<tr>
<td>8</td>
<td>0.0786</td>
<td>0.7459</td>
<td>-0.0014</td>
<td>509</td>
</tr>
<tr>
<td>9</td>
<td>0.0012</td>
<td>0.7563</td>
<td>-0.0030</td>
<td>568</td>
</tr>
<tr>
<td>10</td>
<td>-0.0035</td>
<td>0.7634</td>
<td>-0.0039</td>
<td>617</td>
</tr>
<tr>
<td>11</td>
<td>-0.0076</td>
<td>0.7714</td>
<td>-0.0015</td>
<td>576</td>
</tr>
<tr>
<td>12</td>
<td>-0.0025</td>
<td>0.7563</td>
<td>0.0000</td>
<td>593</td>
</tr>
<tr>
<td>13</td>
<td>0.0524</td>
<td>0.7520</td>
<td>0.0005</td>
<td>543</td>
</tr>
<tr>
<td>14</td>
<td>0.0557</td>
<td>0.7742</td>
<td>0.0004</td>
<td>662</td>
</tr>
<tr>
<td>15</td>
<td>0.0620</td>
<td>0.8001</td>
<td>0.0011</td>
<td>649</td>
</tr>
<tr>
<td>16</td>
<td>0.0080</td>
<td>0.8001</td>
<td>0.0015</td>
<td>320</td>
</tr>
<tr>
<td>17</td>
<td>0.0407</td>
<td>0.8149</td>
<td>0.0015</td>
<td>242</td>
</tr>
<tr>
<td>18</td>
<td>0.0146</td>
<td>0.8121</td>
<td>0.0016</td>
<td>210</td>
</tr>
<tr>
<td>19</td>
<td>-0.0180</td>
<td>0.7998</td>
<td>0.0021</td>
<td>211</td>
</tr>
<tr>
<td>20</td>
<td>0.0393</td>
<td>0.7998</td>
<td>0.0023</td>
<td>295</td>
</tr>
</tbody>
</table>

WALL & MODEL OUTPUT RECORD NO. = 21

STOP --
Wall data not taken at points 1, 2, 22-24

Assumed origin of wall boundary layer

Wall measuring points

Wall jack numbers

Flexible wall

Dummy straight wall extension

Wall anchor point

Wall and diffuser sliding joint

Dummy straight wall extension

FIG. 7 SOFTWARE REPRESENTATION OF EACH FLEXIBLE WALL
APPENDIX A. DIGITAL INPUT/OUTPUT PROTOCOL
FOR TSWT CONTROL SYSTEM

The transfer of digital information between the computer and wind tunnel hardware is an important part of the on-line control system. Digital I/O involves a complex interaction of system software and hardware. A command code has been devised to simplify the operation of the control system. The protocol of digital I/O are described in the following sections.

Data Format

Each packet of information sent to the wind tunnel consists of the data destination and operation plus the data itself. Information from the wind tunnel consists of data only. All sets of information are contained in a single 16-bit word in the following manner:

Digital Output Word (Equivalent to a six digit octal number in binary code)

```
<table>
<thead>
<tr>
<th>Digits</th>
<th>Bits</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>15</td>
</tr>
<tr>
<td></td>
<td>14</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>
```

- Data 2
- Data 1
- Address (0-63 Devices)

Digital Input Word

```
<table>
<thead>
<tr>
<th></th>
<th>Data 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
```

- Read
- Write

- A1 -
Device Addresses

Each address consists of six bits of binary information which corresponds to a decimal number referred to as the software address. The following scheme has been chosen:

<table>
<thead>
<tr>
<th>Device</th>
<th>Software Address</th>
<th>Binary Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stepper Motors</td>
<td>1 → 44</td>
<td>5 4 3 2 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 0 0 0 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 1 1 0</td>
</tr>
<tr>
<td>Pulse Sequence</td>
<td>45</td>
<td>5 4 3 2 1 0</td>
</tr>
<tr>
<td>Generator</td>
<td></td>
<td>1 0 1 1 0</td>
</tr>
<tr>
<td>Scanivalves (1)</td>
<td>46</td>
<td>1 0 1 1 1 0</td>
</tr>
<tr>
<td></td>
<td>(2)</td>
<td>1 1 0 0 0</td>
</tr>
<tr>
<td>Encoders (1)</td>
<td>47</td>
<td>1 0 1 1 1 1</td>
</tr>
<tr>
<td></td>
<td>(2)</td>
<td>1 1 0 0 0 1</td>
</tr>
<tr>
<td>Power Supplies</td>
<td>50 → 63</td>
<td>5 4 3 2 1 0</td>
</tr>
<tr>
<td>etc.</td>
<td></td>
<td>1 1 0 0 1 0</td>
</tr>
</tbody>
</table>

Data Operation

The transfer of data takes three forms - read only, write only and write before read. These operations determine the type of data to be sent, if any. For example, the read only function requires no data from the computer other than the device address.

Information on data operation is sent to the wind tunnel by adding a chosen software value to the decimal equivalent of the digital output word. This effectively sets the required bits for correct information transfer. The software values are chosen thus:

<table>
<thead>
<tr>
<th>Data Information</th>
<th>Binary Representation</th>
<th>Software Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read only</td>
<td>15 8 7 6</td>
<td>-32768 + 128 = -32640</td>
</tr>
<tr>
<td></td>
<td>1 0 1 0</td>
<td></td>
</tr>
<tr>
<td>Write only</td>
<td>0 0 0 0</td>
<td>0 + 0 = 0</td>
</tr>
<tr>
<td>Write before read</td>
<td>0 0 1 0</td>
<td>0 + 128 = 128</td>
</tr>
</tbody>
</table>

Digit 6 3
Data

Actions at the wind tunnel are determined by the corresponding data 1 or 2 bits of the output word. Data 3 bits of the input word allow checks to be made on system devices.

The output data is transferred by the setting of data 1 and 2 bits, achieved by the modification of the digital output word as for the data operating information.

The software values were chosen thus:

<table>
<thead>
<tr>
<th>Function</th>
<th>Binary Representation</th>
<th>Software Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>a) Motor direction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stop</td>
<td>0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>Forward-go</td>
<td>1 0 1</td>
<td>2560</td>
</tr>
<tr>
<td>Reverse-go</td>
<td>1 1 0</td>
<td>3072</td>
</tr>
<tr>
<td>b) Scanivalve Move</td>
<td>14 13 12</td>
<td></td>
</tr>
<tr>
<td>-Home</td>
<td>1 0 1</td>
<td>20480</td>
</tr>
<tr>
<td>-Step on one</td>
<td>1 0 0</td>
<td>16384</td>
</tr>
<tr>
<td>c) Pulse sequence generator</td>
<td>14 13 12</td>
<td></td>
</tr>
<tr>
<td>-Start</td>
<td>0 0 1</td>
<td>4096</td>
</tr>
<tr>
<td>d) Motor Power Supply</td>
<td>14 13 12</td>
<td></td>
</tr>
<tr>
<td>-Off</td>
<td>0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>-On</td>
<td>0 1 0</td>
<td>8192</td>
</tr>
</tbody>
</table>
Function Binary Representation Software Value

e) Pulse sequence generator 14 13 12 11 10 9 8
Increment step size (currently non operational).

No. 10s 100s 1000s

Input data will be in binary code to allow simple software manipulation. The complete input word will always be read regardless of the quantity of information being transferred.

The types of data are as follows:

<table>
<thead>
<tr>
<th>Device</th>
<th>No. of bits of Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scanivalve - encoder output</td>
<td>7</td>
</tr>
<tr>
<td>- at home position</td>
<td>1</td>
</tr>
<tr>
<td>Pulse sequence generator</td>
<td></td>
</tr>
<tr>
<td>- finish pulse</td>
<td>1</td>
</tr>
<tr>
<td>- step value</td>
<td>7</td>
</tr>
<tr>
<td>System monitor</td>
<td>16</td>
</tr>
<tr>
<td>Motor direction</td>
<td>3</td>
</tr>
</tbody>
</table>

Command Coding

Each command sent to the wind tunnel must be unambiguous and provide information on data destination and operation and the data itself. This is achieved by placing a decimal code number in binary on the 16 output lines. This command number N is then decoded by the wind tunnel hardware and some operation performed.

The command number for each operation is determined by the code

\[ N = \text{Device Address} + \text{Data Operation Value} + \text{Data Value} \]
using software values only. For example, if the scanivalve (1) is required to step on one port, the following command number would be sent to the wind tunnel

\[ N = 46 + 0 + 16384 = 16430 \]

**Summary**

The interface between wind tunnel hardware and the computer is by means of 16 output lines and 16 input lines plus control lines. The voltage levels on the output lines are controlled by the software generated command numbers described above. The voltage levels on the input lines are controlled by software selected tunnel hardware. All transfer of data is accompanied by a handshaking procedure to ensure correct sequencing of communication operations.
### Abstract

The current operation of the Transonic Self-Streamlining Wind Tunnel (TSSW) involves on-line data acquisition with automatic wall adjustment. A tunnel run consists of streamlining the walls from known starting contours in iterative steps and acquiring model data. Each run performs what is described as a streamlining cycle. The associated control software is presented here.
End of Document